

UNIVERSIDADE ESTADUAL PAULISTA
“Júlio de Mesquita Filho”
Pós-Graduação em Ciência da Computação

Caio Miguel Marques

**Pangea – Arquitetura semântica para a integração de dados e modelos
geoespaciais na Web**

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

Caio Miguel Marques

**Pangea – Arquitetura semântica para a integração de dados e modelos
geoespaciais na Web**

Dissertação apresentada para a obtenção do título de Mestre em Ciência da Computação, área de Concentração em Sistemas Inteligentes, junto ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Campus de São José do Rio Preto.

Orientador: Prof. Dr. Ivan Rizzo Guilherme

Caio Miguel Marques

Pangea – Arquitetura semântica para a integração de dados e modelos
geoespaciais na Web

Dissertação apresentada para a obtenção do título de Mestre em
Ciência da Computação, área de Concentração em Sistemas
Inteligentes, junto ao Programa de Pós-Graduação em Ciência
da Computação do Instituto de Biociências, Letras e Ciências
Exatas da Universidade Estadual Paulista “Júlio de Mesquita
Filho”, Campus de São José do Rio Preto.

BANCA EXAMINADORA

Prof. Dr. Ivan Rizzo Guilherme
UNESP – Rio Claro
Orientador

Prof. Dra. Marilde Terezinha Prado Santos
UFSCar – Universidade Federal de São Carlos

Prof. Dr. Carlos Roberto Valêncio
UNESP – São José do Rio Preto

Dedico este trabalho à minha mãe Alda e ao meu pai Carlos, meus principais exemplos de dedicação extrema ao trabalho.

AGRADECIMENTOS

Agradeço ao meu orientador Professor Ivan Rizzo Guilherme pela ótima oportunidade de realizar um trabalho de mestrado. Agradeço também por ter confiado em meu trabalho, me orientado nos momentos mais difíceis, ter realizado críticas quando necessário e por ter me dado a oportunidade de realizar o estágio de docência em sua disciplina. Foi um prazer ter trabalhado em conjunto durante todo o tempo de graduação e do mestrado.

Agradeço de forma carinhosa todo o apoio de minha família. Eles foram fundamentais para que eu pudesse concluir o trabalho. Agradeço de forma especial ao meu Pai Carlos, meu irmão Carlinhos e minha irmã Karina por terem me ajudado nessa reta final com os compromissos em casa.

À minha namorada Caroline, que aguentou firme todo o tempo do mestrado. Compreendeu toda a dedicação e o esforço necessários e empenhados no trabalho, em detrimento de momentos felizes ao seu lado. A ela, agradeço todo o apoio, consideração e sobretudo compreensão.

Aos grandes amigos da Caiena, em especial aos meus sócios Eduardo Assis e Eduardo Foster, que permitiram que eu me afastasse da empresa para realizar minhas atividades do mestrado. A eles e todos os demais batalhadores da Caiena, meu agradecimento todo especial.

Aos membros da banca de qualificação e defesa Professor Carlos Roberto Valêncio, Professora Marilde Terezinha Prado Santos e Professor José Remo Ferreira Brega pelas contribuições extremamente valiosas que fizeram para o meu trabalho. Em especial, gostaria de agradecer ao Professor Valêncio pelo abraço fraterno ao final da qualificação. Esse abraço me deu forças para prosseguir adiante.

Ao meu grande amigo João Felipe Von Zuben, pelo companheirismo e pelas anotações que realizou com detalhes sobre minha qualificação.

A todos os amigos que fiz no período de graduação e pós-graduação!

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, CAPES por ter financiado meus estudos! Foi muito importante esse apoio!

Por fim, gostaria de agradecer de forma especial à minha querida mãe Alda! Na manhã do dia da qualificação, mesmo debilitada, ela saiu do seu leito e veio até mim para dar um forte abraço e desejar sorte. Esse episódio foi extremamente marcante e faço aqui meus sinceros agradecimentos.

RESUMO

Em muitas áreas do conhecimento e da atividade humana é requerida, impreterivelmente, a integração de informações geográficas. Atualmente, grande quantidade dessas informações geográficas estão publicadas na Web, por atores diversos, indo desde instituições governamentais, academia, até cidadãos comuns. Esses atores publicam dados geográficos em diversos formatos e utilizando tecnologias variadas. Neste contexto, apesar da enorme quantidade de dados e modelos geográficos publicados na Web, a diversidade de formatos e tecnologias nos quais são disponibilizados, somada à carência das soluções atualmente existentes, limitam o consumo, a integração e o compartilhamento das informações geográficas. Recentemente tem sido propostas abordagens que agregam semântica na descrição das informações geográficas, de modo a possibilitar melhorias no descobrimento e integração desse tipo de informação. Nesse sentido, neste trabalho é apresentado um levantamento das arquiteturas e infraestruturas semânticas utilizadas na integração e compartilhamento de dados e modelos geográficos. Com base nesse levantamento foram identificados os aspectos transversais às infraestruturas estudadas. Tais aspectos foram utilizados na definição do projeto da arquitetura descrita neste trabalho, denominada Pangea, que é composta dos seguintes módulos: anotação semântica, alinhamento de descrição semântica, repositórios semânticos, descobrimento e integração semântica de dados e modelos geográficos. Dentre os módulos mencionados foi implementado o repositório semântico e algumas funcionalidades referentes ao descobrimento e integração semântica de dados. Para avaliar os componentes implementados da Pangea é apresentado um estudo de caso referente ao contexto de derramamento de óleo no litoral.

ABSTRACT

The geographic information is definitely required in many areas of human knowledge and activity. Nowadays, a large part of this geographic information is published on the Web by various authors, from the governmental institutions and academy to the ordinary citizen. These authors publish the geographic data in several formats and using different technologies. In this context, in spite of having a great amount of available data on the Web, the diversity of formats and technologies that they are released, limit the consumption, the integration and the geographic information sharing. Recently, it has been proposed the approach that adds the semantics in the description of geographic information, so the discovery and integration can be enhanced. This work presents a study of semantics architectures and frameworks used in the geographic data integration and sharing. Based in this study, the transversal aspects to the studied architectures were identified. Those aspects were used in the project definition of the Pangea architecture which is composed by the following modules: semantic notation, alignment of semantic description, and semantic integration. In order to evaluate some of the Pangea components, a study of case is conducted in the problems of the environmental domain, considering oil blowout disasters.

SUMÁRIO

RESUMO
ABSTRACT
SUMÁRIO

1 Introdução.....	15
2 Aspectos conceituais.....	18
2.1 Serviços e Serviços web.....	18
2.2 Web Semântica	23
2.3 Serviços Web Semânticos.....	30
2.4 Conclusões do capítulo.....	35
3 Aspectos transversais às abordagens semânticas	36
3.1 Arquiteturas para o compartilhamento de dados e modelos geográficos	36
3.2 Disponibilização dos dados e operações em serviços web OGC	47
3.3 Servidor semântico.....	49
3.4 Anotação semântica.....	53
3.5 Integração de ontologias.....	63
3.6 Especificação e processamento de consultas.....	67
3.7 Descobrimto de serviços web semânticos	73
3.8 Conclusões do capítulo.....	81
4 Pangea.....	82
4.1 Requisitos.....	82
4.2 Organização geral da arquitetura Pangea.....	86
4.3 Módulo Semantic Middleware.....	88
4.4 Módulo Semantic Utils.....	99
4.5 Semantic Server.....	130
4.6 Personalization Utils.....	142
4.7 Language.....	143
5 Aspectos de implementação.....	145
5.1 Processo adotado na Implementação	145
5.2 Disponibilização dos dados geográficos como serviços OGC	148
5.3 Implementação das ontologias de domínio	151
5.4 Implementação do Semantic Server.....	155
5.5 Implementação do Semantic Middleware.....	161
5.6 Disponibilização como serviços web	164
5.7 Estudo de caso.....	167
6 Conclusões e extensões.....	182
6.1 Conclusões.....	182
6.2 Extensões.....	184
REFERÊNCIAS.....	186
APÊNDICE A: Shell script utilizado para realizar a carga dos shapefiles no banco de dados.....	193
APÊNDICE B: Shell script utilizado para criar os DataStores no GeoServer.....	194
APÊNDICE C: Shell script utilizado para criar os FeatureTypes no GeoServer.....	195

LISTA DE ILUSTRAÇÕES

Figura 1. Arquitetura da Web Semântica.	23
Figura 2. Exemplo de um arquivo XML.	25
Figura 3. Representação em grafo das triplas RDF.	27
Figura 4. Esquema geral da arquitetura do DAGIS.	38
Figura 5. Componentes do SWING.	40
Figura 6. Arquitetura do sistema WeBIOS.	42
Figura 7. Modelo conceitual para a anotação semântica de dados geográficos.	55
Figura 8. Estratégia de anotação semântica.	56
Figura 9. Modelo conceitual adotado neste trabalho.	59
Figura 10. Estratégia de anotação semântica adotada neste trabalho.	60
Figura 11. Etapas gerais do descobrimento de recursos.	67
Figura 12. Modelo conceitual para descobrimento de serviços.	76
Figura 13. Esquema geral de descobrimento de serviços adotado na Pangea.	79
Figura 14. Esquema geral da Pangea.	87
Figura 15. Estrutura geral do módulo Semantic Middleware.	88
Figura 16. Representação em grafo da consulta.	91
Figura 17. Pseudocódigo da operação Solv do serviço web Spatial Problem Solver.	92
Figura 18. Consulta SPARQL para o descobrimento de um objetivo que representa a necessidade de informação expressa no nó do grafo de consulta.	93
Figura 19. Pseudocódigo da operação Discovery do serviço web Goal Discovery.	94
Figura 20. Consulta SPARQL para selecionar os ServiceProfiles candidatos ao procedimento de “casamento”	95
Figura 21. Pseudocódigo da operação Match do serviço web Matchmaker.	97
Figura 22. Esquema geral da anotação semântica.	102
Figura 23. Pseudocódigo da operação Extract.	104
Figura 24. Esquema geral do enriquecimento semântico.	105
Figura 25. Pseudocódigo da operação Enrich do serviço web EnrichDomainOntology.	108
Figura 26. Pseudocódigo da operação Enrich do serviço web EnrichServiceOntology.	109
Figura 27. Pseudocódigo da operação createServiceProfile do serviço web Process Annot.	112
Figura 28. Estrutura geral da ontologia ORE.	114
Figura 29. Pseud. da operação Spatial Annot do serviço web Spatial Output Annot.	115
Figura 30. Pseudocódigo da operação Text Annot do serviço web Text Similarity Annot.	117
Figura 31. Pseudocódigo da operação Infer do serviço web Inference Annot.	118
Figura 32. Pseudocódigo da operação Align Ontology.	125
Figura 33. Pseud. da op. Semantic Align Ontology do serviço web Semantic Similarity.	127
Figura 34. Pseud. da op. SpatialAlignOntology do serviço web Spatial Align Ontology.	129
Figura 35. Arquitetura geral do Semantic Server.	133
Figura 36. Pseudocódigo da operação Publish.	138
Figura 37. Pseudocódigo da operação GetKnowledgeByContext.	139
Figura 38. Pseudocódigo da operação GetKnowledge.	139
Figura 39. Pseudocódigo da operação Reasoning.	141
Figura 40. Arquivo de descrição dos DataStore.	148
Figura 41. Arquivo de descrição das FeatureTypes.	149
Figura 42. Fragmento da resposta resultante da chamada à operação GetCapabilities.	150
Figura 43. Fragmento da ontologia SAO (destaque para os procedimentos de limpeza).	152
Figura 44. Fragmento da ontologia POI.	153
Figura 45. Fragmento do arquivo XML descritor de repositórios.	159
Figura 46. Classes implementadas no Semantic Middleware.	161

Figura 47. Área geográfica de interesse do estudo de caso.	167
Figura 48. Serviços OGC considerados no estudo de caso.	170
Figura 49. Modelo conceitual dos elementos criados para anotação.	172
Figura 50. Serviços OGC anotados com os conceitos de domínio.	172
Figura 51. Representação da consulta considerada no estudo de caso.	175
Figura 52. Esquema geral do processamento dos nós do grafo de consulta.	176
Figura 53. Grafo que representa a consulta do usuário.	177
Figura 54. Consulta para descobrir os objetivos que correspondem ao nó.	177
Figura 55. Consulta para descobrir conceitos mais gerais.	178
Figura 56. Consulta para descobrimento de objetivos.	178
Figura 57. Nível de casamento exato entre o objetivo e a descrição do serviço OGC.	179
Figura 58. Visualização dos dados geográficos descobertos e integrados.	180

LISTA DE TABELAS

Tabela 1 - Características gerais de serviços web.	19
Tabela 2 - Síntese dos aspectos transversais discutidos.	80
Tabela 3 - Descrição da operação Solv do serviço web Spatial Problem Solver.	92
Tabela 4 - Descrição da operação Discovery do serviço web Goal Discovery.	94
Tabela 5 - Descrição da operação Match do serviço web Matchmaker.	96
Tabela 6 - Descrição da operação Extract do serviço web Extractor.	104
Tabela 7 - Descrição da operação Enrich do serviço web Enrich Domain Ontology.	108
Tabela 8 - Descrição da operação Enrich do serviço web Enrich Service Ontology.	108
Tabela 9 - Descrição da operação createServiceProfile do serviço web Process Annot.	111
Tabela 10 - Descrição da operação Spatial Annot do serviço web Spatial Output Annot.	113
Tabela 11 - Descrição da operação Text Annot do serviço web Text Similarity Annot.	116
Tabela 12 - Descrição da operação Infer Annot do serviço web Inference Annot.	118
Tabela 13 - Descrição geral da operação Align Ontology.	124
Tabela 14 - Descrição geral da operação Semantic Align Ontology.	127
Tabela 15 - Descrição geral da operação Spatial Align Ontology.	128
Tabela 16 - Descrição da operação Publish do serviço Publisher.	137
Tabela 17 - Descrição da operação GetKnowledgeByContext.	138
Tabela 18 - Descrição da operação GetKnowledge.	139
Tabela 19 - Descrição da operação Reasoning.	140
Tabela 20 - Ferramentas utilizadas na implementação dos componentes da Pangea.	147
Tabela 21 - Descrição dos métodos da classe AbstractRepository.	155
Tabela 22 - Repositórios implementados no Semantic Server.	157
Tabela 23 - Descrição dos métodos da classe Reasoner.	158
Tabela 24 - Descrição dos métodos da classe Bootstrap.	159
Tabela 25 - Descrição dos métodos da classe GoalDiscovery.	162
Tabela 26 - Descrição dos métodos da classe Matchmaker.	162
Tabela 27 - Descrição dos métodos da classe MatchmakerUtils.	163
Tabela 28 - Map. entre projeto e implementação considerando o módulo Semantic Server.	164
Tabela 29 - Map. entre projeto e implem. considerando o módulo Semantic Middleware.	165
Tabela 30 - Descrição dos repositórios a serem criados.	174
Tabela 31 - Comparação entre Pangea e arquiteturas estudadas.	182

LISTA DE SIGLAS

ANA – Agência Nacional de Águas
CONCAR – Comissão Nacional de Cartografia
CSW – Catalog Service for the Web
DAGIS – Discovering Annotated Geospatial Information Service
GIF – Graphics Interchange Format
GML – Geography Markup Language
GPS – Global Positioning System
HTTP – Hypertext Transfer Protocol
IDE – Infraestruturas de Dados Espaciais
IPEA – Instituto de Pesquisa Econômica Aplicada
ISO – International Organization for Standardization
JDT – Java Development Toolkit
JPEG – Joint Photographic Experts Group
MiMS – Mineral resources Management System
MMA – Ministério do Meio Ambiente
OGC – Open Geospatial Consortium
OWL – Web Ontology Language
PNG – Portable Network Graphics
POI – Ponto de Interesse
RDF – Resource Description Framework
REST - REpresentational State Transfer
SAO – Sensibilidade Ambiental ao derramamento de Óleo
SIG – Sistemas de Informação Geográfica
SOA – Service Oriented Architectures
SOAP – Simple Object Access Protocol
SPARQL – SPARQL Protocol And Rdf Query Language
SVG – Scalable Vector Graphics
SWING – Semantic Web Services Interoperability for Geospatial Domain
SWRL – Semantic Web Rule Language
UDDI – Universal Description, Discovery and Integration
URI – Universal Resource Identification
URL – Uniform Resource Locator

XML – eXtensible Markup Language

WCS – Web Coverage Service

WebCGM – Web Computer Graphics Metafile

WeBIOS – Web Service Multimodal Tools for Strategic Biodiversity Research, Assessment and Monitoring

WFS – Web Feature Service

WMS – Web Map Service

WPS – Web Processing Service

WSDL – Web Service Description Language

WSML – Web Service Modeling Language

WSMF – Web Service Modeling Framework

WSMO – Web Service Modeling Ontology

WSMX – Web Service Execution Environment

W3C- World Wide Web Consortium

Capítulo 1

Introdução

Há muitas áreas do conhecimento e da atividade humana que requerem impreterivelmente dados e modelos geográficos para suportar o processo de tomada de decisão. O conceito *dado geográfico* denota todo tipo de dado que possui uma dimensão espacial associada, cujas coordenadas são definidas em função da localização referente à superfície da Terra. O conceito *modelo geográfico* é definido como todo processo que manipula dados geográficos.

Assim, dados e modelos geográficos são utilizados de maneira decisiva em variados contextos: gestão de recursos naturais, serviços à população, prevenção de desastres, agricultura, biodiversidade, meio ambiente, entre muitos outros. Tais contextos dependem fundamentalmente de uma visão integrada e multifacetada da realidade e, além disso, demandam a colaboração entre equipes multidisciplinares, compostas de atores com experiências e culturas diversas. A diversidade de atores bem como de dados e modelos levam a problemas em trocar e compartilhar dados.

Por vários anos, uma solução para esse cenário complexo, foi baseada nos Sistemas de Informação Geográfica – SIG. Os SIGs consistem em sistemas que provêem mecanismos para armazenar, analisar e visualizar dados geográficos (CÂMARA; DAVIS JR.; MONTEIRO, 2001). Se tratam de ferramentas importantes pois, permitem organizar a realidade por meio de uma perspectiva comum: o espaço geográfico. No entanto, com o passar do tempo, notou-se que a coleta e organização de dados geográficos são atividades custosas. Desse modo, foram impostos aos SIGs novos requisitos, sobretudo relacionados à integração e compartilhamento de dados e modelos geográficos. Tais requisitos foram determinantes para a evolução das arquiteturas de SIG, resultando nas *Infraestruturas de Dados Espaciais – IDE*. As IDE são definidas como um conjunto de políticas, padrões técnicos (amplamente baseados em metadados e serviços web) e arranjos institucionais que, se adotados, permitem potencializar a publicação e o compartilhamento de dados geográficos por meio de tecnologias de redes (SDI COOKBOOK, 2009).

A adoção de IDEs permitiu a publicação de repositórios de dados geográficos na Web segundo políticas e tecnologias bem definidas. Além disso, houve aumento na quantidade de dados geográficos publicados na Web. Porém, embora seja evidente o avanço decorrente da adoção das IDEs, o cenário onde haja compartilhamento e integração de dados e modelos geográficos se

encontra longe de ser realizado em sua plenitude. Isso se deve, sobretudo, às enormes quantidades e variedade dos dados geográficos disponíveis atualmente na Web e ao estado atual dos mecanismos de descrição, recuperação e integração de informação.

Atualmente há uma grande quantidade de informações geográficas publicadas na Web. As informações são publicadas por atores diversos, indo desde instituições governamentais, academia, até cidadãos comuns.

O contexto apresentado culmina em dois paradoxos bastante investigados recentemente: (i) apesar da enorme quantidade de dados disponíveis através da Web, a diversidade de formatos e tecnologias nos quais são disponibilizados, somada às soluções atualmente existentes, dificultam o consumo, a integração e o compartilhamento dos dados geográficos; (ii) Ainda que os dados e modelos sejam disponibilizados conforme um padrão de dados e, portanto, possam ser acessados e trocados entre sistemas que compartilham a especificação, os mecanismos atuais de descrição e descobrimento de dados e modelos geográficos são baseados em descrição sintática e, portanto, pouco precisos.

Portanto, há dificuldades relacionadas em localizar os dados e modelos adequados à resolução de um problema. Ainda que os dados e modelos sejam descobertos, há problemas relacionados com a interoperabilidade sintática e semântica de tais recursos. Uma solução possível para os problemas apresentados tem sido a adoção de descrição semântica (ex. Ontologias) para descrever, descobrir e integrar os dados e modelos geográficos.

Desse modo, o principal objetivo desta dissertação é especificar uma arquitetura composta por módulos que permitam descrever, descobrir, integrar e compartilhar dados e modelos geográficos. A arquitetura apresentada neste trabalho é denominada Pangea.

Para permitir alcançar o objetivo proposto, a primeira etapa consistiu no levantamento de arquiteturas que empregam semântica para descrever, descobrir, integrar e compartilhar dados geográficos. Foram selecionadas três arquiteturas para o estudo: (i) *DAGIS – Discovering Annotated Geospatial Information Service* (ALAM; SUBBIAH; KHAN; THURASINGHAM, 2007); (ii) *SWING – Semantic Web Services Interoperability for Geospatial Domain* (ROMAN; KLIEN; SKOGAN, 2006); (iii) *WeBIOS – Web Service Multimodal Tools for Strategic Biodiversity Research* (GOMES JR., 2007). Com base no estudo dessas arquiteturas foram identificados aspectos transversais, que serviram de base para a definição dos requisitos que devem ser suportados pelo projeto da Pangea. Após, foram concebidos e elaborados os módulos que devem suportar os requisitos da arquitetura. Cada módulo é composto de pacotes e serviços web. Por fim, para avaliar alguns dos componentes implementados da Pangea, foi realizado um estudo de caso com serviços OGC de dados e modelos geográficos. No estudo de caso é considerado o

descobrimto e integração de dados geográficos em cenários de derramamento de óleo.

O restante do trabalho está organizado como segue: no capítulo 2 são apresentados os conceitos básicos necessários para o entendimento deste trabalho; no capítulo 3 são discutidos os aspectos transversais identificados nas arquiteturas estudadas; no capítulo 4 o projeto da arquitetura Pangea é apresentado. No capítulo 5 são abordados os aspectos de implementação e é descrito um estudo de caso. No capítulo 6 são feitas as conclusões e consideradas algumas extensões possíveis ao trabalho.

Capítulo 2

Aspectos conceituais

Neste capítulo são apresentados os aspectos conceituais que são necessários para o entendimento do restante do trabalho. Assim, este capítulo é organizado conforme segue: na seção 2.1 são definidos e diferenciados os conceitos *Serviços* e *Serviços web*. Além disso, são apresentados os padrões para serviços web geográficos, definidos pelo *Open Geospatial Consortium – OGC*; Na seção 2.2 são abordados os conceitos e tecnologias relacionadas com a proposta da Web Semântica; Finalmente, na seção 2.3, são descritas as propostas de *frameworks* utilizados para a descrição semântica de serviços web.

2.1 Serviços e Serviços web

Há muitas comunidades que fazem uso do termo Serviço e de suas variações (*Web Services*, *e-services*, entre outros) levando a conceituações e interpretações distintas. Baida e colaboradores (BAIDA; GORDIJN; OMELAYENKO, 2004) apresentam esse confuso cenário, onde o termo Serviço e suas variações são utilizados pelas comunidades de negócios, ciência da informação e ciência da computação com semânticas específicas de cada comunidade. Neste trabalho, dentre as variações de definição mencionadas no trabalho de Baida, Gordijn e Omelayenko (2004), é apresentada a definição de Serviço e de Serviço web.

A definição de Serviço é um tanto geral e advém da comunidade de negócios: um Serviço consiste em uma atividade de negócio em um determinado domínio, que torna disponível um valor ou um benefício a um consumidor (BAIDA; GORDIJN; OMELAYENKO, 2004) (PREIST, 2004). Exemplos de serviços podem ser: tradução de documentos, serviços de crédito bancário, aluguel de carros, entre muitos outros.

Por outro lado, o termo Serviço web designa componentes de *software* que disponibilizam alguma funcionalidade por meio de protocolos da Internet (BAIDA; GORDIJN; OMELAYENKO, 2004) (PREIST, 2004).

Uma vez apresentadas as definições de Serviço e Serviço web utilizados neste trabalho, é

importante destacar os aspectos principais que caracterizam os Serviços web.

Os serviços web são caracterizados por serem fracamente acoplados, auto-descritos, auto-contidos e passíveis de serem reutilizados. O acoplamento fraco decorre do conceito de reuso caixa-preta (MACHADO, 2004). Esse tipo de reutilização é baseado na definição de interfaces e contratos. Os componentes de *software* se ligam através dessas interfaces e, portanto, uma alteração na implementação, desde que a interface permaneça inalterada, não causa impactos em outras partes do sistema. Serviços web são fortemente baseados em definições de interfaces (MACHADO, 2004) permitindo desse modo baixo acoplamento (BREIVOLD; LARSON, 2007).

Os serviços web são auto-descritos pois provêem aos consumidores descrições que habilitam a interação entre as partes. As descrições podem considerar aspectos sintáticos e semânticos relacionados ao serviço web. A característica de serem auto-contidos implica que serviços web são componentes de *software* que podem ser executados independente de outros serviços web. Por fim, essas características em conjunto determinam um alto grau de reutilização, característica marcante dos serviços web. Essas características são sumariadas na tabela 1.

Tabela 1 - Características gerais de serviços web.

Características gerais de serviços web		
Característica	Descrição geral	Aspectos que determinam a presença da característica
Acoplamento fraco	Consiste na baixa interdependência entre entidades de software.	Serviços web são baseados em descrição de interfaces e contratos.
Auto-Descrito	Consiste nas descrições necessárias para utilizar a entidade de software.	Serviços web possuem vários níveis de descrição associados, que permitem invocar suas funcionalidades.
Auto-Contido	Consiste na capacidade de um serviço web executar sua computação sem depender de outros serviços web.	Serviços web são unidades de funcionalidade atômicas que podem ser executadas sem depender de outros serviços web (esse aspecto é relaxado nos casos onde são consideradas composições de serviços atômicos).
Reutilização	Consiste no grau de facilidade com que uma entidade de <i>software</i> pode ser reutilizada.	Serviços web podem ser reutilizados pois são baseados em descrições de interfaces e contratos, auto-descritos e auto-contidos.

A definição apresentada para o conceito serviço web é geral e, no entanto, contrasta com a definição do *World Wide Web Consortium – W3C*, amplamente ligada aos aspectos técnicos de implementação dos serviços web (BAIDA; GORDIJN; OMELAYENKO, 2004).

No decorrer deste trabalho o termo serviço web é empregado com a semântica baseada na definição abstrata exposta anteriormente e não remete a quaisquer aspectos técnicos empregados em sua implementação. As diferenças claras de definição entre serviços e serviços web são importantes, sobretudo, quando considerado o procedimento de descobrimento de serviços web, conforme enfatizam Fensel e colaboradores (FENSEL; KELLER; LAUSEN; POLLERES; TOMA, 2005).

Atualmente, serviços web tem sido adotados para disponibilizar dados e modelos geográficos na Web.

2.1.2 Serviços OGC

O *Open Geospatial Consortium* - OGC (OGC REFERENCE MODEL, 2008) é uma organização internacional composta por setores da indústria, da academia e de governos, cujo objetivo principal é liderar a especificação de padrões para serviços web geográficos. São denominados serviços web geográficos aqueles que disponibilizam funcionalidades de descobrimento, recuperação e processamento de dados geográficos.

Antes de apresentar os padrões de serviços geográficos, é importante apresentar os conceitos de *feature* e *feature type*. Nesse sentido e, de acordo com o *OGC Reference Model* (2008), uma *feature* é denotada como uma abstração de um fenômeno do mundo real. Por conseguinte, uma *feature* geográfica é uma abstração de um fenômeno do mundo real que possui uma localização referente à Terra. Ao conjunto de *features* que compartilham as mesmas características denomina-se *feature type*.

O OGC especificou uma série de padrões de serviços web que são baseadas nos conceitos definidos no modelo de referência do OGC. Os principais padrões para serviços web geográficos são os seguintes (OGC REFERENCE MODEL, 2008):

- **Web Map Service (WMS)** – O serviço web no padrão WMS disponibiliza acesso às operações que permitem gerar representações de dados geográficos por meio da metáfora de mapas. Os mapas retornados podem ser representados em formato matricial ou vetorial. Os formatos matriciais aceitos são: *Portable Network Graphics – PNG*, *Graphics Interchange Format – GIF* e *Joint Photographic Experts Group – JPEG*. Já os formatos vetoriais aceitos são: *Scalable Vector Graphics – SVG* e *Web Computer Graphics Metafile – WebCGM*. O WMS suporta três operações: *GetCapabilities* (obrigatório), *GetMap* (obrigatório) e *GetFeatureInfo* (opcional). A operação *GetCapabilities* retorna os metadados com a descrição sintática do serviço WMS. A operação *GetMap* é a operação responsável por prover os mapas. Uma requisição à operação *GetMap* contém como parâmetros a lista de camadas (*layers*) e a área geográfica de interesse. Como resultado, se tem a representação das camadas sobrepostas em um mapa. A operação *GetFeatureInfo* possibilita obter informações extras acerca de uma determinada *feature* presente em um mapa. Entretanto, essa operação somente está disponível para as camadas que possuem a propriedade *queryable* setada igual a 1;
- **Web Feature Service (WFS)** – O serviço web no padrão WFS disponibiliza operações que

permitem a um consumidor recuperar e atualizar dados geográficos. Os dados geográficos são codificados segundo o padrão *Geography Markup Language – GML*. Há duas classes de WFS: (i) Básico; (ii) Transacional. A classe Básico denota os WFS que permitem apenas recuperar informações geográficas, através das seguintes operações: *GetCapabilities* (obrigatório), *DescribeFeatureType* (obrigatório) e *GetFeature* (obrigatório). A operação *GetCapabilities*, assim como no WMS, retorna os metadados com a descrição sintática referentes ao WFS. A operação *DescribeFeatureType* retorna o esquema de dados que descreve determinada *Feature Type*. A operação *GetFeature* permite recuperar as *Features* por meio da aplicação de filtros (sintáticos ou espaciais). A classe Transacional denota os WFS que disponibilizam também operações para modificar os dados geográficos (i.e. Inserir, atualizar e apagar informações geográficas). O WFS Transacional requer as seguintes operações: (i) *Transaction* (obrigatório); (ii) *LockFeature* (opcional). A operação *Transaction* permite alterar as *Features* disponibilizadas pelo WFS. Por fim, a operação *LockFeature* permite bloquear o acesso a determinada *feature* até que uma operação transacional seja concluída;

- **Web Coverage Service (WCS)** – O serviço web no padrão WCS disponibiliza operações que permitem às aplicações recuperar dados geográficos do tipo cobertura. Uma cobertura consiste em uma representação espacial de uma área geográfica, que divide a mesma em partes (regulares ou irregulares), onde cada parte é associada a um valor. Por exemplo, uma imagem de satélite pode ser considerada uma cobertura regular. O WCS contém 3 operações: (i) *GetCapabilities*; (ii) *DescribeCoverage*; (iii) *GetCoverage*. Como nos outros padrões de serviços web do OGC, a operação *GetCapabilities* é responsável por retornar os metadados associados ao serviço web. A operação *DescribeCoverage* fornece detalhes acerca de uma *coverage* específica. Por fim, a operação *GetCoverage* retorna a *coverage* requerida;
- **Web Processing Service (WPS)** – O serviço web no padrão WPS disponibiliza interfaces que permitem acesso a processos geográficos. Um processo pode ser um algoritmo, cálculo ou modelo que opera sobre dados geográficos. O WPS é composto por três operações: (i) *GetCapabilities*; (ii) *DescribeProcess*; (iii) *Execute*. A operação *GetCapabilities* é responsável por fornecer os metadados referentes ao WPS. A operação *DescribeProcess* retorna informações detalhadas acerca dos processos que estão disponíveis no servidor WPS. Por fim, a operação *Execute* permite ao cliente solicitar que a operação seja executada;

- ***Catalog Service for the Web (CSW)*** – O serviço web conforme o padrão CSW disponibiliza interfaces que permitem acesso às operações para descobrir e consultar metadados relacionados aos serviços web.

Dos padrões de serviços web do OGC descritos anteriormente, são considerados neste trabalho: o WMS, WFS e WPS.

Embora muitos dados e modelos geográficos estejam sendo publicados na Web conforme os padrões do OGC, há ainda dificuldades para organizar e descobrir os recursos geográficos. Uma abordagem adotada tem sido a associação de descrição semântica aos recursos geográficos. Recentemente, a Web Semântica foi proposta, definindo os padrões e tecnologias que estão sendo empregadas na descrição dos recursos. Assim, os conceitos e tecnologias relacionados com a Web Semântica são apresentados na próxima seção.

2.2 Web Semântica

A Web consiste em documentos interligados, disponíveis através de uma rede de computadores, que permite aos usuários acessar tais documentos por meio de requisições do tipo *Hypertext Transfer Protocol – HTTP*. A disseminação da Web resultou em uma grande quantidade de documentos disponíveis na rede. Uma das principais dificuldades, diante dessa enorme quantidade de informações, consiste na localização e recuperação de informação de acordo com uma necessidade específica.

Nesse contexto complexo surge a proposta da Web Semântica (BENNERS-LEE; HANDLER; LASSILA, 2001). Na Web Semântica o objetivo principal é disponibilizar os dados na Web, associados a um conjunto de descrições em formato processável por máquina. A partir das descrições associadas aos dados, é possível que sistemas e agentes sejam capazes de processar o grande volume de informações disponíveis na Web e auxiliar os usuários em suas tarefas. Segundo o W3C (W3C SEMANTIC WEB ACTIVITY, 2001): “*a Web Semântica provê um framework que permite o compartilhamento e reutilização de dados entre aplicações, empresas e comunidades*”. O *framework* mencionado na definição anterior consiste na arquitetura apresentada na figura 1.

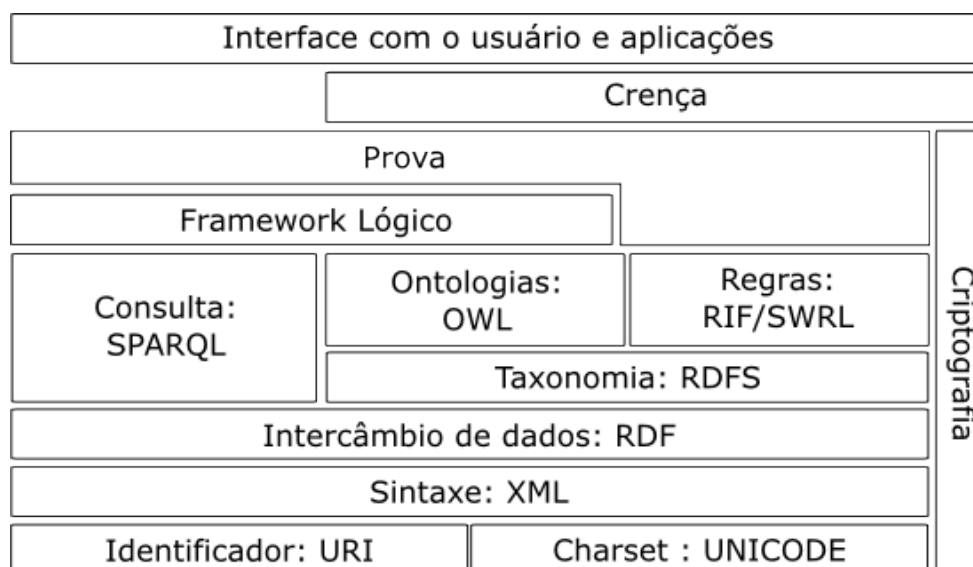


Figura 1. Arquitetura da Web Semântica.

Fonte: (OBITKO, 2007)

Na próxima seção são apresentadas em mais detalhes as tecnologias que compõem a Web Semântica e que foram utilizadas neste trabalho.

2.2.1 Tecnologias da Web Semântica

A arquitetura proposta para a Web Semântica, é baseada na infraestrutura da Web atual. Na Web atual os recursos são definidos como qualquer objeto concreto ou abstrato do mundo (uma pessoa, um livro, uma página Web, entre outros) que pode ser referenciado na Web por meio de um *Universal Resource Identifier – URI*. O URI consiste em uma identificação única atribuída a um recurso. Na Web é comum que um URI seja representado por uma *Universal Resource Locator – URL*. É comum haver confusão de entendimento acerca do URI e do URL, entretanto, suas definições são bastante claras: o URI é um meio de identificar unicamente um recurso; já o URL define a localização do recurso.

Considerando a arquitetura da Web Semântica, a camada seguinte é composta pela *eXtensible Markup Language – XML*, que consiste em uma linguagem e um modelo para transportar e armazenar dados e metadados na Web.

a) XML

O XML foi projetado para servir como um modelo padrão para estruturar, transportar e armazenar dados e metadados na Web. O XML permite que sejam criadas as marcações que são utilizadas para estruturar os dados. As características principais do XML são as seguintes:

- **Separa a descrição do dado e os aspectos de apresentação** – A adoção do XML permite separar a estrutura do dado, dos seus aspectos de apresentação. Essa característica facilita a troca de informações entre sistemas distintos;
- **Simplifica a troca de dados entre sistemas distintos** – Atualmente, há dificuldades em trocar dados entre sistemas distintos devido às diferenças de formato em que os dados são armazenados. O XML permite representar os dados como texto e, portanto, simplifica a troca de informações entre sistemas distintos;
- **XML pode ser utilizada para criar novas linguagens de marcação** – O XML permite que os usuários especifiquem suas próprias marcações. A partir dessa flexibilidade é possível que sejam criadas novas linguagens de marcação.

Um exemplo de um arquivo XML que representa um catálogo de livros é ilustrado na figura 2.

```

<bookstore>
  <book category="COOKING">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="CHILDREN">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="WEB">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>

```

Figura 2. Exemplo de um arquivo XML.

As características evidenciadas fazem do XML a opção para ser a base da Web Semântica. As camadas superiores, que consistem basicamente de linguagens para representação do conhecimento, são especificadas com bases no XML.

Considerando a arquitetura da Web Semântica, a próxima camada é composta pelo *Resource Description Framework – RDF*, que consiste em um *framework* para descrever recursos na Web.

b) RDF

O *Resource Description Framework – RDF*, construído sobre o XML, consiste em uma linguagem para representar informações sobre recursos na Web. Na Web Semântica, há informações associadas aos recursos. Tais informações tem o objetivo de descrever os recursos. As informações utilizadas para descrever os recursos são denominadas metadados. O RDF consiste em uma linguagem e um modelo para representar metadados na Web Semântica.

A adoção do RDF para representar metadados na Web Semântica acarreta uma série de implicações (W3C RDF PRIMER, 2004):

- **Adoção da estrutura de grafos para representar dados e metadados na Web** – O RDF adota um esquema de dados baseado em grafo, denominado “tripas”. As “tripas” são

utilizadas para expressar os metadados de um recurso e tem a seguinte estrutura: $T = \langle s, p, o \rangle$. Onde *s* representa o sujeito, ou seja, o recurso que é o alvo da afirmação representada na tripla. Já *p* representa o predicado associado ao sujeito e, por fim, *o* consiste no valor associado ao predicado do recurso, ou seja, o objeto;

- **Modelo padrão para troca de dados e metadados na Web** – O modelo de dados do RDF permite estabelecer um padrão a partir do qual metadados e dados são representados na Web. A adoção desse modelo habilita as aplicações a trocarem metadados e dados em escala global por meio da Web;
- **Integração de dados independente do esquema** – Por se tratar de uma camada sobre os dados, o RDF permite que dados e metadados sejam integrados de forma independente do esquema que é utilizado para modelar os dados;
- **Suporta evolução dos esquemas** – O esquema utilizado na camada de dados pode evoluir sem, no entanto, impactar na forma como as aplicações trocam dados e metadados através do RDF;
- **Suporta processamento de máquina** – O RDF foi projetado para ser processado por máquina.

Como exemplo de utilização do RDF, considera-se as seguintes assertivas acerca de um recurso na Web (W3C RDF PRIMER, 2004):

(1) *<http://www.exemplo.org/index.html> tem uma **data_criação** cujo valor é **Agosto 16, 2010***

(2) *<http://www.exemplo.org/index.html> tem uma **idioma** cujo valor é **Português***

Ambas as asserções associam uma propriedade e um valor a um recurso na Web. Em outras palavras, as asserções associam um par predicado-objeto a um sujeito. No modelo de triplas adotado pelo RDF, as afirmações anteriores ficam como:

(1) Sujeito: *<http://www.exemplo.org/index.html>*

Predicado: **data_criação**

Objeto: **Agosto 16, 2010**

(2) Sujeito: *<http://www.exemplo.org/index.html>*

Predicado: **idioma**

Objeto: **Português**

Na figura 3, as afirmações acima são representadas na forma de grafo, onde os nós denotam sujeitos e objetos, enquanto as arestas definem as relações ou predicados.

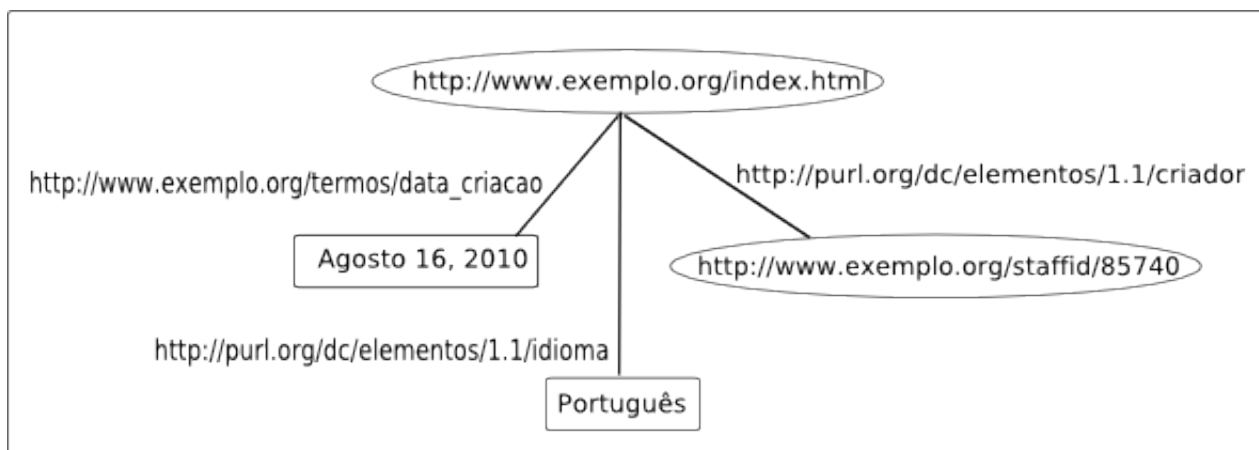


Figura 3. Representação em grafo das triplas RDF.

Fonte: W3C RDF PRIMER (2004) .

O RDF tem sido adotado para representar dados e metadados na Web Semântica. Entretanto, seu poder de expressividade é limitado. Como exemplo de suas limitações: não é possível representar hierarquias em RDF; não é possível representar propriedades lógicas como disjunção, simetria, entre outras. Por conta das limitações expostas, foi proposta uma extensão ao RDF: a *Web Ontology Language – OWL*.

c) Ontologias e OWL

Ontologias são artefatos fundamentais no contexto da Web Semântica. Uma definição é a proposta por Gruber (1993): “*Uma ontologia é uma especificação de uma conceitualização*”. A atualização da definição de Gruber (1993) é a seguinte (GRUBER, 2009): “*Uma ontologia denota um conjunto de primitivas representacionais com as quais é possível modelar um domínio do conhecimento ou um universo de discurso*”. Ainda segundo Gruber (2009), as primitivas representacionais são constituídas por classes, atributos e relacionamentos.

Há muitas linguagens que podem ser utilizadas para codificar uma ontologia. No contexto da Web Semântica, as ontologias tem sido implementadas com a *Web Ontology Language – OWL*.

Uma ontologia especificada em OWL consiste no seguinte conjunto $O = \{C, A, I, R\}$, onde:

- **C** – É o conjunto de classes utilizadas na conceitualização do domínio;
- **A** – São os atributos ou propriedades utilizadas para definir as classes e os relacionamentos

entre classes e indivíduos;

- ***I*** – Compreendem as instâncias do domínio em questão;
- ***R*** – Consiste nas restrições empregadas para modelar o domínio.

As Ontologias tem sido empregadas para permitir que sistemas possam interoperar em nível semântico. Uma das abordagens tem sido a descrição de serviços web por meio de conceitos formalmente definidos em uma ontologia. Os serviços web descritos com conceitos de ontologias são denominados serviços web semânticos. As duas principais abordagens nesse sentido são apresentadas na seção 2.3.

Considerando que o OWL é o modelo formal para representar e descrever universos de discurso na Web Semântica, é natural imaginar que seja necessário um meio para consultar e recuperar as informações contidas na representação OWL. Como o OWL é uma extensão ao RDF e, portanto, compartilha seu modelo de dados baseado em triplas, um dos mecanismos que pode ser utilizado para extrair informações de um modelo OWL é o *SPARQL – Protocol and Query Language for RDF*. O SPARQL é apresentado na próxima seção.

d) SPARQL

O SPARQL consiste em um protocolo e uma linguagem de consulta para dados expressos segundo o padrão RDF. O protocolo SPARQL enfatiza o formato das mensagens que devem representar as consultas especificadas e os resultados obtidos mediante o processamento da consulta. Já a linguagem SPARQL enfatiza a sintaxe empregada na construção das consultas.

Uma consulta em SPARQL consiste em padrões de triplas seguindo a mesma estrutura básica do RDF, isto é: Sujeito-Predicado-Objeto. A questão central nessa definição é que cada parte da tripla pode ser substituído por uma variável. Os padrões de triplas são verificados em conjuntos de descrições RDF e caso haja alguma tripla que satisfaça o padrão expresso na consulta SPARQL, essa tripla é incluída no resultado.

Formalmente, uma consulta SPARQL é representada pela tupla $\langle GP, DS, SM, R \rangle$ (SBODIO; MOULIN, 2007) onde:

- ***GP (Graph Pattern)*** – é um padrão na forma de “tripla”. Em outras palavras: um padrão em um grafo;
- ***DS (DataSet)*** – é o repositório RDF que é alvo da consulta;

- **SM (Solution Modifiers)** – São filtros aplicados ao conjunto solução que permitem alterar a resposta;
- **R (Result)** – São os resultados da consulta.

Com relação aos resultados, estes podem ter formas variadas dependendo do tipo de resultado requerido na consulta. O SPARQL permite três tipos de resultado: *SELECT*, *CONSTRUCT* e *ASK*. O *SELECT* retorna os resultados do repositório RDF que são idênticos com os padrões especificados na consulta. Já o *CONSTRUCT* permite transformar os resultados com base em um modelo pré-estabelecido, que consiste em uma série de padrões de triplas que são aplicados aos resultados de modo a modificar sua estrutura. Por fim, o *ASK* provê um meio de verificar se há uma solução para determinado padrão de tripla em um repositório RDF.

As tecnologias e conceitos da Web Semântica são empregados para descrever aspectos de serviços web. Assim, na próxima seção são apresentadas as duas principais propostas para descrição semântica de serviços web.

2.3 Serviços Web Semânticos

Os Serviços web semânticos são serviços web que tem suas capacidades descritas por conceitos definidos em uma ontologia. A adoção da semântica na descrição dos serviços web visa permitir a automação das tarefas como o descobrimento, composição e execução dos serviços web. Na sequência desta seção são descritas as duas principais iniciativas para serviços web semânticos: *Web Service Modelling Ontology* – *WSMO* (DOMINGUE; ROMAN; STOLLBERG, 2005) e o *Semantic Markup for Web Services* – *OWL-S* (OWL-S, 2004).

2.3.1 WSMO – Web Service Modelling Ontology

O WSMO consiste em um modelo conceitual para a realização dos serviços web semânticos. A definição do WSMO e seus componentes é baseada no *Web Service Modeling Framework* – *WSMF* (FENSEL; BUSSLER, 2002). O WSMF se trata de um *framework* que descreve os aspectos relevantes para a realização de serviços web semânticos. Junto a esta

definição, há uma proposta de projeto baseada em ontologias, objetivos, serviços web e mediadores, que garante incrementos no desacoplamento e na escalabilidade da arquitetura. O WSMF define seus componentes principais conforme segue (FENSEL; BUSSLER, 2002):

- **Ontologias** – Ontologias são apresentadas como a tecnologia chave para a realização da Web Semântica. No contexto dos serviços web semânticos, as ontologias são empregadas para comunicar uma representação formal de um domínio compartilhado, e permitir associar esta representação aos serviços web;
- **Objetivos** – Os serviços web devem estar associados a uma representação formal de Objetivo. Um objetivo é entendido como algo que o consumidor deseja obter a partir da execução de uma tarefa. A descrição de um objetivo sempre é feita em função de conceitos de uma ou mais ontologias;
- **Serviços Web** – No contexto do WSMF, os serviços web são definidos em função de sua utilidade: são ferramentas que auxiliam a alcançar determinados objetivos e são acessíveis via Web;
- **Mediadores** – No contexto do WSMF, os Mediadores tratam problemas de interoperabilidade entre os componentes do *framework*. Os diversos tipos de heterogeneidade e mais detalhes sobre os mediadores, podem ser encontrados no trabalho de Fensel e Bussler (2002).

O WSMO estende as definições do WSMF no sentido de proporcionar uma base mais próxima à uma solução técnica. O projeto do WSMO é centrado nos seguintes princípios (DOMINGUE; ROMAN; STOLLBERG, 2005):

- **Conformidade com a Web** – WSMO adota o padrão Universal Resource Identifier – URI para identificar seus recursos. Além disso, suporta namespaces e padrões baseados em eXtensible Markup Language – XML;
- **Baseado em ontologias** – As ontologias são a base para a descrição de todos os modelos de dados presentes no WSMO, considerando tanto os modelos que descrevem recursos como aqueles que descrevem as interações;
- **Desacoplamento** – O desacoplamento significa que cada recurso é descrito

independentemente da forma como será utilizado;

- **Mediação** - Este princípio se trata de um incremento ao desacoplamento. Os mediadores devem tratar as heterogeneidades entre os componentes;
- **Identificação e separação de papéis para as ontologias** – Há separação entre a perspectiva do consumidor e a perspectiva do provedor do recurso;
- **Descrição versus Implementação** – O WSMO faz distinção entre a formalização/descrição dos aspectos teóricos relacionados aos serviços web semânticos e as tecnologias relacionadas à implementação. O modelo conceitual deve possibilitar a implementação perante as tecnologias emergentes da Web Semântica;
- **Verificação da especificação conceitual** – Uma implementação de referência da especificação é utilizada como meio de avaliar a especificação conceitual. Nesse sentido, a implementação de referência do WSMO é o *Web Service Execution Environment - WSMX* (HEROLD, 2008).

Para habilitar o descobrimento, composição e execução de recursos, o WSMO fornece a definição de cinco componentes utilizados na descrição dos serviços web (DOMINGUE; ROMAN; STOLLBERG, 2005):

- **Propriedades não-funcionais** – Propriedades não relacionadas ao procedimento executado pelo serviço web;
- **Ontologias importadas** – Ontologias importadas pelo serviço web para descrever suas propriedades, objetivos, interface, mediadores, entre outros;
- **Mediadores utilizados** – São os mediadores utilizados pelo serviço web;
- **Capacidades** – Define a funcionalidade do serviço web em termos de suas pré-condições, pós-condições, propriedades não-funcionais, efeitos e pré-supostos. Tais aspectos são expressos a partir de conceitos de ontologias importadas para descrever o serviço web;
- **Interfaces** – Descreve como a funcionalidade do serviço web pode ser invocada. Estas descrições referem-se ao padrão de comunicação utilizado por um serviço web (coreografia) e à descrição das interações realizadas pelo serviço web com outros serviços web, para atingir seus objetivos (orquestração).

Definidos os aspectos conceituais gerais utilizados pelo WSMO é importante definir como estes conceitos são abordados na implementação de referência WSMX. No próximo item é descrito o WSMX.

a) WSMX – Web Service Execution Environment

O WSMX é a implementação de referência para o modelo conceitual WSMO. Trata-se de um ambiente de execução com foco na integração de serviços web a partir de sua descrição semântica.

O WSMX é composto pelos seguintes componentes (HEROLD, 2008):

- **Core** – Contempla funcionalidades básicas e de suporte como: carregar os componentes, controlar as mensagens trocadas entre os componentes, além de definir os caminhos de execução. Funciona como um controlador e integrador dos outros componentes da arquitetura;
- **Coreography** – Resolve problemas de heterogeneidade entre o consumidor e o provedor de serviços web. As diferenças podem estar relacionadas à ordem e à granularidade das operações;
- **Communication Manager** – Trata-se do ponto de entrada para acesso externo aos recursos do WSMX;
- **Data Mediator** – Trata problemas de heterogeneidade entre ontologias. Nos processos realizados pelo WSMX (descobrimto, composição, execução) podem ocorrer diferenças entre as ontologias que são empregadas para descrever os recursos. Nesse contexto, é necessário um componente que realize mapeamentos entre os conceitos;
- **Invoker** – Trata as diferenças de comunicação existentes entre o WSMX e os serviços web baseados em *Simple Object Access Protocol – SOAP*. O SOAP consiste em um protocolo de troca de mensagens entre serviços web. É o protocolo padrão definido pelo W3C. Para obter os resultados de uma funcionalidade proveniente de um serviço web no WSMX, é necessário invocar a implementação externa do serviço web. Na implementação atual, o WSMX apenas suporta o mapeamento WSMX-SOAP. No entanto, há uma nota de desenvolvimento apontando para o suporte aos serviços web REST (*REpresentational State Transfer*);
- **Orchestration** – Responsável por resolver problemas de heterogeneidade no processo de execução de uma tarefa. Sua função é definir como um serviço web deve invocar outros serviços web para conseguir atingir seus objetivos. Este componente é utilizado apenas para testes na versão atual do WSMX;
- **Parser** – Processa os arquivos no formato *Web Service Modeling Language – WSML* e os converte para uma representação em memória. No WSMX, todas as ontologias são

representadas utilizando o WSML;

- **Resource Manager** – Responsável por prover funcionalidades de suporte ao armazenamento persistente;
- **Web Service Discovery** – Componente responsável por descobrir serviços web com base em um objetivo explícito.

2.3.2 OWL-S – Semantic Markup for Web Services

O OWL-S se trata de uma definição conceitual acerca de elementos que devem ser utilizados na descrição de um serviço web. Sua proposta se baseia na definição de conceitos e perspectivas que devem ser formalmente representados para permitir a descoberta, composição e execução de serviços web. A definição dos conceitos é realizada através de ontologias codificadas no padrão OWL (OWL-S, 2004).

As perspectivas utilizadas para descrever os serviços web são baseadas em (OWL-S, 2004):

- **Funcionalidade que o serviço web é capaz de realizar** - Perspectiva funcional e não-funcional que determina o que o serviço web é capaz de realizar, qual a qualidade com que ele realiza e quem é o responsável pelo serviço web. Esta perspectiva serve à tarefa de descobrimento de serviços web;
- **Modo pelo qual o serviço web realiza a atividade** - Perspectiva relacionada à utilização do serviço web. Descreve as entradas do serviço web, resultados, qual a ordem de invocação de suas operações para obter determinados resultados, entre outros. Esta perspectiva relaciona-se às etapas de composição e execução dos serviços web;
- **Modo pelo qual o serviço web é acessado** - Perspectiva relacionada ao acesso à funcionalidade. Descreve a utilização de protocolos, formato de mensagens, entre outros aspectos, todos relacionados à invocação do serviço web.

Cada perspectiva de descrição é implementada como uma ontologia (OWL-S, 2004):

- **ServiceProfile** – Descreve um serviço web em função do seu fornecedor, sua computação e

uma lista de características. A computação realizada por um serviço é representada por meio de suas entradas, resultados, pré-condições e pós-condições. A lista de características fornece descrição sobre a qualidade e popularidade do serviço web, classificação do serviço web (de acordo com algum padrão de classificação) e atributos genéricos que podem descrever qualquer propriedade;

- ***ServiceModel*** – Descreve o serviço web a partir da perspectiva de um processo. Conforme descrito em, um processo não é um programa para ser executado, mas uma descrição das maneiras como um consumidor pode interagir com o serviço. A descrição é realizada em função das suas entradas, saídas, pré-condições e efeitos;
- ***ServiceGrounding*** – Descreve os aspectos de implementação do serviço, como protocolos, formatos de mensagens, entre outros.

2.4 Conclusões do capítulo

Neste capítulo foram apresentados os conceitos básicos que são importantes para o entendimento do trabalho. Nesse sentido, foram abordados os conceitos relacionados com serviços web e, especificamente, com serviços web para dados geográficos. Além disso, foram apresentados os conceitos básicos da Web Semântica. Por fim, foi descrito o conceito de serviço web semântico, que utiliza conceitos das definições de serviços web e da web semântica. Na próxima seção são apresentados os trabalhos correlacionados com a Pangea.

Capítulo 3

Aspectos transversais às abordagens semânticas

Neste capítulo são apresentados os aspectos transversais identificados em trabalhos correlacionados com o projeto dos módulos da arquitetura Pangea. Em geral, esses trabalhos abordam questões relacionadas com o descobrimento e integração de fontes de dados geográficos. Frequentemente, os mecanismos de descobrimento e integração adotados nesses trabalhos são baseados em tecnologias da Web Semântica e em serviços web geográficos. No contexto geográfico, a junção da Web Semântica e de Serviços Geográficos tem sido denominada *Geospatial Semantic Web* (EGENHOFER, 2002). Portanto, muitos dos trabalhos discutidos nesse capítulo fazem referência direta ou indireta à *Geospatial Semantic Web*.

Os aspectos discutidos nesse capítulo são empregados, fundamentalmente, para suportar os requisitos de interoperabilidade sintática e semântica de dados e modelos geográficos entre consumidores e provedores de informação. Os aspectos transversais apresentados são utilizados na definição dos requisitos que devem ser suportados pelos módulos da arquitetura Pangea (apresentados no Capítulo 4).

3.1 Arquiteturas para o compartilhamento de dados e modelos geográficos

Nesta seção são apresentados e discutidos trabalhos que descrevem abordagens arquiteturais e tecnológicas utilizadas na anotação, descobrimento, recuperação e integração de dados e modelos geográficos. O objetivo principal do estudo dessas abordagens é identificar os requisitos que suportam e os componentes arquiteturais empregados. Esses aspectos são utilizados de modo a fornecer as bases para o levantamento dos requisitos que devem ser suportados pelo projeto dos módulos da arquitetura Pangea.

3.1.1 Contextualização e problemática

Com a disseminação da Web e dos padrões de serviços OGC grandes quantidades de dados e modelos geográficos estão sendo publicados na Web. Apesar de estarem disponíveis na Web, há dificuldades relacionadas com o descobrimento e consumo dessas informações e modelos.

Nesse sentido, tem sido propostas infraestruturas cujo foco é a anotação, o descobrimento e a integração de dados e modelos geográficos na Web.

3.1.2 Trabalhos e conceitos relacionados

Nesta seção são apresentados os principais trabalhos desenvolvidos que apresentam aspectos arquiteturais e tecnológicos relacionados com a proposta da arquitetura Pangea. Foram selecionadas três arquiteturas para o estudo: (i) *DAGIS – Discovering Annotated Geospatial Information Service*; (ii) *SWING – Semantic Web Services Interoperability for Geospatial Domain*; (iii) *WeBIOS - Web Service Multimodal Tools for Strategic Biodiversity Research, Assessment and Monitoring*.

3.1.2.1 DAGIS – *Discovering Annotated Geospatial Information Service*

O foco principal do DAGIS (ALAM; SUBBIAH; KHAN; THURASINGHAM, 2007) é proporcionar um mecanismo para o descobrimento e integração de fontes de dados distribuídas, requerendo a menor quantidade possível de intervenção humana. Para tanto, as etapas de descobrimento e integração das fontes de dados são baseadas tanto em lógica, com base em uma ontologia de domínio, como em métricas de recuperação de informação.

A arquitetura geral do DAGIS, apresentada na figura 4, consiste em três camadas principais: apresentação, *middleware* semântico e repositório de ontologias.

Na camada de apresentação há o *DAGIS Query Browser*, que consiste do portal onde o usuário pode especificar suas consultas. Com base nas ontologias de domínio as ambiguidades

existentes na consulta são eliminadas e posteriormente transformadas em um OWL-S *Service Profile*. O *ServiceProfile* gerado representa a consulta e é utilizado nos procedimentos das outras camadas.

Na camada *middleware* semântico há os seguintes componentes:

- i. **DAGIS Agent** – Responsável por receber e transformar a consulta especificada pelo usuário. A consulta é transformada para o padrão OWL-S *Service Profile*, e é submetido ao DAGIS *Matchmaker*;
- ii. **DAGIS Composer** – Componente responsável por decompor a consulta do usuário e criar uma composição de serviços web que seja capaz de resolver a consulta. É utilizado nos casos onde um serviço web atômico não pode resolver a consulta;
- iii. **DAGIS Matchmaker** – Componente responsável por verificar a correspondência entre a consulta do usuário e as descrições dos serviços web armazenadas nos registros. A similaridade entre a consulta e as descrições dos serviços web é obtida a partir da análise dos aspectos funcionais e não-funcionais. Além disso, o mecanismo de *match* é híbrido, ou seja, é baseado em dedução lógica e em técnicas de recuperação de informação;
- iv. **WSDL2OWL-S Converter** – Converte as descrições de serviços web WSDL para a descrição semântica em OWL-S.

No repositório de ontologias há os seguintes componentes:

- i. **OWL-S Ontology** – Conjunto de conceitos que formam o *framework* de descrição de serviços web OWL-S;
- ii. **QoS Ontology** – Conceitos utilizados na classificação dos serviços web de acordo com aspectos não funcionais;
- iii. **Geospatial Domain Ontology** – Conceitos geográficos utilizados na anotação dos serviços web.

Por fim, considerando os registros utilizados para descobrir os serviços web semânticos, há os seguintes componentes:

- i. **OWL-S Registry** – Catálogo semântico contendo as descrições dos serviços web OWL-S disponíveis;
- ii. **WSDL Registry** – Catálogo baseado em UDDI ou qualquer tipo de catálogo público.

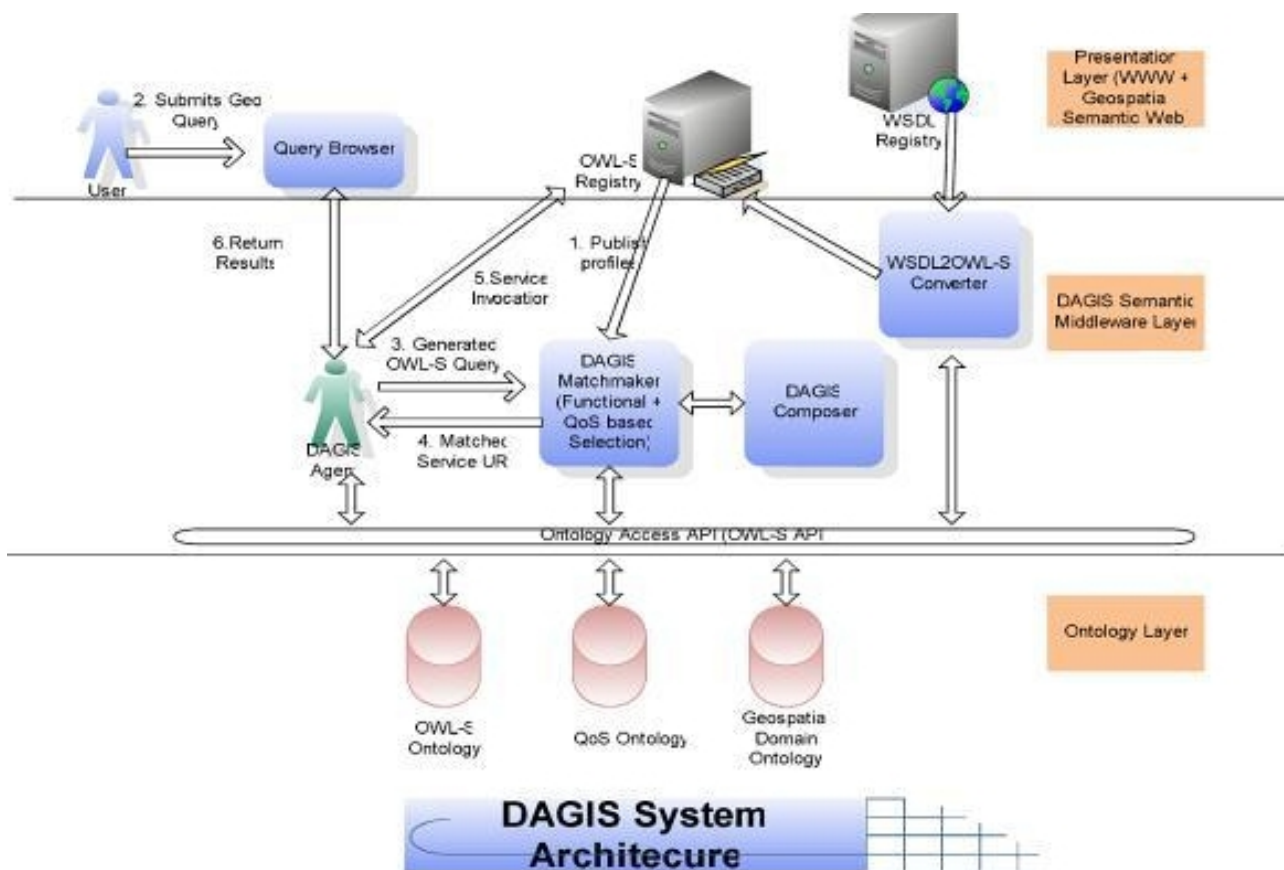


Figura 4. Esquema geral da arquitetura do DAGIS.

Fonte: extraído de Alam, Subbiah, Khan e Thuraisingham (2007).

No DAGIS as descrições dos serviços web são disponibilizadas em registros (passo 1 na figura 4). As descrições são utilizadas no procedimento de resolução de consultas. O procedimento de consulta tem início quando um usuário especifica uma consulta (passo 2 na figura 4) e a submete ao *Query Browser*. O *Query Browser* remete a consulta ao DAGIS Agent, responsável por criar uma representação em OWL-S da consulta (passo 3 na figura 4). A consulta representada em OWL-S é processada pelo DAGIS Matchmaker e os resultados são retornados ao DAGIS Agent (passo 4 na figura 4). Os serviços web que satisfazem a consulta são invocados (passo 5 da figura 4) e os resultados são retornados ao usuário (passo 6 da figura 4).

No que concerne ao procedimento de processamento de consultas e descobrimento de serviços web, o DAGIS permite algumas melhorias em relação às abordagens não semânticas. Durante o processamento da consulta é possível desambiguar os termos em linguagem natural utilizando a ontologia de domínio. No processo de desambiguação são identificados os operadores espaciais, permitindo que a consulta seja processada levando em consideração a semântica da operação espacial.

Considerando o descobrimento de serviços web, o DAGIS permite resultados mais

precisos em decorrência de dois aspectos de projeto: (i) adoção do *framework* semântico de descrição de serviços web OWL-S; (ii) Utilização de um mecanismo de *match* híbrido (i.e. Emprega dedução lógica e métricas de recuperação de informação), baseado em aspectos funcionais e não-funcionais.

A adoção do aspecto (i) no projeto do DAGIS, incorpora funcionalidades para a descrição dos serviços web de acordo com suas entradas, saídas, pré-condições e efeitos (i.e. *Inputs, Outputs, Pre-conditions, Effects – IOPE*). Além disso, é possível descrever o modelo de processamento executado pelo serviço web. Com isso, o mecanismo de *match* tem mais informações para proceder o cálculo da similaridade entre a consulta e a descrição dos serviços web. Isso acaba por resultar em mecanismos mais precisos para o descobrimento de serviços web que “casam” com a consulta do usuário.

O aspecto (ii) está relacionado com as métricas utilizadas para descobrir os serviços web que satisfazem uma consulta. No DAGIS as métricas são híbridas, baseadas tanto em dedução lógica, como em métricas de recuperação de informação. Para determinar a relevância dos serviços web descobertos são utilizados aspectos não funcionais modelados como *Quality of Service – QoS*.

Além dos aspectos e melhorias evidenciados anteriormente, o DAGIS possibilita a decomposição da consulta do usuário de modo a compor uma cadeia de serviços web que satisfaça a consulta. Esse mecanismo de composição é invocado quando não é retornado um serviço web atômico que satisfaça a consulta do usuário.

3.1.2.2 SWING – *Semantic Web Services Interoperability for Geospatial Domain*

O SWING (ROMAN; KLIEN; SKOGAN, 2006) consiste em uma iniciativa da União Européia para a construção de um *framework* que suporte a associação de semântica a serviços web do domínio geográfico. O foco principal consiste em disponibilizar ferramentas e métodos que auxiliam os usuários na tarefa de anotar, descobrir, compor e recuperar serviços web de informação geográfica. Atualmente, o SWING é empregado na gestão de recursos naturais, em especial minerais, na União Européia.

Para suportar os requisitos mencionados anteriormente, sua arquitetura, apresentada na figura 5, é composta de sete componentes:

- i. **MiMS (Mineral resources Management System)** - aplicativo utilizado pelos especialistas para descobrir e anotar serviços web geográficos;
- ii. **DEV** - utilitário de modelagem UML que auxilia o especialista na composição de serviços web;
- iii. **CAT** - catálogo baseado no padrão OGC CS/W utilizado para armazenar os metadados utilizados na descrição dos serviços web;
- iv. **Query Annot** - responsável por auxiliar na anotação semântica das consultas do especialista;
- v. **Service Annot** - responsável por auxiliar na anotação dos serviços web geográficos;
- vi. **ANNOT** - responsável pelo mapeamento de uma descrição em linguagem natural para uma descrição semântica;
- vii. **WSMX (Web Service Modelling eXecution environment)** - ambiente utilizado no descobrimento semântico de serviços web. O ambiente WSMX é uma implementação de referência do modelo conceitual WSMO – *Web Service Modeling Ontology*. O WSMO por sua vez é baseado no WSMF - *Web Service Modeling Framework* (FENSEL; BUSSLER, 2002).

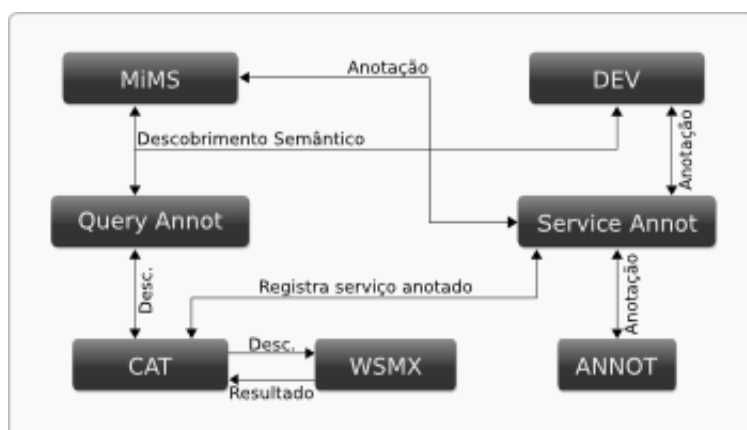


Figura 5. Componentes do SWING.

Fonte: Adaptado de ROMAN; KLIEN; SKOGAN (2006) .

O MiMS é o aplicativo (*front end*) utilizado pelos especialistas para descobrir, anotar e publicar serviços web geográficos. O descobrimento de serviços web é realizado a partir da

elaboração de uma consulta semântica. Uma consulta semântica é especificada em função de conceitos provenientes das ontologias de domínio disponíveis ao especialista.

Uma vez que a consulta semântica tenha sido especificada em função dos conceitos do domínio, ela é enviada ao componente CAT que remete a parte semântica da consulta para ser resolvida no componente WSMX. No ambiente WSMX os serviços web que satisfazem a consulta são selecionados e retornados ao componente CAT que retorna o conjunto de serviços web ao *Query Annot*. Por fim os resultados são enviados ao MiMS e apresentados ao especialista.

O especialista também pode anotar serviços web com conceitos provenientes de suas ontologias de domínio e publicar novamente no SWING. Para anotar um serviço web o especialista interage através do MiMS com os componentes *Service Annot* e ANNOT. A anotação de um serviço web consiste em mapear conceitos do domínio para o esquema de dados do serviço web. Uma vez realizado o mapeamento entre os conceitos do domínio e o esquema de dados do serviço web, esta descrição é publicada no catálogo e está pronta para ser descoberta e reutilizada.

3.1.2.3 WeBIOS – Web Service Multimodal Tools for Strategic Biodiversity Research, Assessment and Monitoring

O WeBIOS propõe um ambiente computacional flexível, orientado a serviços, que permite consultas multimodais a fontes heterogêneas de dados. Este ambiente serve como infraestrutura para a integração de dados no contexto das pesquisas em biodiversidade (GOMES JR., 2007) (DALRIO, 2007). Assim como as arquiteturas apresentadas nas seções 3.1.2.1 e 3.1.2.2, a proposta arquitetural do WeBios é fortemente baseada em serviços web, nos padrões OGC de serviços e nos conceitos e tecnologias da Web Semântica.

A arquitetura do WeBIOS, apresentada na figura 6, é composta de quatro camadas principais: Camada de Armazenamento, Camada de Suporte, Camada de Serviços Avançados e Camada Cliente.

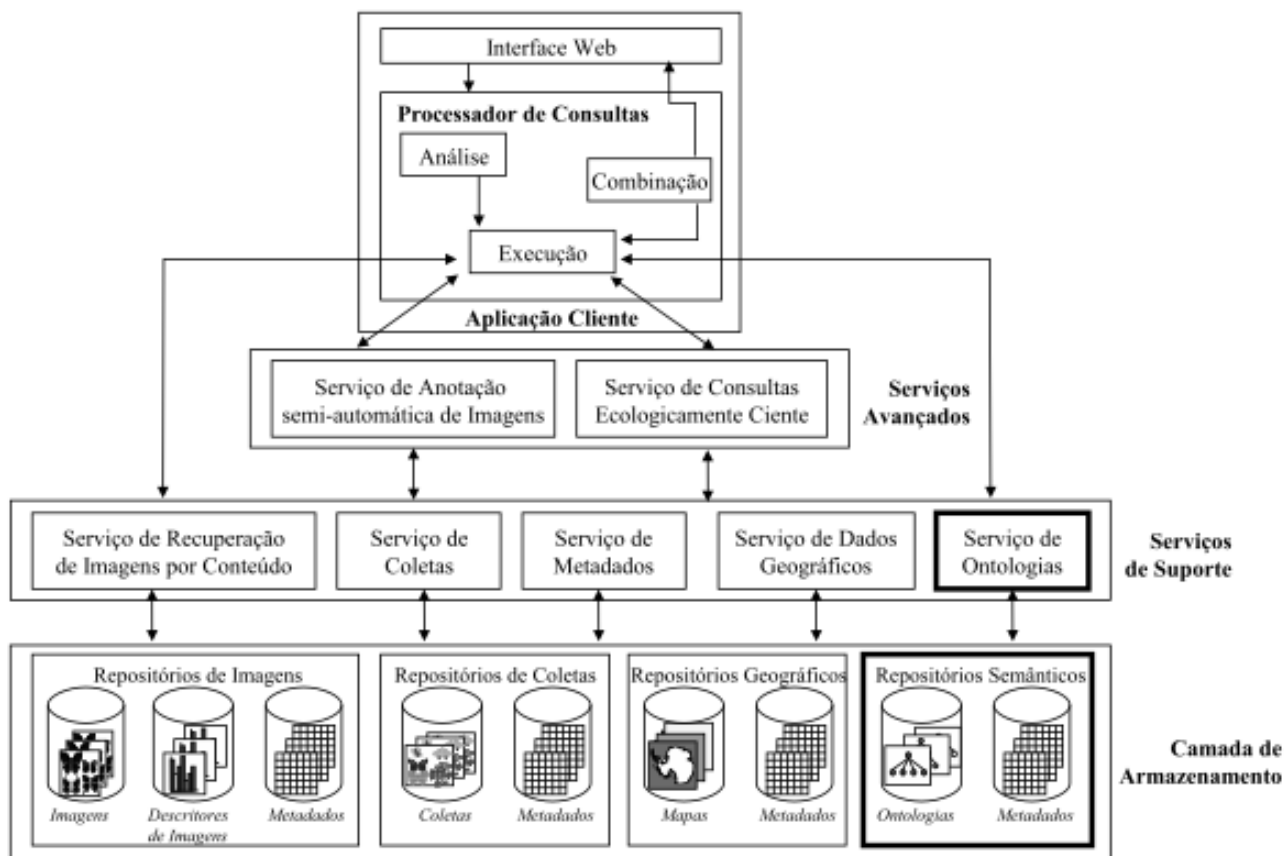


Figura 6. Arquitetura do sistema WeBIOS.

Fonte: Extraído de Daltio (2007).

As camadas do sistema WeBIOS, apresentadas na figura 6, são descritas abaixo (GOMES JR., 2007):

- i. **Camada de Armazenamento** - Compreende os repositórios de dados, metadados e ontologias. No contexto do WeBIOS há cinco tipos de dados: imagens, dados de coletas, informações georeferenciadas, metadados e ontologias.
- ii. **Camada de Suporte** - Trata-se da camada que agrupa os serviços web que realizam o acesso aos repositórios de dados e metadados. Divide-se em: serviços web para recuperação de imagens por conteúdo, serviços web de metadados, serviços web de coleções, serviços web de informações georeferenciadas e serviços web de ontologias. Cada categoria de serviços web de suporte permite acesso a uma classe de dados;
- iii. **Camada de Serviços web Avançados** – Nos casos onde a consulta do usuário demandar a combinação de vários tipos de dados (dados geográficos e registros de coletas como apresentado no trabalho de Gomes Jr. (2007)), os serviços web avançados são utilizados. Para acessar os repositórios de dados, os serviços web nessa camada utilizam os serviços

web da camada de suporte;

- iv. **Camada Cliente** - compreende as aplicações utilizadas pelos pesquisadores e estudantes para acesso aos repositórios heterogêneos de dados e metadados. Fundamentalmente, o acesso aos repositórios de dados se dá por meio da especificação e processamento de consultas. As consultas são especificadas em uma interface e enviadas a um componente Mediador. O Mediador é o responsável por traduzir a consulta do usuário para um conjunto de requisições aos serviços web nas camadas de suporte e de serviços web avançados. Os serviços web de suporte e avançados são os responsáveis pelo acesso aos dados na camada de armazenamento. Os dados então são retornados ao Mediador que compõe os resultados e apresenta ao usuário na interface.

Para ilustrar a interação entre os componentes da arquitetura, considera-se o exemplo apresentado por Gomes Jr. (2007): *“retorne todos os registros de ocorrência de espécies que são predadas pela mosca Adaina Bipunctata e que foram encontradas no interior da Mata Atlântica Paulista”*.

Considerando a resolução desta consulta, a interação entre os componentes se dá como segue: a consulta, anteriormente expressa em linguagem natural, é representada em SPARQL e repassada ao componente Mediador de Consultas. A consulta em SPARQL é representada como um grafo antes de iniciar o processamento da consulta. Após, o Mediador consulta o Catálogo de Ontologias com a finalidade de descobrir quais as espécies predadas pela mosca *Adaina Bipunctata*. Após o descobrimento das espécies predadas, o Mediador recorre ao Catálogo de Repositórios para descobrir quais os repositórios que possuem dados sobre as espécies de interesse. É necessário ainda aplicar as restrições geográficas decorrentes da localização “Mata Atlântica Paulista”. Os repositórios selecionados para as espécies, são então filtrados de acordo com um parâmetro espacial que determina a área de interesse.

Neste ponto, caso a consulta demande dados de mais de um tipo, o Mediador remete as requisições aos serviços web da Camada Avançados. Caso contrário, o mediador acessa diretamente serviços web na Camada de Suporte. No exemplo, a consulta envolve a composição de dados geográficos, registros de coletas e, eventualmente, imagens. Nesses casos as requisições são repassadas à Camada de Serviços web Avançados. Após, as requisições são submetidas para os serviços web da Camada de Suporte que acessam os repositórios e retornam os dados. Os dados retornados são compostos pelo Mediador de Consultas e apresentados ao usuário.

Considerando a descrição exposta, há muitos pontos interessantes com relação à arquitetura do WeBios:

- i. **Tipos de dados e serviços web de dados bem definidos** - A Camada de Armazenamento opera com tipos de dados e serviços web bem definidos. Essa característica permite que sejam definidos protocolos (serviços web) para acesso aos repositórios de cada tipo de dado. Essas definições garantem que a aplicação na Camada Cliente não tenha dificuldades em invocar cada tipo de serviço web, já que as interfaces para cada categoria de serviço web são bem definidas e conhecidas pela Camada Cliente.
- ii. **Adoção de ontologias** – As ontologias estabelecem a relação entre os conceitos de domínio e os repositórios de dados. No WeBIOS as ontologias estão presentes nos catálogos internos ao Mediador de Consultas e na Camada de Armazenamento, por meio dos repositórios de ontologias. No WeBios as ontologias são utilizadas para explicitar a semântica dos repositórios de dados e também para estabelecer e estender um esquema de consultas. Essa camada permite aos usuários expressar consultas multimodias, que envolvam predicados de várias naturezas (taxonômicos, ecológicos, geográficos, entre outros), além de funcionar como um “índice” para os repositórios de dados;
- iii. **Separação entre serviços web para um e vários tipos de dados** – Há uma separação conceitual entre serviços web simples (que acessam um tipo de dado) e serviços web complexos (que acessam vários tipos de dados). Os serviços web complexos envolvem a composição de mais de um tipo de dado, e requerem cuidados adicionais, relacionados à composição e execução. A composição e execução dos serviços web são realizadas de acordo com os tipos de dados requeridos por determinada consulta e sua lógica está implícita no Mediador de Consultas;
- iv. **Adoção de padrões de metadados** – O WeBIOS adota o *Darwin Core* (2002) como padrão para a marcação de registros de ocorrência de espécies. Esse padrão de metadados é amplamente adotado na comunidade de biodiversidade. A adoção de um padrão de metadados amplifica as possibilidades de compartilhamento dos dados;
- v. **Camada de serviços web de suporte** – As heterogeneidades sintáticas para acesso aos repositórios são resolvidas pela adoção de serviços web que permitem acesso aos repositórios de dados. Esta característica permite menor acoplamento entre os componentes da arquitetura e os dados;
- vi. **Adoção dos padrões OGC WFS e WMS** – O WeBIOS adota os padrões de serviços OGC WMS e WFS para disponibilizar fontes de dados geográficos. Tais padrões são amplamente adotados pela comunidade e incrementam as possibilidades de compartilhamento, integração e reutilização dos repositórios, além de garantir baixo acoplamento entre a Camada Cliente e os repositórios;

vii. **Adoção de catálogos** – No WeBIOS cada aplicação cliente possui um catálogo que relaciona os conceitos de domínio aos provedores de dados. A adoção de um catálogo permite desacoplar ainda mais os clientes dos recursos de dados. Entretanto, um catálogo adiciona preocupações e responsabilidades relacionadas ao procedimento de atualização dos catálogos, de anotação semântica dos recursos e de buscas distribuídas por diversos catálogos. No caso do WeBIOS, conforme já mencionado, o catálogo possibilita buscas semânticas baseadas em predicados de classes variadas, e não há experimentos que evidenciem as possibilidades de buscas distribuídas por diversos catálogos, ainda que seja possível inferir que tal situação é viável.

A arquitetura proposta para o WeBIOS atende às necessidades básicas para a integração de fontes de dados heterogêneas no contexto das pesquisas em biodiversidade. Sua abordagem baseada em serviços web, padrões de metadados, padrões WMS e WFS OGC, ontologias e catálogos, garante flexibilidade e baixo acoplamento à arquitetura.

3.1.2.4 Problemas em aberto

Após a análise das arquiteturas apresentadas, foram identificados os seguintes aspectos em aberto, apresentados a seguir:

- i. **Interoperabilidade entre domínios heterogêneos** – No caso do DAGIS não há um componente responsável por sanar problemas de interoperabilidade entre domínios distintos. O problema da interoperabilidade pode ficar evidente, principalmente quando há necessidade de compor serviços web para satisfazer uma consulta. No SWING o problema da interoperabilidade entre domínios é sanado com a adoção de mediadores. No WeBIOS há o serviço web AONDÊ (DALCIO, 2007) que permite estabelecer as relações semânticas entre os domínios adotando métricas de similaridade entre textos;
- ii. **Registro das interações entre os usuários e a arquitetura** – Nas arquiteturas descritas não há um componente responsável por registrar as interações dos usuários. Com isso, não é possível aproveitar experiências prévias para melhorar o desempenho dos componentes;

- iii. **Personalização do funcionamento com base no perfil de cada usuário** – Nos *frameworks* apresentados, os usuários são considerados todos iguais. Não há a associação dos usuários a estereótipos de modo a permitir personalizar o funcionamento;

Além dos aspectos supracitados, há ainda melhorias possíveis no que diz respeito a facilitar a percepção do espaço, de modo a permitir que atores, com experiências e culturas diferentes possam descobrir e compartilhar informações, utilizando como base as percepções que cada um faz do espaço. Esses e outros aspectos relacionados são descritos em detalhes nos trabalhos (DAVIS JR.; FONSECA; CÂMARA, 2009) (CRAGLIA et al. 2008).

Uma vez que foram apresentados alguns trabalhos relacionados à Pangea, bem como explicitados alguns problemas ainda em aberto, as próximas seções apresentam e discutem os aspectos transversais identificados nas arquiteturas.

3.2 Disponibilização dos dados e operações em serviços web OGC

Atualmente há uma vasta quantidade de dados geográficos publicados na Web. Os dados são disponibilizados em diversos formatos: imagens de satélite, texto em linguagem natural, *shapefiles*, bancos de dados, serviços Web OGC, entre outros. Ainda que os padrões básicos da Web (HTTP, TCP/IP, DNS) e seus mecanismos de recuperação de informação (Google, Yahoo, Bing) permitam localizar e recuperar estes dados, a ampla variedade de formatos nos quais são disponibilizados, somada às técnicas de recuperação de informação com base em palavras-chave, leva a limitações, sobretudo nos casos onde há necessidade de descobrir e integrar fontes de dados e modelos geográficos.

Os problemas de interoperabilidade mencionados são sintáticos, ou seja, decorrem das diferenças do esquema utilizado para descrever e acessar os dados e modelos geográficos. Essas heterogeneidades limitam a reutilização dos recursos geográficos.

Em virtude da ampla variedade de formatos, na última década o OGC vem propondo o desenvolvimento de padrões para possibilitar o acesso aos dados e processos geográficos distribuídos na Web. Os serviços OGC são empregados nas arquiteturas estudadas na seção 3.1.

Nesse sentido, na seção 3.2.1 são apresentados os trabalhos e iniciativas que os adotam como meio de resolver os problemas de interoperabilidade sintática de dados e processos geográficos.

3.2.1 Trabalhos e conceitos relacionados

Na última década o *Open Geospatial Consortium* – OGC liderou a especificação de padrões para possibilitar o acesso uniforme aos dados e modelos geográficos, independentemente da tecnologia na qual são implementados e armazenados. Os padrões do OGC são baseados no estilo arquitetural orientado a serviços (*Service Oriented Architectures* – SOA) e estabelecem serviços web e interfaces que permitem a interoperabilidade sintática entre plataformas/formatos distintos.

Atualmente, os padrões OGC são amplamente aceitos e utilizados pela comunidade que trabalha com dados/aplicações geográficas (BRGM, 2007). No Brasil se destacam iniciativas do Ministério do Meio Ambiente – MMA, da Agência Nacional de Águas – ANA, da Comissão Nacional de Cartografia – CONCAR (2009) e do IPEA (2010) para disponibilizar dados geográficos através de serviços OGC.

A estratégia proposta pelo OGC consiste na inserção de uma camada de serviços web que permite o acesso uniforme aos dados e modelos geográficos, resolvendo as diferenças sintáticas. Dentre a ampla variedade de padrões propostos pelo OGC, a grande maioria dos trabalhos estudados adotam especialmente: *Web Map Service* – WMS, *Web Feature Service* – WFS e *Web Coverage Service* – WCS. O *Web Processing Service* – WPS é menos adotado se comparado aos anteriores.

No WeBIOS (DALCIO, 2007)(GOMES JR., 2007) os repositórios de dados são agrupados na “Camada de Armazenamento”. Cada repositório é classificado de acordo com o tipo de dado que armazena. As classes de repositórios são: imagens, registros de coletas, informações georeferenciadas, metadados e ontologias. Para cada tipo de dado é definida uma classe de serviço web. Os serviços web são agrupados na “Camada de Serviços de Suporte”. Considerando especialmente os repositórios de informações geográficas, estes são disponibilizados de acordo com o padrão WFS do OGC.

No SWING (BRGM, 2007) os dados e modelos geográficos são disponibilizados em serviços Web WMS, WFS e WPS do OGC.

A adoção dos padrões de serviços web como o WMS, WFS e WPS permitem a troca de dados e modelos geográficos independentemente do formato e tecnologia na qual são implementados/armazenado. Tal patamar de interoperabilidade será referenciado no decorrer deste trabalho como interoperabilidade em nível sintático.

3.2.2 Abordagem adotada

Considerando os problemas de interoperabilidade decorrentes da heterogeneidade sintática dos dados e modelos geográficos, neste trabalho são adotados os padrões WMS, WFS e WPS do OGC.

Ainda que a adoção dos padrões OGC para serviços web possibilite a interoperabilidade sintática de dados e modelos geográficos, há um outro nível de interoperabilidade que não é tratado pelos padrões OGC: o nível semântico. A falta da descrição semântica dos dados e modelos limita o descobrimento e a integração dos recursos geográficos aos aspectos sintáticos. Com isso, não é possível recuperar e integrar os dados e processos levando em consideração a semântica relacionada aos mesmos.

Para resolver consultas mais complexas e permitir automatizar a descoberta e composição de serviços OGC, alguns trabalhos tem adotado ontologias para descrever os serviços web. A descrição semântica por meio de ontologias permite incrementar as técnicas de descobrimento e integração de serviços web. Nesse sentido, nas próximas seções são descritos aspectos relacionados à semântica.

3.3 Servidor semântico

A evolução e disseminação da Web, sobretudo como plataforma para sistemas de informação, somadas à especificação e adoção de padrões conceituais e tecnológicos (definidos por entidades como o W3C, OGC, ISO IEC, entre outros), potencializam as possibilidades de interoperabilidade entre sistemas de informação (FONSECA; EGENHOFER; BORGES, 2000).

Essencialmente, há duas grandes classes de interoperabilidade: (i) Interoperabilidade sintática; (ii) Interoperabilidade semântica. Os problemas de interoperabilidade sintática decorrem das heterogeneidades em nível de esquema (i.e. Formato no qual o dado está representado) e protocolo de comunicação (i.e. Padrão utilizado para comunicação entre as partes).

Por outro lado, os problemas de interoperabilidade semântica decorrem do fato de as informações possuírem mais de um significado associado (FONSECA; EGENHOFER; BORGES, 2000). Desse modo, ainda que a interoperabilidade sintática permita trocar recursos, a ausência de

semântica formal associada aos recursos prejudica a interoperabilidade em nível semântico.

Na medida em que cresce o número de informações e operações publicadas na Web segundo um padrão (ex. um padrão OGC de serviços web) surge a possibilidade de compor as informações e as operações de variadas formas. A composição dos dados permite criar novas aplicações, diferentes das inicialmente planejadas pelos provedores originais dos dados. Além disso, a integração das informações pode ser utilizada para a resolução de problemas diversos, indo além das expectativas iniciais de quem modelou e coletou as informações. Tim Berners-Lee (2009) apresenta e caracteriza essa potencialidade na medida em que apresenta as relações entre o modelo relacional e o modelo de grafo orientado da Web Semântica.

Entretanto, tal potencialidade é acompanhada de um desafio: garantir que as informações e as operações sejam interoperáveis tanto no patamar sintático como no semântico (FONSECA; SHETH, 2002). Tal desafio ainda persiste, atualmente, como um problema de pesquisa relevante.

Considerando os SIGs, a adoção de uma camada de serviços web baseada nos padrões OGC permite resolver os problemas de interoperabilidade sintática (como já abordado na seção 3.2), porém, a semântica permanece um problema em aberto. Atualmente, a principal proposta para permitir que sistemas interoperem no nível semântico em escala global é a Web Semântica (BERNERS-LEE; HENDLER; LASSILA, 2001). Decorrente da ampla adoção dos padrões da Web Semântica (como exemplo o RDF, o OWL, o SPARQL, entre outros) surgiu uma proposta de extensão espacial a esta idéia, denominada de *Geospatial Semantic Web*. De forma geral, a *Geospatial Semantic Web* consiste na inserção da dimensão espacial à Web Semântica.

Fonseca e Sheth (FONSECA; SHETH, 2002) apresentam uma síntese da *Geospatial Semantic Web*, e levantam os principais desafios de pesquisa relacionados com a interoperabilidade semântica de dados e operações geográficas. Entre os desafios mencionados por Fonseca e Sheth (2002) estão: criar e manter ontologias geográficas, prover mecanismos para o casamento de entidades geográficas presentes em páginas Web e conceitos em uma ontologia, e integração de ontologias geográficas.

Conforme verificado nas arquiteturas estudadas, ontologias são artefatos centrais para permitir a interoperabilidade em nível semântico entre sistemas na Web. Portanto, é importante que existam mecanismos que possibilitem descobrir, recuperar e compartilhar ontologias. É neste contexto que está inserido o Servidor Semântico. No contexto deste trabalho, o servidor semântico é concebido como um repositório de ontologias, com a principal funcionalidade de fornecer às aplicações meios de busca, acesso e inferência sobre descrições semânticas. Com isso, na seção seguinte são apresentadas abordagens utilizadas na construção de servidores semânticos.

3.3.1 Trabalhos e conceitos relacionados

A definição do conceito Servidor Semântico ainda é bastante confusa devido, principalmente, ao tema ser alvo recente de pesquisas e trabalhos (AHMAD; COLOMB, 2007).

Não há também uma definição clara sobre as funcionalidades do Servidor Semântico. Algumas iniciativas focam nas atividades de armazenamento e recuperação de ontologias. Este é o caso do Swoogle (SWOOGLE, 2007), do *framework* Sesame (SESAME, 2008) e dos componentes TDB e SDB do *framework* Jena (JENA, 2005). Outras iniciativas tem focado no suporte às etapas de engenharia de ontologias, fornecendo funcionalidades para o controle de versão, a criação, a edição, entre outras. Este é o caso do projeto Ontolingua (FARQUHAR; FIKES; RICE, 1997) cujo foco principal é disponibilizar ferramentas para o desenvolvimento colaborativo de ontologias.

De forma geral os servidores semânticos servem para fornecer a semântica (ex. Ontologias) necessária às aplicações, durante a realização de operações como: a anotação semântica, a formulação de consultas semânticas, o descobrimento e a integração de recursos, entre outras.

Daltio (2007) apresenta uma extensão à arquitetura descrita por Gomes Jr. (2007) introduzindo serviços web de suporte ao armazenamento, recuperação, alinhamento e gerenciamento de ontologias. Daltio (2007) divide o servidor semântico conceitualmente em repositórios e operações. São descritos dois tipos de repositórios: os Repositórios Externos de Ontologias e os Repositórios Semânticos. Os Repositórios Externos (ex. Swoogle) são de propósito geral e permitem acesso a ontologias de domínios diversos, desenvolvidas por terceiros. Já os Repositórios Semânticos armazenam ontologias de interesse do usuário além dos metadados associados. Na camada de operações há a especificação de um serviço web que encapsula funcionalidades relacionadas à busca, recuperação, integração e registro de ontologias. Considerando a implementação da abordagem, as ontologias são armazenadas em um banco de dados PostgreSQL utilizando o *framework* Jena para acesso e manipulação, e o protocolo SPARQL para a formulação de consultas.

Schade (2008) apresenta o servidor semântico como um repositório de conceitos. Para obter conceitos de interesse a partir do repositório, o usuário/agente deve fornecer como entrada um Contexto. Um Contexto é definido como um conjunto de palavras-chave. Para o Contexto de entrada o serviço web de busca que encapsula o repositório semântico retorna os conceitos relacionados ao Contexto. Um Contexto pode ser associado a uma ontologia e desta forma todos seus conceitos herdam o contexto definido. Mais detalhes sobre a implementação desse mecanismo

podem ser obtidos em (SCHADE, 2008). Uma demonstração do repositório de conceitos pode ser experimentada a partir de (SWING CONCEPT REPOSITORY, 2010). Esta organização permite recuperar descrição semântica a partir da navegação transversal entre Contextos.

No projeto Starlab (2009) o servidor semântico é organizado em quatro camadas: (i) Camada de Aplicação; (ii) Camada de objetos ontológicos; (iii) A API para persistência; (iv) Camada de Persistência. A camada (i) consiste das aplicações que utilizam descrição semântica. A camada (ii) expõe a descrição semântica do servidor aos aplicativos da camada (i) por meio de objetos, representados em alguma linguagem de alto nível. No caso do Starlab a linguagem Java é utilizada. A camada (iii) consiste de uma API que fornece acesso aos repositórios semânticos. Por fim, a camada (iv) consiste do mecanismo de armazenamento das descrições semânticas.

Organização arquitetural semelhante é empregada no SESAME (SESAME, 2008). No SESAME também há uma API para acesso aos repositórios, bem como uma camada de persistência e de aplicações. Entretanto, no SESAME não há uma camada de objetos que expõe os conceitos dos repositórios à camada de aplicativos.

Com base no exposto, na próxima seção é apresentada a abordagem adotada neste trabalho para o servidor semântico.

3.3.2 Abordagem adotada

A definição da abordagem adotada neste trabalho para o projeto e construção do servidor semântico passa por dois aspectos principais: (i) As funcionalidades requeridas para o servidor semântico; (ii) A organização arquitetural.

Considerando (i), o foco do Servidor Semântico deve ser armazenar e disponibilizar descrição semântica para as aplicações. Conforme apresentado por Schade (2008), o Servidor Semântico deve ser organizado como um repositório de conceitos agrupados por contexto. Ainda em (i) e considerando o trabalho de Daltio (2007), o servidor semântico deve possibilitar o acesso a repositórios externos (i.e. Swoogle, DbPedia, entre outros). Nesse aspecto é importante considerar os repositórios semânticos evidenciados nos trabalhos do *W3C Semantic Web Education and Outreach – SWEO Community Project*.

Com relação a (ii), a abordagem adotada neste trabalho toma como base o Sesame (2008) e o Starlab (2009). O servidor semântico descrito neste trabalho deve ser composto das seguintes

camadas:

- **Camada de persistência** - Esta camada consiste dos variados dispositivos de persistência, como banco de dados relacionais, arquivos, entre outros. Os mecanismos de persistência suportados devem decorrer daqueles suportados pelo Sesame (2008) e pelo Jena SDB e TDB (JENA, 2005);
- **Camada de API para acesso e manipulação da camada de persistência** - Consiste de uma API para facilitar o acesso aos repositórios internos e externos ao servidor;
- **Camada de serviços web** - Consiste nos serviços web que permitem o acesso às funcionalidades da camada de API através da Web;
- **Camada de aplicação** - Consiste das aplicações que utilizam os objetos da camada de objetos ontológicos.

Uma vez que tenham sido apresentadas as linhas gerais da abordagem adotada para o projeto do módulo servidor semântico, na seção 3.4 são descritos trabalhos relacionados à anotação semântica de dados e processos geográficos.

3.4 Anotação semântica

Áreas de pesquisa relacionadas à biodiversidade (GOMES JR., 2007), agricultura (MACÁRIO; MEDEIROS, 2009), gestão de políticas públicas sustentáveis para meio ambiente (DAVIS; CÂMARA; FONSECA, 2009), gestão de recursos minerais (KLIEN, 2008) são apenas alguns exemplos de situações que corroboram o papel determinante da dimensão geoespacial na resolução de problemas e, principalmente, a necessidade de mecanismos mais eficientes de descobrimento, recuperação e integração de dados e modelos geográficos distribuídos.

Apesar das especificações OGC consistirem de uma evolução na troca de dados e processos geográficos entre sistemas distintos, estas limitam o descobrimento, recuperação, integração e processamento dos dados e modelos geográficos aos aspectos sintáticos, desconsiderando a semântica relacionada.

Desse modo, os principais mecanismos de descobrimento e recuperação de informação ficam restritos aos aspectos sintáticos dos dados e modelos geográficos. A grande maioria desses mecanismos são baseados em palavras-chave (BEPPLER, 2008) (HILDEBRAND; OSSENBRUGGEN; HARDMAN, 2007), o que gera resultados insatisfatórios devido

principalmente aos problemas de definição da semântica, inerentes à linguagem natural (i.e. homônimos, sinônimos, entre outros). Portanto, para permitir descobrir, recuperar e integrar dados e modelos geográficos de maneira mais eficiente, é necessário considerar os aspectos semânticos.

É neste cenário que se insere o conceito de anotação semântica. No contexto específico dos dados geográficos, a anotação semântica é definida como o estabelecimento das relações entre os dados/operações geográficas e os conceitos formais de um domínio compartilhado por uma comunidade (KLIEN, 2008) (MAUÉ; SCHADE; DUCHESNE, 2009).

De modo formal, uma anotação semântica consiste em um conjunto composto por unidades de anotação semântica. Cada unidade de anotação semântica é definida por uma tripla $\langle s, m, o \rangle$, onde “s” representa o sujeito (i.e. O recurso ao qual a anotação se refere), “m” denota um campo de metadado e “o” um termo de uma ontologia (MACÁRIO, 2009). As unidades de anotação semântica estabelecem a relação entre o recurso e o conceito do domínio.

O estabelecimento dessas relações permite que os dados e modelos possam ser descobertos, recuperados e integrados a partir da análise dos conceitos e suas relações de domínio. Isso permite a realização de consultas semânticas e, por consequência, diminui os impactos negativos decorrentes do procedimento de descoberta e recuperação com base em palavras-chave (HILDEBRAND; OSSENBRUGGEN; HARDMAN, 2007). Desse modo, se torna importante definir os meios através dos quais as relações semânticas entre os dados/processos e o domínio serão encontradas e como serão representadas.

Porém, antes de especificar os meios utilizados para encontrar e representar as relações entre os recursos (i.e. dados e modelos) e os conceitos do domínio, é fundamental considerar três aspectos: (i) as características dos recursos, alvos do procedimento de anotação; (ii) o modelo formal utilizado para representar os conceitos e as relações de domínio; (iii) o padrão de descrição empregado. Os aspectos (i) e (ii) são relevantes pois é a partir das características do universo de dados e do modelo utilizado para representar formalmente o domínio que é possível caracterizar e estabelecer um modelo conceitual e, além disso, derivar o procedimento utilizado para encontrar e representar as relações entre os recursos e os conceitos de domínio. O aspecto (iii) permite estabelecer um padrão de metadados para descrever os aspectos relacionados aos serviços web.

Na seção seguinte é apresentado e discutido um modelo conceitual que suporta a anotação semântica de serviços web geográficos. Além disso, são apresentados mecanismos de anotação semântica para dados geográficos encontrados em trabalhos semelhantes a este. Muitos dos trabalhos discutidos na próxima seção, bem como o modelo conceitual descrito não consideram a anotação semântica de modelos geográficos.

3.4.1 Trabalhos e conceitos relacionados

Embora hajam muitos trabalhos abordando a utilização de ontologias no procedimento de anotação semântica, ainda há poucas iniciativas no sentido de especificar um modelo conceitual e uma estratégia que permitam guiar e tornar reproduzível o procedimento de anotação semântica de serviços web geográficos (KLIEN, 2008).

3.4.1.1 Modelo conceitual para anotação de dados geográficos

No contexto da definição de um modelo conceitual para a anotação de dados geográficos, Klien (2008) apresenta inicialmente um modelo conceitual para o procedimento de anotação semântica de dados geográficos e, deste modelo, é adotada uma estratégia para obter anotações semânticas.

Klien (2008) sugere que as *features types* podem ser associadas às classes de um modelo formal (i.e. ontologia) e, conseqüentemente, as *features* pertinentes a esta *feature type* podem ser classificadas como instâncias da respectiva classe no modelo formal. Um exemplo de anotação de uma *feature type*, utilizando o modelo conceitual descrito por Klien (2008) é apresentado na figura 7.

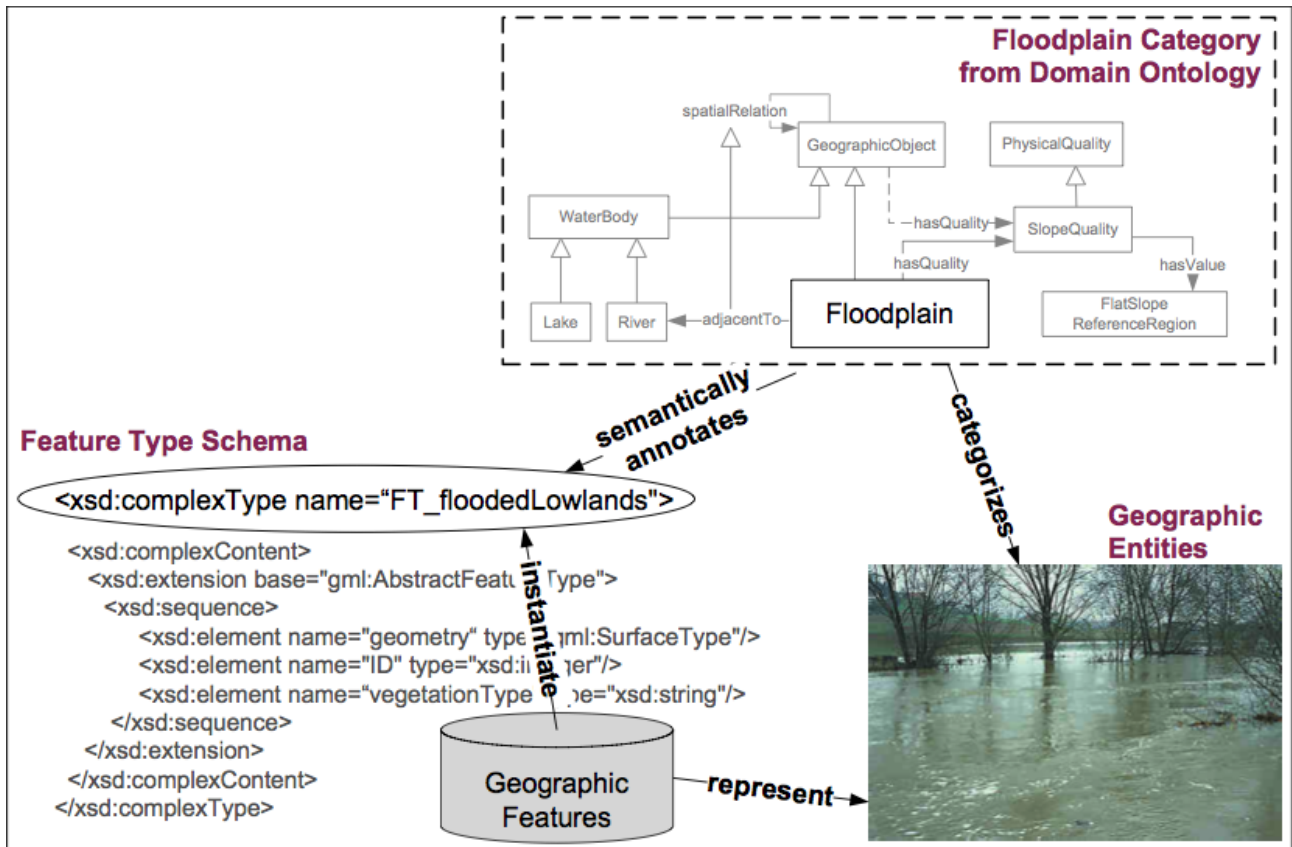


Figura 7. Modelo conceitual para a anotação semântica de dados geográficos.

Fonte: Extraído de Klien (2008).

Neste modelo conceitual, as entidades geográficas reais (*Geographic Entities*) são representadas e instanciadas por *features* em um banco de dados e disponibilizadas por uma interface de serviço web no padrão WFS do OGC. Uma ontologia de domínio (*Floodplain Category from Domain Ontology*) categoriza as entidades de acordo com um modelo formal, constituído de conceitos e suas relações. Os conceitos da ontologia de domínio são utilizados na anotação semântica das *feature types*.

Em decorrência do modelo conceitual exposto, Klien apresenta uma estratégia de anotação para serviços Web de dados geográficos. Esta estratégia consiste em transformar sintaticamente o *feature type schema* (GML) para uma ontologia (OWL, WSMML, ou outro formato qualquer) e então associar os conceitos extraídos do *feature type schema* aos conceitos de uma ontologia de domínio. Estes passos são apresentados na figura 8.

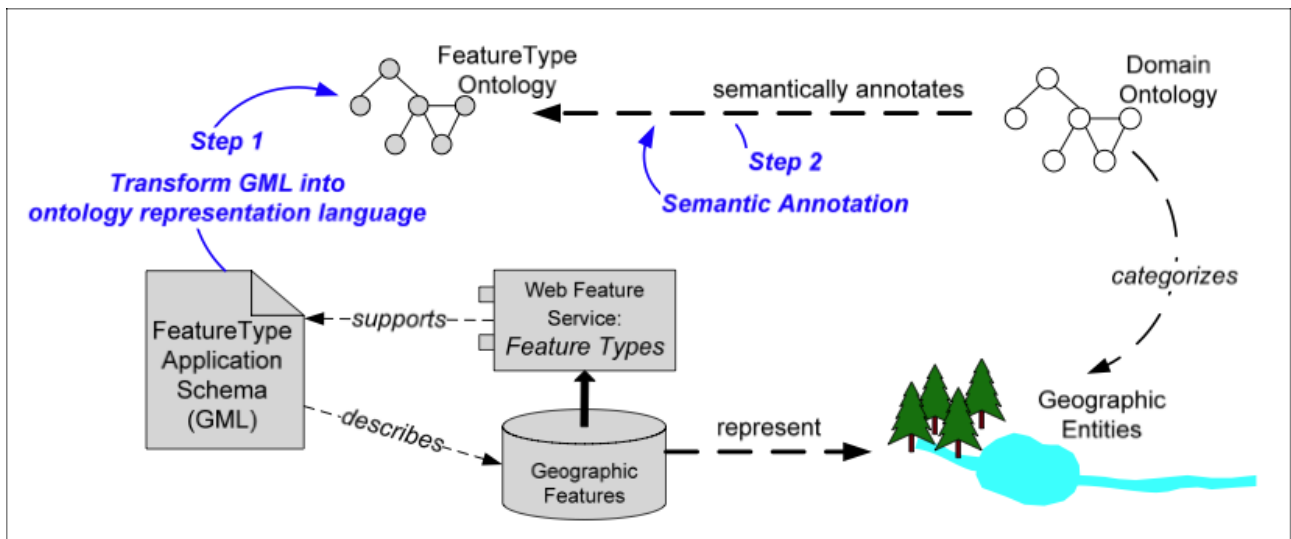


Figura 8. Estratégia de anotação semântica.

Fonte: Extraída de Klien (2008).

A primeira etapa da estratégia consiste em transformar a descrição do esquema de dados (*FeatureType Application Schema*), disponível em GML (padrão utilizado pelo WFS), para o formato OWL. Esta transformação é estritamente sintática e não há adição de informação ou relações semânticas ao esquema de dados. O resultado da transformação é uma ontologia denominada *FeatureTypeOntology*. A segunda etapa consiste no procedimento de anotação semântica. Neste processo, os conceitos da *FeatureTypeOntology* são associados aos conceitos da ontologia de domínio (*DomainOntology*). No modelo conceitual, bem como na estratégia apresentados por Klien (2008) é considerado apenas o padrão WFS no procedimento de anotação.

Uma vez expostos em linhas gerais o modelo conceitual e a estratégia de anotação semântica, na seção seguinte são apresentados e discutidos mecanismos empregados no procedimento de anotação semântica.

3.4.1.2 Abordagens utilizadas para a anotação de dados geográficos

De forma geral, as dificuldades relacionadas ao procedimento de anotação de recursos (serviços web geográficos, páginas XHTML, imagens, vídeos, entre outros) decorrem da grande quantidade e heterogeneidade dos dados a serem anotados (KIRYAKOV; POPOV; TERZIEV; MANOV; OGNYANOFF, 2004) (REEVE; HAN, 2005) (GRCAR, 2008). Neste cenário, métodos manuais não são considerados adequados sendo as técnicas semi-automáticas e automáticas as mais

recomendadas.

Neste sentido, a questão central a ser respondida antes de definir e caracterizar o método utilizado para a anotação semântica é definir qual o universo de dados a ser anotado.

Há casos em que o alvo da anotação semântica são arquivos XHTML disponíveis na Web ou até mesmo textos livres de estruturação. Há outros casos em que o alvo da anotação são serviços web. Neste trabalho o universo que será alvo do procedimento de anotação semântica é composto por serviços web conforme os padrões WMS, WFS e WPS do OGC.

Grcar (2008) apresenta um mecanismo de anotação onde o alvo do procedimento de anotação semântica são serviços web de informações geográficas conforme os padrões WMS, WFS e WPS do OGC. Nesta abordagem a anotação de serviços web é semi-automática e baseada em aprendizado supervisionado. Os serviços web alvo da anotação possuem métodos que retornam os metadados e o esquema dos dados relacionados. Estes métodos são o *getCapabilities* e o *describeFeatureType* no caso do WFS.

O procedimento de anotação apresentado por Grcar utiliza as informações providas pelos métodos *getCapabilities* e *describeFeatureType* em duas etapas: (i) as informações retornadas pelos métodos são transformadas sintaticamente do formato XML para o formato WSML; (ii) As informações representadas no WSML são utilizadas para classificar e anotar o recurso (GRCAR, 2008).

Ainda no contexto da anotação dos serviços OGC, Maué, Schade e Duchesne (2009) argumentam que há três fontes de informações que podem ser utilizadas no procedimento de anotação semântica: o documento de metadados relacionado ao serviço OGC, obtido por meio da chamada ao método *getCapabilities*; o documento de esquema que descreve a estrutura sintática das *feature types*, obtido por meio da chamada ao método *describeFeatureType*; os dados propriamente ditos.

Estas fontes de informação relacionadas aos serviços web geográficos permitem realizar a anotação semântica em três níveis distintos: (i) Anotação em nível de metadados; (ii) Anotação em nível de esquema; (iii) Anotação em nível de dados.

A anotação semântica em nível de metadados é a abordagem mais simples e pode ser implementada estendendo o documento de metadados do serviço web (documento obtido pela invocação do método *getCapabilities*). A extensão se dá por meio de uma alteração da seção de palavras-chave, onde são adicionadas referências para conceitos de uma ontologia de domínio. A maior vantagem desta abordagem é a simplicidade de sua implementação. Por outro lado, esta abordagem limita o descobrimento dos recursos à sua camada de metadados.

A anotação em nível de esquema permite suplantar as limitações em nível de metadados na

medida em que considera os atributos que caracterizam a *feature type* no processo de anotação semântica. A vantagem direta desta abordagem consiste no maior nível de detalhes disponível para a anotação semântica. Com isso, mais serviços web podem satisfazer uma consulta, visto que há mais informações anotadas. Entretanto, o maior nível de detalhes resulta em maior quantidade de anotações semânticas.

Por fim, a abordagem em nível de dados ou instância considera a anotação semântica de cada *feature* geográfica em separado. Esta abordagem pode ser interessante quando não há disponível um modelo de dados que explicita a estrutura lógica das *features* (i.e. *feature schema*). Em contrapartida, esta abordagem pode ser bastante custosa devido à grande quantidade de *features* geográficas, aumentando significativamente a quantidade de anotações semânticas.

Há outros trabalhos que fazem uso dos aspectos espaciais dos recursos sendo anotados. As características espaciais são utilizadas em Klien (2005).

Klien e Lutz (2005) apresentam um método de anotação que considera as relações espaciais (especialmente as topológicas) entre as *features* para anotar semi-automaticamente dados geográficos.

No método descrito por Klien e Lutz (2005), o primeiro passo consiste em selecionar uma ontologia de domínio e o conjunto de dados a ser anotado. Após, são identificadas na ontologia conceitos e suas relações espaciais. Cada relação espacial descoberta é então derivada para uma operação espacial. Então, o conjunto de dados a ser anotado é submetido à operação espacial selecionada e caso satisfaça as restrições espaciais, é anotado com o conceito em questão. No exemplo demonstrado por Klien um conjunto de dados sobre áreas de alagamento é anotada semanticamente a partir da relação espacial que define o conceito “*Áreas de Várzea*”: “*Toda área de várzea é adjacente a um rio.*” A expressão “*adjacente*” é identificada como uma relação espacial associada ao conceito “*Áreas de Várzea*” e, então, é traduzida em uma operação espacial que representa a adjacência. Esta operação é aplicada aos dados sendo anotados, utilizando uma base de referência (i.e. o conjunto de dados de rios). Todas as *features* que satisfizerem a condição espacial são anotadas como “*Áreas de várzea*”.

Uma das vantagens da abordagem utilizada por Klien e Lutz (2005) é a independência do processamento de linguagem natural no procedimento de anotação, o que evita as ambiguidades decorrentes da linguagem (i.e. sinônimos, homônimos, entre outros). Uma extensão ao trabalho de Klien e Lutz (2005) é descrita em (KLIEN, 2008), onde as principais diferenças são: uma etapa de pré-processamento que avalia a similaridade entre textos; utilização de regras (conforme o padrão SWRL) para expressar condições de classificação de instâncias. A utilização de regras torna a formalização do domínio mais expressiva na medida em que permite a utilização de variáveis. A

partir das regras, as anotações sugeridas pela etapa de pré-processamento são avaliadas.

3.4.2 Abordagem adotada

Neste trabalho é adotado um modelo conceitual de anotação adaptado do descrito por Klien (2008). A principal extensão ao modelo de Klien consiste na inserção de serviços de processamento (i.e. WPS). O modelo conceitual e a estratégia de anotação adotados neste trabalho são apresentados nas Figuras 9 e 10 respectivamente.

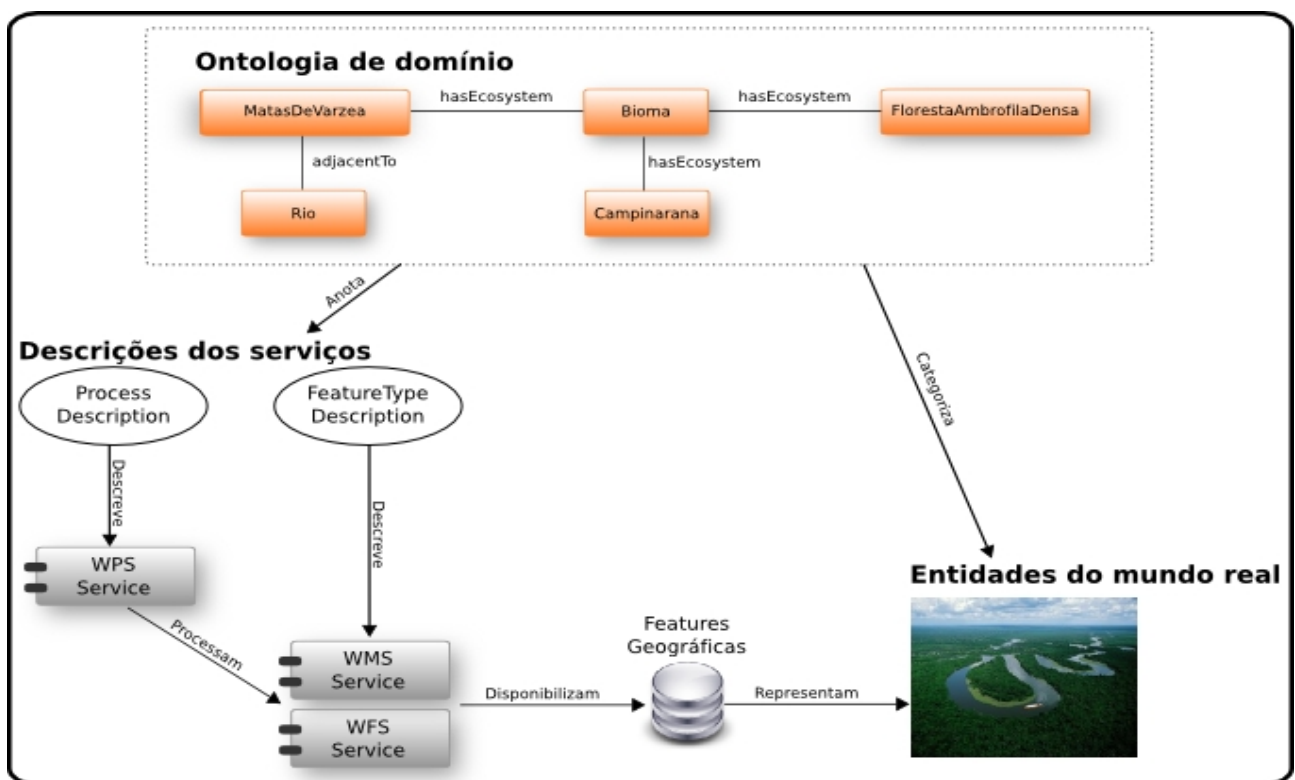


Figura 9. Modelo conceitual adotado neste trabalho.

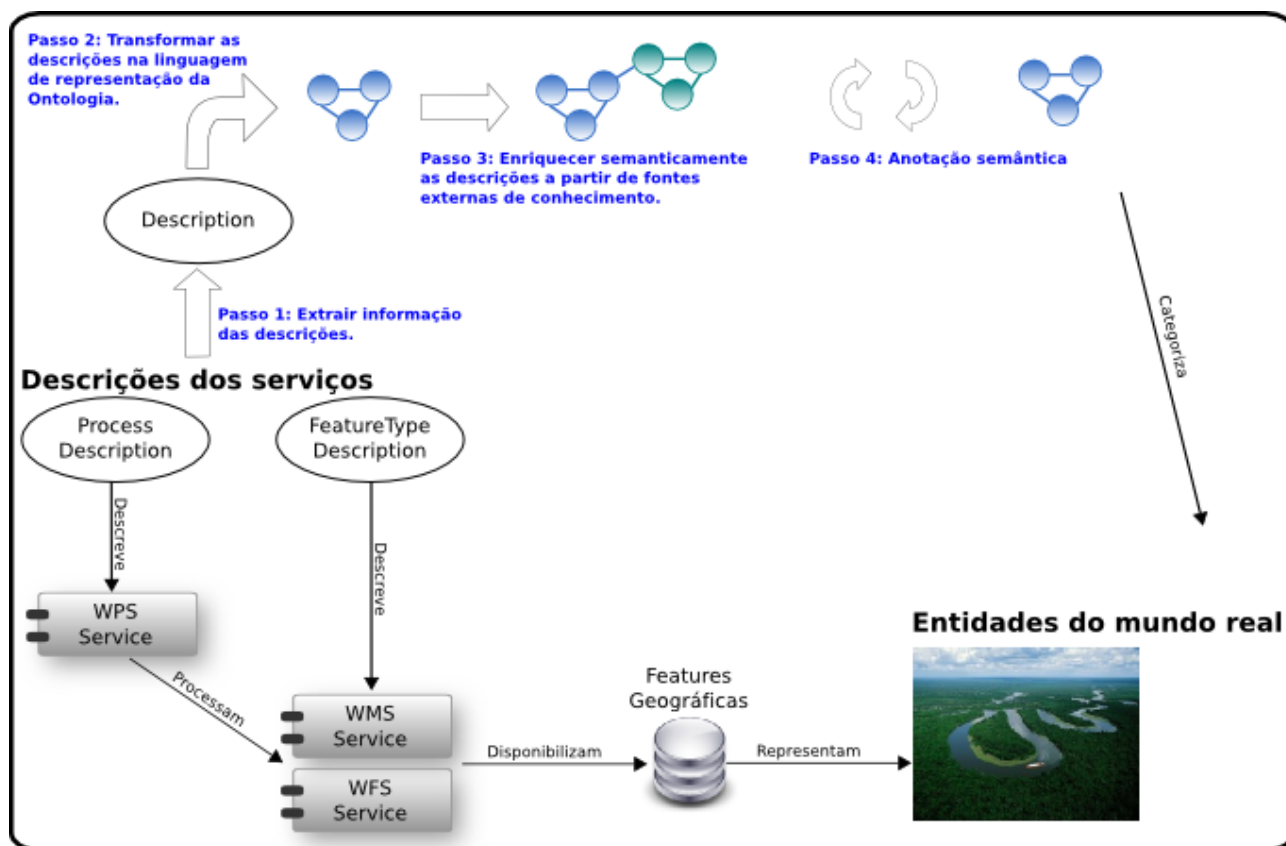


Figura 10. Estratégia de anotação semântica adotada neste trabalho.

A definição de um modelo conceitual bem como de uma estratégia de anotação permitem estabelecer os componentes e as etapas básicas que devem guiar um processo de anotação semântica de serviços OGC. Esses aspectos são utilizados na definição dos requisitos e do projeto do módulo de anotação semântica, apresentado no Capítulo 4.

Neste trabalho são considerados como alvo da anotação semântica serviços OGC conforme os padrões WFS, WMS e WPS. Além disso, as descrições de domínio são especificadas em lógica descritiva, conforme o padrão OWL-DL. As anotações semânticas são formalizadas utilizando como base o *framework* de descrição OWL-S.

O processo de anotação semântica adotado consiste em associar os serviços web de dados e modelos geográficos aos conceitos de um domínio, levando em conta as entradas e saídas aceitas pelos serviços geográficos. Com isso, a anotação semântica resultante é formalizada como um *OWL-S Service Profile*.

De modo geral, a abordagem empregada neste trabalho para anotação semântica dos recursos geográficos é baseada em quatro etapas:

- (i) **Extração de metadados** – Inicialmente são extraídos dos serviços geográficos os metadados relacionados. A extração dos metadados varia de acordo com a natureza dos

serviços (i.e. serviços de dados ou processos) sendo anotados. De forma similar ao apresentado por Klien (2008) os metadados extraídos são transformados do formato GML para OWL. Após a fase de extração dos metadados, são agregadas aos dados do serviço e à ontologia de domínio informações de provedores externos. A etapa de agregação de informações é apresentada no item (ii);

(ii) **Enriquecimento semântico** – Essa etapa consiste na associação aos dados e à ontologia de domínio de informações provenientes de provedores externos. A associação de informações externas permite enriquecer as informações alvo da anotação, fornecendo maiores subsídios para o procedimento de anotação semântica conduzido na etapa seguinte e também durante a etapa de validação por parte do usuário (etapa (iv));

(iii) **Anotação semântica** – Consiste na anotação semântica dos serviços de dados geográficos. A anotação leva em consideração os aspectos espaciais (de forma similar ao descrito em Klien e Lutz (2005)) e também os textuais, utilizados na descrição dos serviços de dados geográficos;

(iv) **Validação das anotações** - Uma vez que tenham sido anotados os recursos com base em aspectos espaciais e textuais, as anotações semânticas são retornadas ao usuário para validação. Nessa etapa o usuário deve avaliar as anotações semânticas sugeridas para os recursos geográficos e então validar as anotações pertinentes;

(v) **Inferência de novas anotações** – Por fim, a etapa de inferência consiste na aplicação de inferência para descobrir novas anotações semânticas a partir das anotações validadas pelo usuário na etapa (iv).

Essas etapas são melhor detalhadas no Capítulo 4, juntamente com os componentes pertinentes a cada etapa.

A anotação semântica permite melhorias no descobrimento dos serviços web, conforme já mencionado. Ainda assim, há casos onde serviços web são anotados frente a diversos domínios. Esse cenário determina a necessidade por mecanismos que permitam alinhar e integrar domínios distintos, de modo a permitir a integração de informações e processos. Desse modo, na seção 3.5 são apresentados e discutidos trabalhos relacionados com a integração de ontologias.

3.5 Integração de ontologias

A adoção de ontologias em sistemas de informação se dá em diversos contextos: integração de fontes de dados heterogêneas (GOMES JR., 2007) (FONSECA; EGENHOFER, 1999), apoio na formulação de consultas (FONSECA; EGENHOFER, 1999) (GOMES JR., 2007) (VIEGAS; SOARES, 2007), personalização através de *profile* semântico (AYRES; FURTADO, 2007), descobrimento semântico de recursos, entre outros.

Nas situações mencionadas, seria ideal que houvesse uma grande ontologia formalizada e compartilhada por todos os usuários/agentes, de modo a evitar problemas de interoperabilidade. No entanto, a realidade é que há uma grande quantidade e variedade de modelos semânticos disponíveis na Web Semântica. Tais modelos diferem em diversos aspectos, que podem variar desde diferenças sintáticas, linguísticas e até mesmo de modelagem (DALCIO, 2007).

Nesse cenário, é fundamental que haja meios de descobrir e representar as ligações semânticas que existem entre os conceitos de ontologias distintas, de modo a permitir potencializar a interoperabilidade semântica entre os sistemas que operam sobre essas ontologias. O descobrimento das relações semânticas entre conceitos de ontologias distintas é denominado *Ontology Matching*, enquanto a materialização das relações é denominado *Ontology Alignment*.

Conforme já mencionado, há muitos casos onde o alinhamento de ontologias é importante. No caso da integração de fontes de informação heterogêneas a falta de alinhamento entre ontologias distintas pode limitar o desempenho do integrador, uma vez que pode impedir que determinadas informações sejam recuperadas durante o procedimento de integração. Já no caso da formulação e resolução de consultas, a falta de relação semântica entre domínios pode limitar a capacidade de descobrir informações relacionadas. Portanto, mecanismos que permitam identificar as relações semânticas entre domínios distintos são fundamentais.

Desse modo, na seção 3.5.1 são apresentados e discutidos trabalhos que abordam o alinhamento semântico de ontologias. Os aspectos discutidos na seção 3.5.1 são utilizados na definição do módulo de alinhamento semântico empregado no Pangea.

3.5.1 Trabalhos e conceitos relacionados

A interoperabilidade semântica consiste da capacidade que um sistema possui para trocar mensagens (dados, operações) entre universos de discurso distintos e relacionados, levando em consideração os aspectos semânticos destes universos. Conforme já mencionado, a interoperabilidade semântica exerce papel fundamental em diversos contextos relacionados à sistemas de informação e ambientes abertos.

Um dos aspectos necessários para alcançar a interoperabilidade semântica consiste em estabelecer as relações semânticas entre conceitos de domínios distintos. O estabelecimento das relações semânticas se dá em duas etapas: (i) descobrir relações semânticas (i.e. *Ontology Matching*); (ii) representar a relação semântica entre os conceitos (i.e. *Ontology Alignment*). Com isso, na seção 3.5.1.1 são apresentados trabalhos que descrevem técnicas para descobrir as relações semânticas. Na seção 3.5.1.2 os meios de representar os mapeamentos semânticos são relacionados.

3.5.1.1 Descobrimto de relações semânticas

No que concerne ao descobrimto de relações semânticas, as técnicas podem ser classificadas em exatas ou baseadas em similaridade (MCGUINNESS; SHVAIKO; GIUNCHIGLIA; SILVA, 2004). As técnicas baseadas em medidas de similaridade são as mais utilizadas e são obtidas com base em diversas características do modelo semântico. As características extraídas e utilizadas no procedimento de *match* são provenientes dos aspectos estruturais, linguísticos e semânticos das ontologias.

Daltio (2007) apresenta uma abordagem de *match* baseada em aspectos linguísticos (sinônimos, homônimos, entre outros), similaridade de textos, e análise taxonômica. Na abordagem apresentada por Daltio (2007) é possível estabelecer pesos para determinar qual classe de características deve ser evidenciada no cálculo da similaridade.

Noy e Musen (2004) apresentam a ferramenta PROMPT. Nessa ferramenta, os mapeamentos são descobertos com base em aspectos léxicos e estruturais das descrições semânticas, sendo os resultados refinados através da interação com o usuário.

Stoutenburg (2008) descreve uma abordagem baseada no pressuposto de que as descrições

semânticas a serem relacionadas são derivadas de uma descrição mais geral (descrição de topo) e que estão representadas utilizando a linguagem OWL. Esse é um ponto importante no trabalho de Stoutenburg: o método descrito está diretamente acoplado ao formato específico OWL, o que permite que a técnica apresentada se aproveite de características decorrentes do formato para inferir relações semânticas. Ainda considerando o trabalho de Stoutenburg (2008), é feita a utilização de dicionários externos (recursos linguísticos) como o WordNet para descobrir relações de linguagem como sinônimos, hipônimos, hiperônimos, entre outros. A descrição de topo utilizada como base comum para as descrições de domínio é o OpenCyc. Em síntese: são avaliadas evidências primitivas derivadas das ontologias OWL, do dicionário WordNet e da descrição de topo OpenCyc. Estas evidências são utilizadas para determinar quais os conceitos que estão relacionados semanticamente e qual é a relação. Mais detalhes sobre os conjuntos de evidências e a maneira como são identificadas podem ser obtidos em (STOUTENBURG, 2008).

Sabou, D'Aquin e Motta (2006) apresentam uma abordagem que utiliza conhecimento externo (descrições externas) para superar as limitações de técnicas baseadas apenas no casamento de estruturas e de textos. Sabou apresenta argumentos nos quais defende que os mecanismos de identificação de mapeamentos semânticos podem ser melhorados através da utilização de outras descrições disponíveis na Web. No trabalho descrito em (SABOU; D'AQUIM; MOTTA, 2006) o Swoogle é utilizado para recuperar descrições semânticas relacionadas aos termos sendo analisados (essa etapa é denominada *anchoring*). Para cada descrição recuperada no Swoogle, o algoritmo analisa se há relação semântica entre os conceitos. Para melhorar a acurácia, nesta abordagem os termos sendo analisados podem ser substituídos por sinônimos (descobertos por meio de consultas a dicionários como o WordNet) e o cálculo da similaridade é realizado com base nos sinônimos. Uma extensão ao trabalho de Sabou, D'Aquin e Motta (2006) é apresentada por Gracia, Lopez e D'Aquin (2007), onde é proposto um mecanismo para avaliar a validade das associações semânticas feitas na etapa de *anchoring*. Para tanto, são utilizados mecanismos de desambiguação de palavras baseados em análise de sinônimos e também na investigação do contexto semântico no qual estão inseridos os termos. Essa extensão, permitiu melhorar o desempenho do mecanismo de alinhamento em até 15% (GRACIA; LOPEZ; D'AQUIN, 2007).

No contexto da interoperabilidade semântica geoespacial, Hess (2008) apresenta um mecanismo que considera as características comumente empregadas (linguística, taxonomia, entre outras), e inclui as características espaciais, para definir mapeamentos entre conceitos de descrições semânticas geoespaciais. Nesta abordagem é possível atribuir um peso a cada característica medida e com isso adequar o descobrimento dos mapeamentos às características das descrições sendo relacionadas. Além disso, é discutido e proposto o desenvolvimento de uma descrição semântica

geoespacial com o objetivo de obter melhorias no procedimento de descoberta dos mapeamentos.

Uma vez que as relações semânticas tenham sido identificadas, é necessário representar tais relações. Portanto, na seção seguinte são apresentadas as maneiras de representar tais relações semânticas.

3.5.1.2 Representação das relações semânticas

Encontrar as relações semânticas entre as entidades de duas ontologias é o primeiro passo necessário para a integração entre domínios distintos. Porém, para que as relações semânticas entre os domínios sejam efetivamente utilizadas, é fundamental representar as relações descobertas na etapa de *matching*.

Há duas abordagens principais para representar os mapeamentos entre ontologias: (i) Utilizar primitivas da linguagem OWL; (ii) Adotar uma linguagem de representação de mapeamentos. A abordagem (i) é baseada nas primitivas *owl:sameAs* e *owl:equivalentClass*. Por outro lado, a abordagem (ii) permite representar relações mais complexas entre os conceitos, do que simplesmente as relações de equivalência propostas pelas primitivas OWL já mencionadas.

Daltio (2007) adota em seu trabalho as primitivas OWL. Para representar a equivalência entre classes é empregada a propriedade *owl:equivalentClass*. A propriedade *owl:sameAs* é utilizada para relacionar instâncias e também classes com instâncias.

Ainda que a abordagem (i) seja bastante simples de ser implementada, há algumas limitações referentes à representação de relações mais complexas. Scharffe (2007) apresenta os requisitos que devem ser considerados no projeto de uma linguagem para representar mapeamentos mais complexos do que os de equivalência. Em síntese, Scharffe (2007) argumenta que uma linguagem de representação de mapeamentos deve suportar: a expressão de relações condicionais entre as entidades (ex. Se o atributo velocidade de uma instância de carro for maior que 200 então as instâncias podem ser mapeadas como instâncias de um conceito carro-esportivo em outra ontologia); (ii) Representar a direção do mapeamento; (iii) Representar mapeamentos que indicam a união ou intersecção de entidades; (iv) Ser padronizada a ponto de regras de mapeamento poderem ser compartilhadas entre sistemas. Scharffe propõe uma linguagem de representação de mapeamentos com base nos requisitos anteriores. Porém, a maioria das técnicas de *matching* propostas atualmente (conforme discutido na seção 3.5.1.1) permitem apenas descobrir

mapeamentos de equivalência entre entidades, sendo portanto, os recursos da linguagem proposta por Scharffe subutilizados.

3.5.2 Abordagem adotada

O procedimento de integração adotado neste trabalho é multicritério e toma como base os trabalhos de Sabou, D'Aquim e Motta (2006); Gracia, Lopez e D'Aquin (2007); Hess (2008). Para a etapa de cálculo da similaridade são empregadas medidas léxicas, semânticas e espaciais. Para cada um dos aspectos citados, o usuário pode determinar o peso aplicado em cada medida. A saída dessa etapa é o grau de similaridade entre as entidades da ontologia, considerando os aspectos léxico, semântico e espacial. Após, o usuário valida as sugestões de mapeamentos.

Os mapeamentos validados da etapa anterior devem ser materializados em algum formalismo lógico. Como os mapeamentos medidos são de equivalência, estes são representados utilizando as primitivas da linguagem OWL. Os componentes bem como as métricas empregadas no *matching* e na integração (i.e. *Align*) de ontologias são apresentados em detalhes no Capítulo 4.

3.6 Especificação e processamento de consultas

Nessa seção são apresentadas as etapas principais envolvidas no descobrimento de recursos. O objetivo é fornecer uma visão geral que permita, para cada etapa, considerar os aspectos importantes. Por outro lado, na seção 3.7 é apresentada uma visão mais específica acerca do descobrimento de recursos. Tal visão contextualiza as abordagens de descobrimento de serviços Web utilizando como contexto as macro etapas evidenciadas nessa seção.

3.6.1 Contextualização e problemática

A busca é o meio mais utilizado de descobrir recursos na Web. Um procedimento de busca

tem início quando um usuário/agente especifica suas necessidades de informação por meio de uma consulta. As necessidades são remetidas ao sistema responsável por processar a consulta. Por fim, os resultados são retornados ao usuário. Um esquema ilustrando as etapas gerais de um procedimento de busca é apresentado na figura 11.

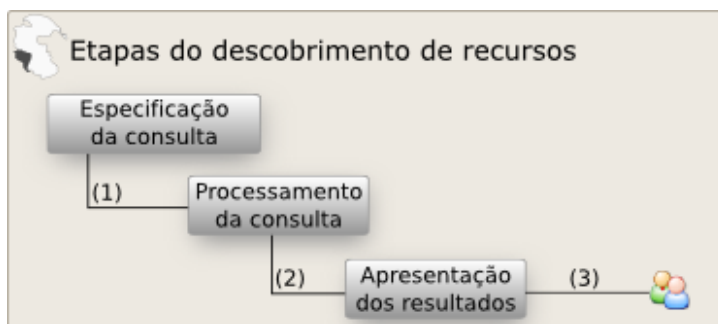


Figura 11. Etapas gerais do descobrimento de recursos .

Atualmente o processo de descobrimento de recursos é bastante dispendioso para o usuário pois, em geral, os sistemas de busca tradicionais retornam grandes quantidades de resultados irrelevantes. Esse fato obriga o usuário a percorrer pelo volume de resultados e filtrar aquilo que realmente interessa e condiz com as necessidades expressas na consulta.

Beppler (2008) discute e apresenta os fatores que contribuem para a ocorrência dos problemas expostos anteriormente. Ele argumenta que boa parte dos problemas ocorrem devido à ambiguidade das entradas, ou seja, a consulta especificada pelo usuário é pouco clara quanto a seus objetivos. Outro fator preponderante, argumenta Beppler (2008), é causado pela falta de interatividade entre os sistemas de busca e o usuário. O cenário é agravado devido à “desconexão” que ocorre entre o usuário e o sistema de busca durante as várias atividades de busca.

Os fatores expostos por Beppler (2008) são fundamentados em aspectos relacionados ao comportamento humano, considerando uma pessoa engajada no processo de busca por informação. Conforme apresentado por Beppler: *“incerteza e ansiedade são características que fazem parte do processo de busca por informação.”*. Somada a essas características está a dificuldade enfrentada por um usuário para expressar suas necessidades em uma consulta, o que acaba levando-o a uma sequência de interações indeterminadas com o sistema de busca (BEPPLER, 2008).

O fato preponderante é que sistemas convencionais de recuperação de informação possuem duas limitações: (i) Em geral, na etapa de especificação da consulta não são fornecidos mecanismos que permitam aos usuários/agentes serem mais expressivos em suas necessidades de informação; (ii) Os algoritmos de recuperação de informação consideram apenas aspectos sintáticos/léxicos para retornar os documentos relevantes. A limitação (i) leva a representações das necessidades dos

usuários/agentes bastante pobres, considerando a semântica, enquanto a limitação (ii) remete a resultados pouco precisos.

Diante do exposto e aproveitando a ampla revisão (foram analisadas 37 ferramentas de busca) realizada por Hildebrand et. al (HILDEBRAND e OSSENBRUGGEN e HARDMAN 2007), nas seções 3.6.1.1 e 3.6.1.2 são apresentadas em mais detalhes as etapas: (1) Especificação da consulta; (2) Processamento da consulta.

3.6.1.1 Especificação de consultas

A etapa de especificação da consulta consiste na formulação das necessidades de informação do usuário/agente. Hildebrand, Ossenbruggen e Hardman (2007) afirmam que entradas ambíguas levam a resultados irrelevantes. Portanto, a etapa de especificação da consulta tem importância fundamental nos resultados retornados pelo sistema. Com isso, é importante considerar as características das principais formas de expressar as consultas. Uma consulta pode ser expressa das seguintes formas:

- **Texto livre** – São consultas especificadas por textos livres. Nessa categoria há três formas de consultas: baseadas em palavras-chave, baseadas em frases (expressões em linguagem natural) e restritas a um conjunto de termos relacionados com o domínio de uma propriedade conhecida;
- **Estruturada** – São consultas baseadas em conceitos e suas relações. Um exemplo desse tipo de consulta são as expressões descritas em SPARQL. Nessa abordagem a consulta possui uma estrutura e semântica definidas.

Um dos problemas relacionados às consultas baseadas em texto livre é a dificuldade relacionada com a definição da semântica da consulta. Para que o sistema de busca retorne resultados relevantes é importante que a consulta expressa pelo usuário em texto livre seja interpretada e representada de maneira não ambígua. Na consulta baseada em estrutura, o mecanismo de busca tira vantagem das informações inerentes à estrutura para diminuir a ambiguidade da consulta.

Uma vez que a consulta tenha sido interpretada e representada formalmente, o sistema de busca deve executar o algoritmo para selecionar os resultados relevantes. O projeto do algoritmo de busca é fortemente relacionado às características da consulta (i.e. texto livre, frases, estruturas

semânticas, entre outras). A etapa de execução da busca é descrita na seção 3.6.1.2.

3.6.1.2 Processamento de consulta

Um mecanismo de busca pode ser caracterizado pela estratégia adotada para descobrir e organizar os resultados frente a uma consulta expressa pelo usuário/agente. Uma estratégia de busca deve considerar o universo de dados alvo da busca e os índices utilizados na organização do universo de dados.

O universo de dados pode ter características específicas que devem ser exploradas pelo mecanismo de busca para tornar os resultados mais precisos, além de ser possível um tempo de resposta menor. Com isso, as estratégias de busca são classificadas segundo duas abordagens (HILDEBRAND; OSSENBRUGGEN; HARDMAN, 2007):

- **Match Sintático:** baseado apenas em aspectos sintáticos (prefixo, “casamento” exato, distância de edição) do texto de entrada e dos recursos; e,
- **Match Semântico:** baseado em características semânticas. Os termos da consulta são associados a conceitos que são utilizados na busca. Os conceitos e suas relações semânticas são explorados para selecionar os resultados da busca.

Nesse contexto, Hildebrand, Ossenbruggen e Hardman salientam que comumente os mecanismos de busca baseados em texto utilizam uma etapa baseada em sintaxe e outra baseada em semântica no processo de recuperação e classificação dos dados.

Conforme já mencionado, a acurácia de um mecanismo de busca é determinada por diversos fatores, em especial: a capacidade de expressão dos objetivos, as características do universo de dados sendo buscado, índices, formato da consulta e interatividade. Na seção 3.6.2 são apresentados trabalhos que endereçam os fatores citados.

3.6.2 Trabalhos e conceitos relacionados

Nesta seção são apresentadas abordagens relacionadas à especificação e resolução de consultas semânticas.

Com o objetivo de minimizar os impactos dos fatores incerteza, ansiedade e dificuldade de expressão, Beppler (2008) apresenta o conceito de *information seeking*. Esse conceito é fundamentado em um ciclo de busca altamente interativo, que “conecta” as diversas interações do usuário na busca por informação. A navegação do usuário pelos resultados intermediários em processo de “vai-e-vem” é suportada por ontologias e permite refinar buscas para chegar a resultados mais precisos. O diferencial principal no trabalho de Beppler é a conexão entre os ciclos de busca, representados nas macro etapas: Especificação da Consulta, Processamento e Apresentação dos Resultados.

Considerando o formato com que as consultas são expressas, há trabalhos que consideram a entrada estruturada em conceitos e relações semânticas. Há outros trabalhos onde a entrada é realizada em texto livre, porém, antes de a busca ser executada, os termos são anotados semanticamente. Gomes Jr. (2007) apresenta uma arquitetura considerando consultas semanticamente estruturadas. Nessa arquitetura as consultas são expressas no formato SPARQL e resolvidas utilizando ontologias de domínio e de repositórios. O foco no trabalho de Gomes Jr. (2007) é a integração de fontes de dados, portanto, uma consulta pode ser resolvida com a composição de diversas fontes. As regras de composição de fontes são baseadas nos tipos de dados envolvidos e uma tabela de prioridades de resolução é aplicada para decidir como resolver cada consulta.

Lopez, Uren, Motta e Pasin (2007) apresentam o sistema AquaLog. Esse sistema foi projetado de modo a responder perguntas (*i.e. Question Answering System*). No AquaLog há três componentes principais que são responsáveis por processar as consultas em linguagem natural e obter as respostas: (i) *Linguistic Component*; (ii) *Relation Similarity Service*; (iii) *Inference Engine*. O AquaLog recebe como entradas uma pergunta em linguagem natural e uma ontologia que deve ser utilizada para processar a consulta e obter as respostas. O *Linguistic Component* recebe a consulta em LN como parâmetro e a transforma em uma representação intermediária de triplas, utilizando como base o modelo de dados <sujeito, predicado, objeto>. Essa transformação é feita com base na análise linguística da sentença e, posteriormente, com a detecção de padrões (representados por expressões regulares) especificados em uma gramática. Uma vez que a consulta

(no caso específico: pergunta) tenha sido transformada para sua representação em tripla, esta é classificada segundo três categorias gerais: *Basic Queries*, *Basic Queries with Clauses*, *Combination of queries*.

Após o processamento de linguagem, transformação para uma representação em triplas e classificação da consulta, o componente RSS (*Relation Similarity Service*) é invocado sob responsabilidade de transformar a tripla genérica em uma tripla lógica baseada na ontologia de domínio fornecida inicialmente como parâmetro para o AquaLog. O *Relation Similarity Service* possui um funcionamento ligeiramente distinto para cada classe de pergunta, entretanto, de forma geral o procedimento consiste em associar os termos da tripla às entidades da ontologia. Isso é feito com base em métricas de similaridade de texto e também por meio de consultas ao WordNet. Após, é realizada uma busca por relações entre os termos. Durante o procedimento de casamento entre os termos da consulta e as entidades da ontologia pode haver casos onde não seja possível identificar as relações. Nesses casos, o AquaLog emprega um mecanismo que permite aprender novos termos, a partir da colaboração com o usuário. Uma vez que os termos da consulta tenham sido relacionados aos elementos da ontologias o *Inference Engine* é invocado para obter as respostas. Para mais detalhes sobre o AquaLog consultar (LOPEZ; UREN; MOTTA; PASIN, 2007).

No projeto SWING (ROMAN; KLIEN; SKOGAN, 2006) as consultas são expressas tanto de forma estruturada como em texto livre. A especificação das consultas estruturadas é auxiliada por uma ferramenta visual que exhibe os conceitos e as relações semânticas de domínio. De forma geral, consultas em texto livre ou linguagem natural tem de ser interpretadas e anotadas semanticamente. Uma consulta semântica permite diminuir as ambiguidades das entradas, o que culmina em resultados mais precisos. Mais detalhes podem ser encontrados em (ROMAN; KLIEN; SKOGAN, 2006).

Em Gomes Jr. (2007) o universo de dados é constituído de serviços. Considerando especialmente os dados geoespaciais, estes são publicados conforme os padrões WMS e WFS do OGC. No SWING (ROMAN; KLIEN; SKOGAN, 2006) o universo de dados é composto unicamente por serviços geoespaciais conforme os padrões WMS, WFS e WPS do OGC. Já Viegas e Soares (2007) apresentam uma metodologia baseada em ontologias para consultar dados geoespaciais armazenados em um banco de dados relacional. Nessa abordagem as consultas são transcritas para SQL a partir de dedução lógica em ontologias de domínio.

3.6.3 Abordagem adotada

No caso da Pangea o universo de dados a ser consultado e descoberto é composto por serviços de dados e de operações geográficas. Esses serviços são anotados conforme o *framework* OWL-S e, portanto, a etapa de especificação e processamento da consulta devem levar em consideração as formalizações expressas nesse *framework*.

Diante do exposto, as consultas podem ser especificadas de duas formas:

- **Linguagem natural** – As consultas em linguagem natural são processadas de modo a serem transformadas em uma representação lógica que possa ser processada pelo mecanismo de resolução de consultas; e,
- **Estruturadas em SPARQL** – As consultas submetidas ao Pangea podem ser expressas em SPARQL. Essas consultas devem ser pré-processadas de modo a serem transformadas em um OWL-S *Service Profile*.

De forma geral o processo de especificar uma consulta consiste em criar uma representação formal do problema exposto pelo usuário na consulta, utilizando o OWL-S como *template* para representação de problemas. Já a etapa de processamento da consulta é baseada tanto em *Match* Sintático como Semântico. Essa etapa consiste em “casar” a consulta com as descrições de serviços OGC (i.e. *OWL-S Service Profiles* que descrevem os serviços OGC).

3.7 Descobrimto de serviços web semânticos

O descobrimto consiste no procedimento de localizar e recuperar serviços web com base na correspondência entre a consulta especificada pelo usuário/agente e as descrições semânticas associadas aos serviços web.

O procedimento de descobrimto é tipicamente empregado em cenários onde, por um lado, há serviços web responsáveis por disponibilizar acesso à informações e operações, e por outro, há uma ampla variedade de usuários/agentes, com perfis distintos, que utilizam as informações providas para sanar suas necessidades. Os aspectos relacionados com o descobrimto de serviços web semânticos são apresentados em mais detalhes na seção 3.7.1.

3.7.1 Contextualização e problemática

Com a evolução das tecnologias de microeletrônica e, sobretudo, de redes, a Web evoluiu de um sistema de troca de informações científicas entre poucas universidades para um repositório bastante rico e diversificado de informações. Além disso, passou a ser adotada de forma contundente como plataforma para sistemas de informação, de modo a suportar os requisitos da sociedade moderna, amplamente fundamentada na troca de informações em tempo real e em escala global. Com isso, grandes volumes de informações e processos vem sendo publicados na Web. Na última década, houve uma tendência em adotar serviços web como forma de disponibilizar informações e processos na Web.

Nesse cenário, um problema recorrente consiste em organizar e descobrir serviços web que satisfaçam uma consulta. Os sistemas de descobrimento de serviços web atuais são pouco precisos no que tange aos resultados retornados frente a uma consulta especificada pelo usuário/agente. Desse modo, requerem um esforço bastante grande por parte dos usuários/agentes na tarefa de filtrar os resultados irrelevantes. Por vezes, essa tarefa se torna entediante, acabando por inviabilizar a descoberta e o consumo dos serviços Web.

Portanto, prover mecanismos que permitem melhorar a eficiência do descobrimento de serviços web é de extrema relevância, considerando cenários onde haja grande quantidade de recursos disponíveis e uma ampla variedade de usuários/agentes que necessitam reutilizar tais recursos de maneiras diversificadas. Esse é justamente o cenário que caracteriza a Web e a Web Semântica atualmente.

Para possibilitar mecanismos de descobrimento mais precisos e menos custosos para os usuários/agentes, estão sendo empregadas ontologias na descrição das capacidades dos provedores de serviços web.

No entanto, encontrar descrições semânticas de serviços web que suportem os desejos expressos nas consultas dos usuário/agentes pode ser uma tarefa computacional que demanda tempo, pois, em geral, envolve processos de dedução lógica. Desse modo, é importante considerar dois aspectos (KELLER; LARA; POLERES; TOMA; KIFER; FENSEL, 2004) (FENSEL; KELLER; LAUSEN; POLLERES; TOMA, 2005):

- i. **Os usuários/agentes expressam suas necessidades de informação de forma bastante específica e detalhada** – É comum que os desejos expressos pelos usuários/agentes em suas consultas sejam detalhados e específicos. Por exemplo: um usuário/agente está interessado em obter informações sobre cinemas que estejam exibindo o filme *Whatever Works* em São

Paulo, ou então, informações sobre vôos entre São Paulo e Rio de Janeiro em 15/05/2010 às 18:00. Tais exemplos servem para ilustrar que os desejos expressos carregam consigo muitos detalhes. Os detalhes podem tornar o procedimento de descobrimento uma tarefa inviável considerando a escala da Web;

- ii. **Descrever as capacidades de um serviço web em detalhes pode levar à dificuldades em manter as descrições atualizadas e também à uma quantidade muito grande de descrições para um mesmo serviço** – Um serviço web que forneça informações sobre vôos pode incluir: a data, o horário e itinerário. Nesse caso os detalhes fornecidos pelo serviço web são informações que podem se alterar com facilidade. Essa característica acarreta impactos na manutenção das descrições dos serviços web. Considerando o exemplo, a cada mudança de horário, itinerário, data, as descrições teriam de ser atualizadas. Além disso, haveria uma quantidade bastante grande de descrições, contemplando cada vôo oferecido pelo serviço. Desse modo, fica claro que a quantidade de detalhes e os aspectos dinâmicos associados ao serviço podem prejudicar o desempenho do processo de descobrimento de serviços web.

Os aspectos apresentados em (i) e (ii) devem ser considerados em um modelo conceitual que suporte o descobrimento de serviços web em escala global.

No contexto das informações geográficas, os padrões de serviços OGC tem sido amplamente adotados pela comunidade de geoinformação para prover e compartilhar informações e processos geográficos na Web. Entretanto, o problema se repete: a grande quantidade de serviços OGC providos na Web torna o procedimento de descobrimento de serviços OGC uma tarefa custosa para o usuário e bastante complexa para agentes de software.

Como forma de organizar os provedores de informação geográfica e auxiliar o procedimento de descobrimento, o OGC estabelece o padrão de serviço web de Catálogo: o Catalog Service (CS). Apesar da especificação atual de catálogo do OGC ser bastante geral e flexível, grande parte das implementações permitem apenas descobrir serviços Web com base em palavras-chave, metadados e aspectos espaciais.

Com base no exposto, na seção 3.7.2 é apresentado um modelo conceitual que pode ser empregado para suportar o descobrimento de serviços web. Além disso, são discutidos trabalhos que abordam os métodos empregados para obter a correspondência entre uma descrição de um serviço web e uma consulta.

3.7.2 Trabalhos e conceitos relacionados

Nesta seção são apresentados os trabalhos e conceitos relacionados com a definição do módulo de descobrimento adotado na Pangea. Inicialmente é apresentado um modelo conceitual que serve aos propósitos do procedimento de descobrimento de serviços web. Após, são discutidos métodos e técnicas empregadas na etapa de correspondência entre as necessidades dos usuários/agentes e as descrições dos serviços web. Por fim, é apresentada em linhas gerais a abordagem adotada na Pangea.

3.7.2.1 Modelo conceitual para o descobrimento de serviços web semânticos

Há muitas propostas de técnicas e métodos para o descobrimento de serviços web, entretanto, há poucos trabalhos que abordam um modelo conceitual que suporte este processo. Nesta seção é apresentado um modelo conceitual (KELLER; LARA; POLERES; TOMA; KIFER; FENSEL, 2004) (FENSEL; KELLER; LAUSEN; POLLERES; TOMA, 2005) que pode ser utilizado como base para o processo de descobrimento de serviços web.

Como já mencionado, o descobrimento é o processo de localizar e recuperar serviços web, com base na correspondência entre a descrição dos serviços web e uma consulta. Nesse contexto, a consulta representa as necessidades de informações e processos dos usuários/agentes (KELLER; LARA; POLERES; TOMA; KIFER; FENSEL, 2004)(KLUSCH, 2008). Considerando os aspectos apresentados em 3.7.1, (KELLER; LARA; POLERES; TOMA; KIFER; FENSEL, 2004) (FENSEL; KELLER; LAUSEN; POLLERES; TOMA, 2005) propõem adotar como modelo conceitual para o descobrimento de serviços web a classificação heurística.

Conforme descrito por Clancey (1985) a classificação heurística é um tipo especial de classificação. De forma geral, o problema tratado pelos métodos de classificação consistem em classificar um objeto desconhecido frente a um conjunto de classes que representam estereótipos. Na classificação heurística o processo de classificar um objeto desconhecido é realizado a partir de três etapas:

- **Abstração** – Consiste na extração de características do caso real, de modo a permitir classificar o caso. Há três tipos de inferências utilizadas durante o processo de abstração: (i)

abstração de definição (ex. Se o objetivo é um restaurante, então deve possuir cardápio); (ii) abstração qualitativa (ex. Se o preço de um prato é menor do que 40, então é barato); (iii) generalização (ex. Se o objetivo do consumidor é um restaurante italiano, então o objetivo é um restaurante);

- **Matching** – Consiste no procedimento de dedução lógica realizado sobre o problema abstrato com o objetivo de encontrar soluções coerentes;
- **Refinement** – Procedimento de refinamento dos casos encontrados na etapa de *matching*. O objetivo é encontrar uma solução concreta para o problema abstraído a partir das soluções gerais recuperadas anteriormente.

Considerando o modelo conceitual de classificação heurística proposto por Clancey (1985), Keller, Lara, Poleres, Toma, Kifer e Fensel (2004) adaptam uma estratégia para o descobrimento de serviços web. Esta estratégia é apresentada na figura 12.

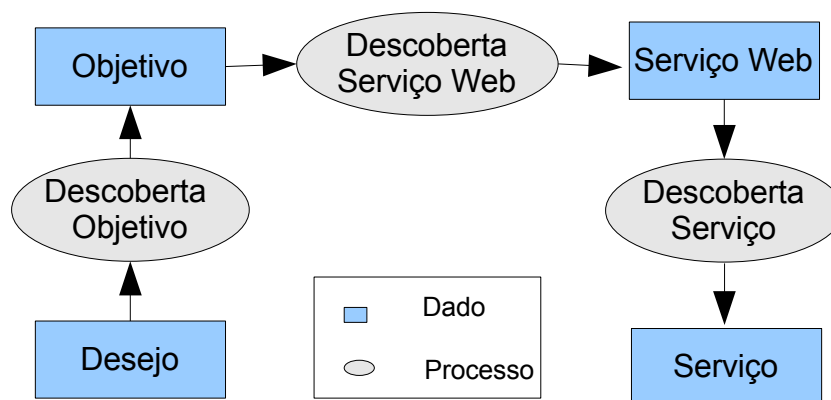


Figura 12. Modelo conceitual para descobrimento de serviços.

Fonte: Adaptado do trabalho de Keller, Lara, Poleres, Toma, Kifer e Fensel (2004)

Na abordagem proposta por Keller, Lara, Poleres, Toma, Kifer e Fensel (2004), o primeiro passo (i.e. Descoberta Objetivo na figura 12) é abstrair o desejo do usuário/agente, para permitir identificar um ou mais objetivos que representam, de forma abstrata, o desejo do usuário. Considerando o modelo de classificação heurística, esta etapa está relacionada à Abstração do caso (i.e. *Abstraction*).

Na sequência, o objetivo abstrato é utilizado para encontrar as correspondências com os serviços web. No modelo de classificação heurística esta etapa está relacionada à classificação da descrição geral do caso com classes de soluções previamente conhecidas (i.e. *Matching*).

Após, os serviços web que casam com a descrição geral do objetivo são refinados (i.e. Descoberta Serviço na figura 12) de modo a resultar em soluções concretas. Na classificação heurística esta etapa está relacionada com o refinamento das soluções gerais encontradas na etapa de *Matching* (i.e. *Refinement*). Este processo pode ser repetido inúmeras vezes durante um único procedimento de descobrimento de serviços.

A definição de um modelo conceitual e de uma estratégia de descobrimento permitem estabelecer as bases conceituais a partir das quais o módulo de descobrimento de serviços web na Pangea é definido. O módulo de descobrimento de serviços web é apresentado no Capítulo 4.

Uma vez expostos em linhas gerais o modelo conceitual e a estratégia de descobrimento, na seção seguinte são discutidos mecanismos empregados no procedimento de correspondência (i.e. *match*) entre as descrições dos serviços web e os desejos dos usuários.

3.7.2.2 Abordagens utilizadas para o *match* de serviços web

Recentemente tem sido desenvolvidos muitos trabalhos (PAOLUCCI; KAWAMURA; PAYNE; SYCARA, 2002) (KLUSCH; FRIES; SYCARA, 2006) (KLUSCH; KAUFER, 2009) (GOMES JR., 2007) (KLUSCH, 2008) nos quais o foco é o descobrimento de serviços web. Os diversos mecanismos de descobrimento de serviços web se diferenciam basicamente por três aspectos: (i) *Framework* utilizado para descrever o serviço web; (ii) Método empregado para obter a similaridade entre uma consulta e a descrição de um serviço web; (iii) Arquitetura (centralizada ou P2P) empregada no procedimento de descobrimento (KLUSCH, 2008). Um aspecto importante a ser considerado consiste nos aspectos da descrição do serviço web que devem ser considerados pelo descobrimento: entradas, saídas, pré-condições, pós-condições ou até mesmo modelo de processamento. Considerando esse último aspecto, os trabalhos apresentados nessa seção focam no procedimento de *match* com base nas entradas e saídas aceitas por um serviço web e requeridas em uma consulta.

Na abordagem apresentada por Paolucci, Kawamura, Payne, Sycara (2002) a similaridade entre a consulta e a descrição dos serviços web é medida levando em consideração os relacionamentos semânticos entre os conceitos que anotam as entradas e as saídas dos serviços e expressos na consulta. A correspondência (i.e. *Match*) entre a consulta e a descrição do serviço web é expressa em quatro níveis: (i) *Exato*; (ii) *PlugIn*; (iii) *Subsumes*; (iv) *Fail*. O nível exato é obtido

em duas situações: nos casos onde as saídas esperadas pela consulta são semanticamente equivalentes às saídas serviço web, e também nos casos onde a saída esperada pela consulta é subclasse da saída do serviço web. O nível *PlugIn* é obtido quando a saída do serviço web contém (i.e. É mais abstrata) a saída requerida pela consulta. Já o nível *Subsumes* é obtido se a saída esperada pela consulta contém (i.e. É mais abstrata) a saída do serviço. Por fim, caso não haja relação entre as saídas, o *match* retorna *Fail*. As descrições dos serviços web e as consultas são expressas em DAML-S (predecessor do *framework* OWL-S).

Klusch, Fries e Sycara (2006) propõem que o grau de similaridade entre os conceitos utilizados para descrever os serviços web e as consultas, deve levar em consideração a semântica formal e implícita. Os autores argumentam que técnicas baseadas apenas em lógica (como é o caso em (PAOLUCCI; KAWAMURA; PAYNE; SYCARA, 2002)) não consideram a semântica implícita e portanto estão limitadas. Inclusive, essa argumentação é amparada por experimentos conduzidos em (KLUSCH; FRIES; SYCARA, 2006). Nesta abordagem, são empregadas, além de lógica, técnicas de recuperação de serviços web com base no conteúdo da descrição semântica. Para tanto, são aplicadas métricas de similaridade entre textos. As abordagens que empregam lógica e técnicas de *information retrieval* – IR para descobrir serviços web são denominadas híbridas. Com isso, são especificadas 5 níveis de *match*, sendo os quatro primeiros baseados em lógica e a última em técnicas de IR: (i) *Exact*; (ii) *PlugIn*; (iii) *Subsumes*; (iv) *Subsumed-By*; (v) *Nearest-Neighbor*.

O *match* exato ocorre quando as entradas e saídas expressas na consulta e no serviço são equivalentes. Já o *PlugIn match* é obtido quando o serviço web possui menos entradas do que as fornecidas na consulta e sua saída é algo mais específico do que o solicitado. A classe *Subsumes* retorna serviços que fornecem saídas mais específicas do que as requeridas, porém, não restritas a um nível de hierarquia (diferentemente da classe *PlugIn*). A classe *SubsumedBy* ocorre quando as saídas fornecidas pelo serviço web são mais gerais do que as requeridas na consulta. Essas são as medidas de similaridade com base nas relações lógicas entre os conceitos. No caso dessas medidas falharem, é aplicada a classe *Nearest-Neighbor*, que é baseada nas métricas de similaridade entre textos.

Apresentadas as abordagens anteriores, na seção 3.7.3 é descrita em linhas gerais a abordagem empregada neste trabalho. No Capítulo 4 o componente de descobrimento da Pangea é apresentado em detalhes.

3.7.3 Abordagem adotada

O esquema de descobrimento de serviços web é baseado no modelo conceitual adaptado por Keller (figura 12). Considerando o apresentado na seção precedente, o *framework* utilizado para a descrição dos serviços web é o OWL-S (apresentado na seção 2.3.2). O método empregado para obter a correspondência entre as descrições dos serviços web e a consulta é híbrido, ou seja, baseado nas relações semânticas entre os conceitos e em métricas de similaridade de texto (similar à abordagem em (KLUSCH; FRIES; SYCARA, 2006)) e, a arquitetura empregada é centralizada. Além disso, é importante destacar que os aspectos considerados para o *match* são baseados nas entradas e saídas dos serviços web.

Considerando que o universo de busca é constituído por serviços geográficos o mecanismo de descobrimento da Pangea estende as abordagens apresentadas na seção 3.7.2, de modo a permitir a aplicação de uma restrição espacial (quando aplicável) para selecionar os serviços web que devem ser retornados ao usuário. Um esquema geral do procedimento de descobrimento de serviços adotado na Pangea é apresentado na figura 13.



Figura 13. Esquema geral de descobrimento de serviços adotado na Pangea.

3.8 Conclusões do capítulo

Neste capítulo foram apresentadas e analisadas propostas de arquiteturas relacionadas ao projeto da Pangea. Em decorrência desta análise, foram definidos e analisados os aspectos transversais encontrados em trabalhos correlatos. Na tabela 2 os aspectos transversais discutidos neste capítulo são sintetizados.

Tabela 2 - Síntese dos aspectos transversais discutidos.

Aspecto transversal identificado	Descrição do problema
Disponibilização de dados e modelos geográficos como serviços OGC	Este aspecto é considerado devido às diferenças sintáticas entre formatos de armazenamento e intercâmbio de dados e modelos geográficos.
Servidor semântico	Está relacionado às dificuldades existentes com o armazenamento, recuperação, dedução lógica e compartilhamento de ontologias na Web.
Anotação semântica	Este aspecto está relacionado ao problema de associar semântica às informações.
Integração de ontologias	Este aspecto está relacionado ao problema de encontrar correspondência semântica entre ontologias, haja visto que os dados são anotados por diversas ontologias, desenvolvidas por atores com perspectivas da realidade e objetivos distintos.
Descobrimto semântico de serviços web	Este aspecto é considerado para sanar as dificuldades que existem no descobrimto de serviços web, apenas levando em conta as descrições sintáticas associadas aos serviços.

Fundamentalmente, os aspectos transversais apresentados compõem a base de requisitos que devem ser suportados pelo projeto da arquitetura Pangea. Neste sentido, no Capítulo 4 é apresentado o projeto da arquitetura Pangea, em função de seus requisitos específicos e dos módulos funcionais.

Capítulo 4

Pangea

A Pangea consiste em uma arquitetura composta de módulos e serviços web que permite organizar semanticamente dados e modelos geográficos, de modo a potencializar o descobrimento, integração e reutilização dos mesmos.

Nesta seção são apresentadas as especificações do projeto da arquitetura Pangea. Inicialmente, são definidos os requisitos principais que devem ser suportados pela Pangea. Posteriormente, são apresentados os módulos, pacotes e serviços web que compõem a arquitetura.

4.1 Requisitos

Alguns dos requisitos que devem ser suportados pelo projeto da arquitetura Pangea foram definidos com base nos aspectos discutidos no Capítulo 3 e outros foram verificados durante o desenvolvimento deste trabalho. Os requisitos definidos considerando os aspectos discutidos no Capítulo 3 são:

- i. **Disponibilizar dados e modelos geográficos como serviços OGC** – Os dados e modelos geográficos considerados pela Pangea devem seguir as especificações de serviços OGC. Em especial, os padrões: WMS, WFS e WPS. Os padrões de serviços OGC são adotados de modo a estabelecer um padrão sintático para troca de dados e modelos. Esse requisito é atendido pelas arquiteturas estudadas e apresentadas no Capítulo 3;
- ii. **Disponibilizar aos usuários/agentes interfaces para descobrir, recuperar e realizar dedução lógica em ontologias** – As ontologias são artefatos centrais na Pangea. As ontologias são utilizadas para anotar, descobrir e compor os recursos geográficos. Desse modo, é importante que a Pangea disponibilize mecanismos que permitam descobrir, recuperar e realizar dedução lógica em ontologias. As arquiteturas estudadas propõem componentes que armazenam e disponibilizam acesso às ontologias utilizadas pelos usuários/agentes. Esse requisito está relacionado ao aspecto transversal (c – Servidor

Semântico);

- iii. **Organizar semanticamente os dados e modelos geográficos** – Os dados e modelos geográficos são disponibilizados aos consumidores por diversos provedores distribuídos na Web. Para facilitar o processo de descobrimento, os provedores publicam as descrições referentes aos dados e modelos em componentes denominados catálogos. Os catálogos disponibilizam interfaces e mecanismos que facilitam o descobrimento por parte dos consumidores. Entretanto, grande parte dos catálogos permitem apenas o descobrimento com base em palavras-chave e critérios espaciais, desconsiderando a semântica relacionada aos recursos. Nesse cenário, é fundamental que os dados e modelos geográficos sejam organizados semanticamente através da utilização de ontologias. A organização dos recursos geográficos por meio de descrição semântica (ex. Ontologias) permite obter melhores resultados nos processos de descobrimento, recuperação e integração de informação geográfica. Esse requisito está relacionado ao aspecto transversal (d – Anotação semântica);
- iv. **Permitir a especificação de consultas semânticas** – Uma vez que os dados e modelos estejam organizados com base em descrição semântica, a Pangea deve possibilitar aos consumidores especificar as consultas através de conceitos e relações definidas formalmente. As consultas especificadas por meio de conceitos formais são denominadas consultas semânticas. Utilizando as consultas semânticas, os consumidores podem expressar de maneira mais precisa as suas necessidades. Desse modo, a consulta possui menos ambiguidades e os resultados tendem a serem mais precisos. Esse requisito decorre dos aspectos transversais (f – Especificação e processamento de consultas) e (g – Descobrimento de serviços web semânticos);
- v. **Permitir o descobrimento de dados e modelos geográficos através de dedução lógica** – Conforme já abordado, as descrições dos dados e modelos geográficos são armazenadas em componentes catálogos. A partir das descrições, as informações e os modelos são descobertos pelos consumidores. Atualmente, a grande maioria dos catálogos permite apenas descobrir os dados por meio de palavras-chave ou então por critérios espaciais, o que acaba prejudicando o procedimento de descobrimento. Nesse cenário, uma solução é a associação de modelos semânticos formais (i.e. Ontologias) aos dados e modelos. Uma vez que haja semântica associada aos recursos, é possível realizar o descobrimento utilizando procedimentos baseados em dedução lógica. Esse requisito é baseado nos aspectos transversais (f – Especificação e processamento de consultas) e (g – Descobrimento de serviços web semânticos);

- vi. **Permitir a integração de dados e modelos geográficos** – Em muitos dos problemas do mundo real ocorre a necessidade de compor dados e modelos geográficos para permitir obter soluções para um problema. Nesse contexto, a Pangea deve permitir a composição de dados e modelos geográficos. Esse requisito decorre do estudo das arquiteturas (a – Arquiteturas para o compartilhamento de dados e modelos geográficos);
- vii. **Permitir integrar conceitos de ontologias distintas** – É comum que as ontologias utilizadas para anotar os recursos geográficos sejam heterogêneas em diversos aspectos. Como forma de relacionar conceitos de ontologias distintas, a Pangea deve contar com a funcionalidade de descobrir e representar as relações semânticas entre conceitos de ontologias distintas. Esse requisito é baseado no aspecto transversal (e – Integração de ontologias);
- viii. **Interpretar consultas em linguagem natural e reconhecer predicados geográficos** – A Pangea deve ser capaz de processar consultas em linguagem natural e reconhecer nas mesmas os predicados geográficos. Esse requisito decorre do estudo das arquiteturas (a – Arquiteturas para o compartilhamento de dados e modelos geográficos);

Os requisitos identificados durante o desenvolvimento são os seguintes:

- ix. **Organizar semanticamente os consumidores de dados e modelos geográficos** – A resolução de problemas complexos requer a cooperação entre atores com experiências e culturas diversificadas. Nesse caso, a cooperação pode ocorrer de diversos modos, mas fundamentalmente se dá por meio da troca e compartilhamento de informações. Para permitir a troca e o compartilhamento de informações de modo efetivo é necessário considerar as diferenças culturais e de perspectiva entre os consumidores e provedores. Com isso, é importante que os consumidores de informação e processos geográficos sejam organizados segundo perfis. Os perfis consistem em estereótipos pré-concebidos que permitem agrupar os consumidores semelhantes. A classificação de um consumidor em um ou mais estereótipos ocorre inicialmente a partir das informações básicas pessoais do consumidor, e evolui a partir da interação entre o consumidor e os componentes da Pangea. O objetivo da associação aos estereótipos é personalizar o funcionamento da Pangea e evoluir a arquitetura no sentido de recomendar dados e modelos geográficos;
- x. **Organizar as ontologias conforme sua generalidade e objetivo** – As ontologias são

utilizadas na Pangea para descrever os dados, modelos, consumidores e provedores. Cada ontologia utilizada na Pangea deve ser diferenciada de acordo com sua generalidade e objetivo. Com isso, as descrições semânticas suportadas pela Pangea, podem ser classificadas conforme os seguintes tipos: topo (mais gerais), domínio, perfil, aplicação, modelos geográficos e trilhas;

- xi.* **Facilitar a compreensão do espaço geográfico por meio da adoção de perspectivas semânticas** – A compreensão do espaço geográfico é complexa pois depende dos aspectos culturais e experiências peculiares aos diversos atores que percebem e atuam no espaço. Na ampla maioria dos casos, os atores são muito diversificados, bem como suas percepções e experiências. Com isso, é fundamental permitir que os dados e modelos que representam o espaço sejam organizados de acordo com as perspectivas inerentes aos diversos atores, dessa forma, facilitando o entendimento e o fluxo de informação entre as partes. As perspectivas semânticas são utilizadas para proporcionar mecanismos de interpretação do espaço compartilhados por um grupo de usuários. As perspectivas devem ser associadas aos estereótipos de usuários a fim de permitir que o fluxo de informação entre atores com experiências e culturas diversificadas ocorra de maneira mais fácil e automática;
- xii.* **Registrar as interações entre os consumidores e os provedores de dados** – Os consumidores de recursos geográficos expressam suas necessidades por meio de consultas. As consultas são resolvidas pelo Pangea e os dados, processos e modelos pertinentes são retornados ao consumidor. Após, o consumidor pode finalizar o ciclo da consulta, adicionando os dados retornados às suas preferências ou então proceder de modo a refinar a consulta. Todas essas interações são armazenadas em modelos formalmente especificados, denominados Trilhas. As trilhas são a memória das interações entre os consumidores e os provedores de dados;
- xiii.* **Suportar a descrição e execução de modelos geográficos formalmente especificados** – Muitos dos fenômenos naturais e sociais que ocorrem no espaço podem ser melhor descritos e entendidos por modelos. Tais modelos nos fornecem uma lógica de entendimento do espaço e de seus fenômenos, resultando em informações e simulações úteis ao processo de tomada de decisão. Nesse cenário, é importante permitir que os diversos consumidores e provedores sejam capazes de especificar, executar e compartilhar modelos geográficos de modo a potencializar sua criação, avaliação e reutilização.

Com base nos requisitos descritos, na seção seguinte são apresentados os módulos que compõem a Pangea.

4.2 Organização geral da arquitetura Pangea

Para suportar os requisitos definidos, a arquitetura Pangea, apresentada na figura 14, é composta de cinco módulos: (i) *Semantic Middleware*; (ii) *Semantic Utils*; (iii) *Semantic Server*; (iv) *Personalization Utils*; (v) *Language Utils*. Os módulos são descritos em linhas gerais a seguir:

- ***Semantic Middleware*** – O *Semantic Middleware* é o módulo responsável por suportar a interação entre os provedores e consumidores de recursos geográficos. A interação se dá por meio do processamento de consultas, que consiste no descobrimento de serviços web que satisfazem a consulta especificada pelos usuários/agentes;
- ***Semantic Utils*** - O *Semantic Utils* agrupa os pacotes que fornecem as funcionalidades de anotação semântica e integração de ontologias. As anotações semânticas são utilizadas pelo *Semantic Middleware* no procedimento de descobrimento de serviços web. Em um ambiente aberto é comum que os serviços web sejam anotados com conceitos provenientes de diversas ontologias. Nesse cenário, é fundamental que sejam encontradas e estabelecidas as relações semânticas entre os conceitos das ontologias utilizadas na anotação semântica. As funcionalidades de integração de ontologias tem o objetivo de encontrar e materializar as relações entre os conceitos;
- ***Semantic Server*** – O módulo *Semantic Server* é composto de pacotes responsáveis por prover interfaces para descobrir, recuperar e realizar dedução lógica em ontologias utilizadas na Pangea;
- ***Personalization Utils*** – O módulo *Personalization Utils* é constituído de pacotes responsáveis por suportar as funcionalidades de personalização. A personalização deve levar em conta o histórico de buscas realizadas pelo usuário/agente e, a partir disso, ser capaz de recomendar dados geográficos;
- ***Language Utils*** – O módulo *Language Utils* agrupa funcionalidades relacionadas ao processamento de linguagem natural. As consultas expressas em linguagem natural consistem em um dos tipos de consulta que devem ser suportados pela Pangea. Nesse sentido, há necessidade por processar esse tipo de consulta, de modo a criar uma

representação inequívoca da consulta a ser resolvida pelo *Semantic Middleware*.

Os serviços OGC, que são alvo das funcionalidades propostas para os módulos da Pangea, são publicados em servidores de dados geográficos na web por diversos atores (figura 14). Os módulos propostos para a Pangea interagem de modo a permitir anotar e descobrir serviços web de dados e processos geográficos.

Para tanto, o primeiro passo consiste em anotar semanticamente os serviços OGC. A anotação é realizada pelo pacote *SemanticAnnot* (figura 14). As anotações semânticas podem ser realizadas com conceitos de ontologias publicadas no *Semantic Server*, ou então fornecidos pelo usuário/agente que solicita o procedimento de anotação semântica. O módulo *Semantic Annot* anota semanticamente os serviços web da camada *OGC Services Layer* (figura 14). A camada *OGC Services Layer* (figura 14) consiste em servidores de dados geográficos na Web, que disponibilizam serviços OGC. As anotações semânticas são publicadas no *Semantic Server* (figura 14). O *Semantic Server* dispõe de métodos que permitem descobrir e realizar dedução lógica nas descrições semânticas armazenadas em seus repositórios. Com base nas anotações semânticas e nas ontologias armazenadas no *Semantic Server* as consultas são processadas.

O processamento das consultas é de responsabilidade do módulo *Semantic Middleware* (figura 14). As consultas são especificadas pelos usuários/agentes em dois formatos possíveis: (i) Linguagem Natural; (ii) SPARQL. A consulta fornecida pelo usuário/agente deve ser pré-processada de modo a ser representada de maneira inequívoca e padronizada. No caso da consulta ser especificada em linguagem natural, o *Semantic Middleware* interage com o módulo *Language Utils* (figura 14) para desambiguar a consulta. As consultas em SPARQL são pré-processadas pelos pacotes do *Semantic Middleware*. Como resultado do pré-processamento, se tem uma consulta representada como um grafo, onde são indicados quais os serviços web de dados geográficos que devem ser descobertos.

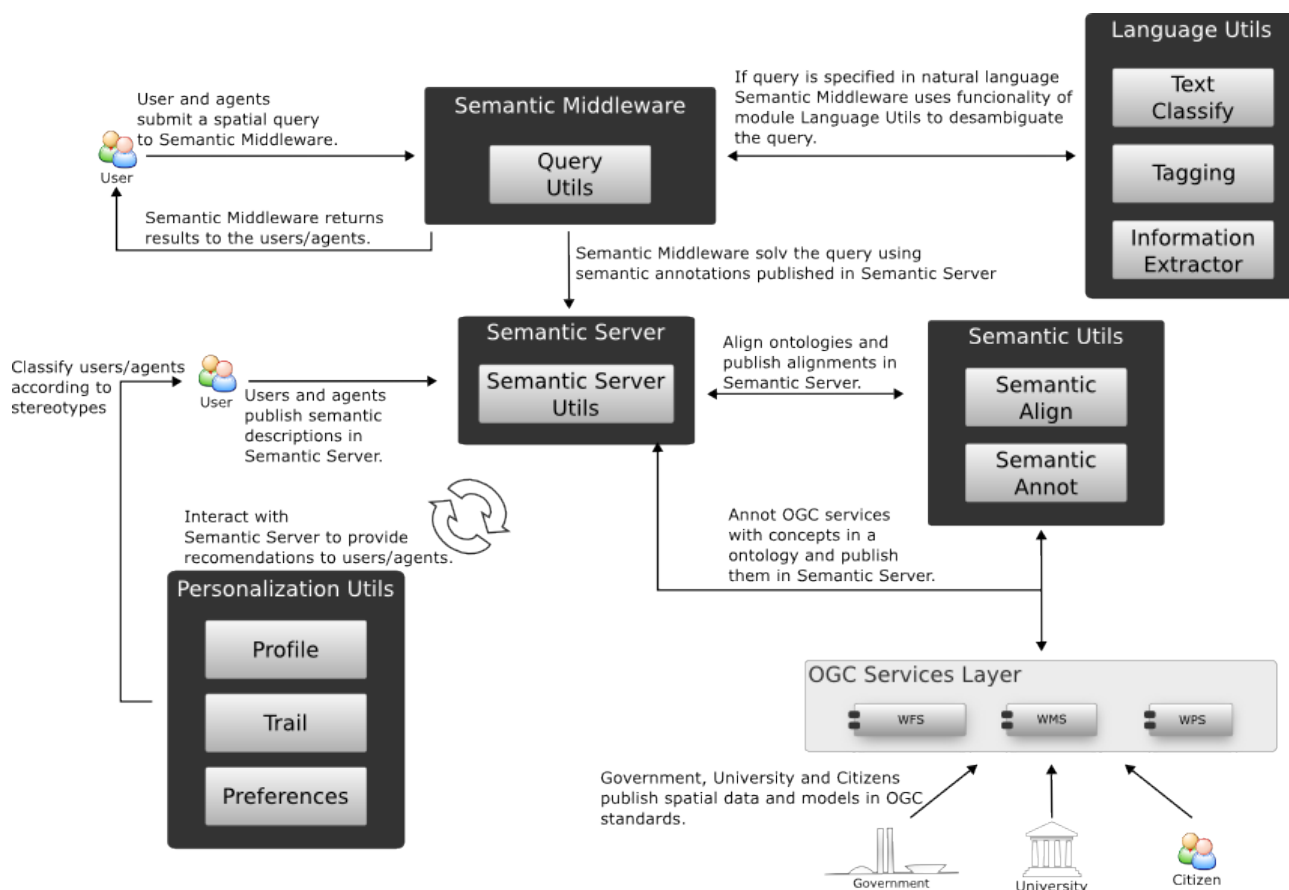


Figura 14. Esquema geral da Pangea.

Uma vez que a consulta tenha sido pré-processada, o *Semantic Middleware* interage com o *Semantic Server* para descobrir os serviços OGC que satisfazem os nós do grafo de consulta. As interações entre os usuários/agentes e a Pangea são registradas no *Semantic Server*. Tais interações são processadas pelo módulo *Personalization Utils* (figura 14). O objetivo principal do módulo *Personalization Utils* é fornecer recomendações de dados e modelos geográficos aos usuários/agentes.

Os módulos da Pangea são apresentados em detalhes nas próximas seções.

4.3 Módulo *Semantic Middleware*

Um *middleware* consiste em um componente computacional responsável por mediar a interação entre sistemas distintos. No contexto da Pangea, a mediação ocorre entre os provedores e os consumidores de recursos geográficos. Os provedores de informações são os atores responsáveis por coletar e publicar informações geográficas. Por outro lado, os consumidores de informação são

os atores que localizam e consomem as informações de acordo com suas necessidades. As necessidades de informação dos consumidores são especificadas por meio de consultas. Em síntese, o *middleware* deve possibilitar a interação entre os consumidores e os provedores de informação, por meio da resolução de consultas.

O *middleware* é caracterizado como semântico pois se baseia, fundamentalmente, em lógica descritiva e dedução para executar a mediação entre os provedores e os consumidores de informações. A lógica descritiva utilizada pelo *middleware* é representada em ontologias no padrão OWL-DL.

O *Semantic Middleware* é o módulo que reúne os pacotes, classes e serviços web responsáveis pelo processamento das consultas geográficas dos usuários/agentes. Nesta seção são apresentados em detalhes os pacotes e serviços web que compõem o *Semantic Middleware*. Na figura 15 são ilustrados os pacotes e serviços web que compõem este módulo.

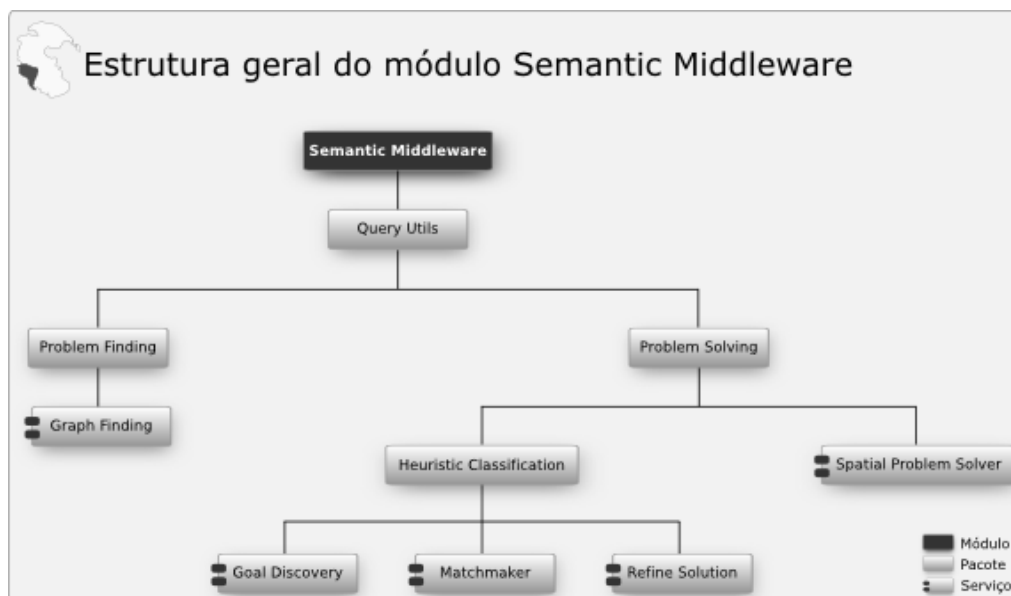


Figura 15. Estrutura geral do módulo *Semantic Middleware*.

4.3.1 Requisitos

O *Semantic Middleware* provê as funcionalidades necessárias para resolver as consultas geográficas dos usuários/agentes. Resolver as demandas por informações representadas nas consultas dos usuários/agente significa descobrir os serviços web que fornecem acesso às informações requeridas. O descobrimento de serviços web é definido como o processo de localizar uma descrição de um serviço web: (i) Que seja processável por máquina; (ii) Que não seja previamente conhecida; (iii) Que atenda alguns requisitos funcionais formalmente especificados. Com base no cenário do problema os requisitos que devem ser suportados pelo *Semantic*

Middleware são:

- (i) **Tratar consultas especificadas em linguagem natural e SPARQL** – Seres humanos utilizam linguagem natural para expressar suas necessidades de informação, enquanto agentes de software utilizam linguagens formais, como SPARQL. Com isso, o *Semantic Middleware* deve ser capaz de tratar consultas especificadas em linguagem natural e SPARQL;
- (ii) **Criar uma representação formal da consulta dos usuários/agentes independente da linguagem em que a consulta foi inicialmente especificada** – É importante garantir que as funcionalidades do *Semantic Middleware* sejam independentes da linguagem empregada na especificação da consulta. Essa característica garante que o *middleware* possa ser estendido de modo a tratar outras linguagens de consulta sem, no entanto, ser necessário alterar seus componentes internos de resolução de consultas;
- (iii) **Encontrar serviços web com base em descrições abstratas** – Em geral as consultas dos usuários/agentes refletem necessidades de informação bastante específicas (ex. Web Service que venda passagens de Campinas para o Rio de Janeiro, no dia 23/05/10 às 21:00). Encontrar a correspondência entre uma descrição específica de um problema e as descrições específicas dos serviços web pode ser inadequado, principalmente devido à grande quantidade de serviços web a serem considerados nesses casos. Como exemplo desse cenário, pode-se imaginar uma agência de viagens que provê serviços na ponte aérea entre cidades dos estados de São Paulo e do Rio de Janeiro. Nesse caso, se fossem consideradas as descrições concretas, seria necessário criar uma descrição semântica representando cada itinerário (i.e. Horário da viagem, data, entre outros). Isso levaria a uma quantidade muito grande de informação o que acabaria por tornar o descobrimento de serviços web uma tarefa inviável, considerando aspectos relacionados a tempo de resposta. Portanto, é fundamental considerar descrições abstratas no momento de realizar o “casamento” entre as descrições do objetivo e dos serviços web;
- (iv) **“Casar” as descrições de serviços web e do objetivo com base em relações semânticas e em técnicas de recuperação de informação** – O “casamento” entre as descrições dos objetivos e dos serviços web inicialmente é realizada com base nas relações semânticas. No caso do casamento com base em semântica não retornar nenhum resultado, devem ser aplicadas técnicas com base em similaridade de cadeias de texto para recuperar os serviços;

4.3.2 Serviços

As funcionalidades pertinentes ao *Semantic Middleware* são disponibilizadas como serviços agrupados no pacote *Query Utils*, que reúne as classes e os serviços web responsáveis pelo processamento de consultas. As consultas aceitas pelos serviços web do *Query Utils* podem ser especificadas semanticamente (i.e. Utilizando SPARQL), ou então em linguagem natural. Conforme já abordado, a resolução de consultas é dividida em duas etapas: (i) Especificação da consulta; (ii) Resolução da consulta. O pacote *Problem Finding* contém as classes e os serviços web responsáveis por realizar a especificação da consulta. Já o pacote *Problem Solving* reúne as classes e os serviços web responsáveis por processar a consulta formalmente especificada. Nas seções seguintes são apresentados os componentes dos pacotes *Problem Finding* e *Problem Solving*.

4.3.2.1 Problem Finding

O pacote *Problem Finding* é composto pelo serviço web *Graph Finding* que é o responsável por criar uma representação formal da consulta do usuário/agente, especificada em linguagem natural ou em SPARQL. O objetivo principal do *Graph Finding* é reconhecer na consulta os predicados e dados espaciais, e criar uma representação em grafo da consulta do usuário. Os nós do grafo podem ser de quatro categorias: (i) Dados geográficos; (ii) Operadores espaciais; (iii) Operadores booleanos; (iv) Modelos geográficos. Por exemplo, o grafo correspondente à consulta “*Praias com o Índice de Sensibilidade do Litoral superior a 4 em São Sebastião ou Mongaguá*” apresentado na figura 16.

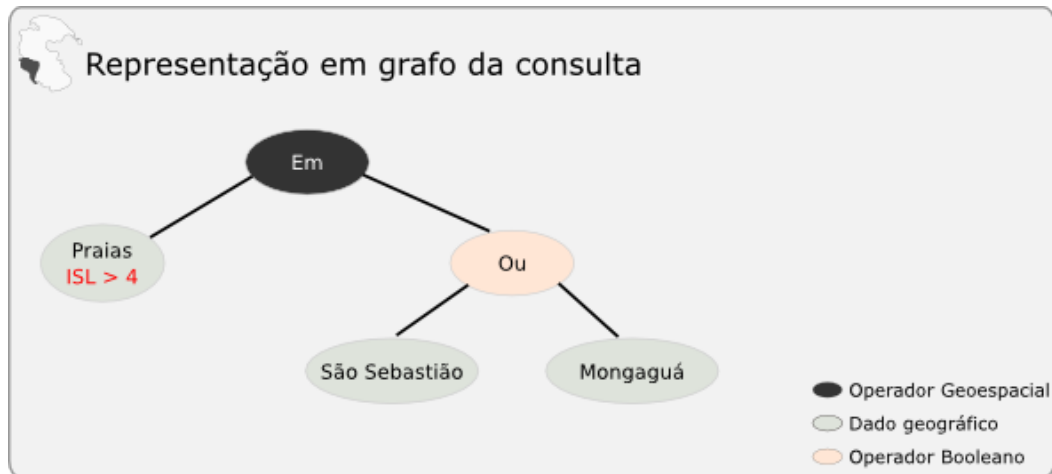


Figura 16. Representação em grafo da consulta.

Neste trabalho é considerado que a consulta em linguagem natural ou SPARQL já foi pré-processada e, portanto, sua representação em grafo está bem definida. Após o pré-processamento da consulta e da conseqüente representação em grafo, há condições necessárias para iniciar a etapa de resolução da consulta. Na próxima seção são apresentados os serviços web responsáveis por processar o grafo e obter os resultados.

4.3.2.2 Problem Solving

O pacote *Problem Solving* é proposto na arquitetura para reunir as funcionalidades referentes ao procedimento de resolução de consultas. No caso deste trabalho, o objetivo é a resolução de consultas espaciais utilizando conceitos relacionados ao método de classificação heurística. Nesse sentido, o pacote *Problem Solving* contém o serviço web *Spatial Problem Solver*, que é responsável por resolver as consultas utilizando os serviços web do pacote *Heuristic Classification*.

A organização proposta para o pacote *Problem Solving* permite que sejam adicionados e testados outros métodos de resolução de consultas no contexto da Pangea, sem impactar nos já projetados e implementados.

Nas próximas seções são apresentados os serviços web responsáveis pelo processamento da consulta.

a) Spatial Problem Solver

O serviço web *Spatial Problem Solver* é o responsável por processar os nós do grafo que representam a consulta do usuário/agente. Para tanto, conta com a operação *Solv*. A operação *Solv*, apresentada na tabela 3 e detalhada na figura 17, recebe como entrada o grafo que representa a consulta do usuário/agente e retorna os serviços web capazes de satisfazer a consulta. O descobrimento dos serviços web é realizado em três etapas: (i) *Goal discovery* (ver linha 5 da figura 17); (ii) *Matchmaker* (ver linha 10 da figura 17); (iii) *Refine Solution* (ver linha 11 da figura 17).

Tabela 3 - Descrição da operação *Solv* do serviço web *Spatial Problem Solver*.

Serviço web Spatial Problem Solver – Operação Solv	
Entradas:	SearchGraph (URL) – Referência para a consulta do usuário representada como um grafo, sem ambiguidades.
Saídas:	SolutionGraph (JSONArray) – Referências para os serviços web que podem satisfazer a consulta do usuário.

Pseudocódigo operação Solv
<pre>1: Solv (SearchGraph) 2: { 3: foreach (SearchGraph.node.equals(SPATIAL_DATA or SPATIAL_PROCESS)) 4: { 5: new goals = GoalDiscovery.discovery(SearchGraph.node); 6: 7: foreach (goals.goal) 8: { 9: new goal = goals.goal; 10: new webServices = Matchmaker.match(goal); 11: new solution = RefineSolution.refine(webServices); 12: 13: SolutionGraph.add(solution, SearchGraph.node); 14: } 15: } 16: } 17: return SolutionGraph; 18: } 19: }</pre>

Figura 17. Pseudocódigo da operação *Solv* do serviço web *Spatial Problem Solver*.

Nos próximos itens são apresentados os serviços web *GoalDiscovery*, *Matchmaker* e *RefineSolution*.

b) Goal Discovery

O serviço web *Goal Discovery* é o responsável por descobrir objetivos pré-existentes que “casam” com o objetivo expresso na consulta do usuário/agente. Os objetivos são descrições

abstratas que são reutilizadas no procedimento de resolução de consultas. Os objetivos são representados como uma subclasse do conceito *Profile*, especificado na ontologia *OWL-S Service Profile*.

Cada nó do grafo da consulta pode ser de 4 tipos distintos: (i) dados geográficos; (ii) operadores espaciais; (iii) operadores booleanos; (iv) modelos geográficos. Na etapa de descobrimento do objetivo são considerados especialmente os nós que representam dados e modelos geográficos (ver linha 3 da figura 17). Os dados e modelos geográficos expressos no grafo representam necessidades de informação do usuário/agente. Portanto, os dados e modelos representados no grafo da consulta são as saídas requeridas pelos usuários/agentes. Assim, o procedimento de descobrir um objetivo consiste em encontrar objetivos que tenham como saída esperada as saídas representadas no grafo.

Para suportar o descobrimento de objetivos o serviço web *Goal Discovery* contém a operação *discovery*. Essa operação recebe como entrada o nó do grafo de consulta em análise e retorna os objetivos que possuem como saída o dado/modelo geográfico requerido pelo nó.

A primeira etapa da operação *discovery* consiste em representar o nó do grafo como uma consulta SPARQL. Considerando que os objetivos são expressos como *OWL-S Service Profiles* a consulta SPARQL referente a um determinado nó do grafo de consulta deve representar uma tripla onde a propriedade *hasOutput* tenha o valor requerido no grafo. Considerando como exemplo o grafo da figura 16, o nó que contém o conceito “*Mongagua*” pode ser representado pela consulta SPARQL apresentada na figura 18.

```
Consulta SPARQL representando nó do grafo de consulta

1: PREFIX
2: owls: <http://www.daml.org/services/owl-s/1.1/Process.owl>
3: isl: <http://www.caiena.net/semanticweb/ontologies/isl.owl>
4: g: <http://www.caiena.net/semanticweb/ontologies/goal.owl>
5:
6:
7: SELECT ?goal
8: WHERE
9: {
10:
11:   ?goal owls:hasOutput isl:Mongagua .
12:   ?goal a g:Goal
13:
14: }
```

Figura 18. Consulta SPARQL para o descobrimento de um objetivo que representa a necessidade de informação expressa no nó do grafo de consulta.

A segunda etapa consiste na resolução da consulta (ver linha 10 da figura 19). A resolução da consulta SPARQL deve retornar todos os objetivos que possuem como saída o conceito solicitado. No exemplo anterior o conceito solicitado é “*Mongagua*”. No caso de não haver um objetivo pré-existente que satisfaça a consulta SPARQL, o conceito referente ao nó sendo analisado

é abstraído (com base nos axiomas da ontologia de domínio) e uma nova consulta SPARQL é produzida (ver linha 9 da figura 19). Se mesmo assim nenhum objetivo for encontrado, então é construído um novo objetivo para representar o nó em questão (ver linha 15 da figura 19). A construção do novo objetivo consiste na terceira etapa da operação. A descrição das entradas e saídas utilizadas na operação *discovery* é apresentada na tabela 4 e o pseudocódigo na figura 19.

Tabela 4 - Descrição da operação *Discovery* do serviço web *Goal Discovery*.

Serviço web Goal Discovery – Operação Discovery	
Entradas:	NodeGraph (URL) – Nó em análise do grafo que representa a consulta.
Saídas:	Goals (JSONArray) – Lista de objetivos que “casam” com o nó do grafo.

Pseudocódigo operação Discovery
<pre> 1: Discovery (NodeGraph) 2: { 3: new sparql = (SPARQL) NodeGraph; 4: 5: new goals = SemanticServer.GoalRepository.executeSPARQL(sparql); 6: 7: if (goals.length == 0) 8: { 9: new sparql = SemanticServer.GoalRepository.Abstract(NodeGraph); 10: goals = SemanticServer.GoalRepository.executeSPARQL(sparql); 11: } 12: 13: if (goals.length == 0) 14: { 15: goals = SemanticServer.GoalRepository.createGoal(NodeGraph); 16: } 17: 18: result.add(goals); 19: 20: return result; 21: }</pre>

Figura 19. Pseudocódigo da operação *Discovery* do serviço web *Goal Discovery*.

Os objetivos descobertos nesta etapa são utilizados no próximo passo que consiste em “casar” as descrições dos objetivos com as descrições dos serviços OGC. Essa funcionalidade é realizada pelo serviço web *Matchmaker*.

c) Matchmaker

Uma vez que um objetivo pré-definido ou criado tenha sido selecionado, o serviço web *Matchmaker* é o responsável por encontrar as correspondências entre a descrição do objetivo com as descrições dos serviços web. O algoritmo para encontrar as correspondências é baseado em métricas

de similaridade entre cadeias de texto e também nas relações semânticas entre os conceitos.

No serviço web *Matchmaker*, a operação *match* implementa o algoritmo para encontrar as correspondências. Essa operação recebe como entrada uma lista de objetivos e retorna uma lista de referências para serviços web que correspondem a cada objetivo (tabela 5).

O primeiro passo executado pela operação *match* consiste em selecionar as descrições dos serviços web que devem ser candidatas ao procedimento de “casamento”. Isso é feito para cada objetivo selecionado na etapa executada pelo serviço web *Goal Discovery* (descrito no item (b)). A seleção é baseada nas saídas esperadas pelo objetivo e é realizada por consultas SPARQL. Considerando um objetivo cuja saída esperada seja o conceito “*Mongagua*”, a consulta SPARQL correspondente para encontrar as descrições dos serviços web candidatas ao procedimento de “casamento” é apresentada na figura 20.

Após a execução da consulta SPARQL, pode ocorrer de não haver descrições de serviços web diretamente associadas aos conceitos de saída do objetivo em análise. Nesses casos, são recuperadas descrições de serviços web que tenham como saída conceitos que são mais gerais e mais específicos em relação a saída esperada pelo objetivo. Para selecionar os conceitos mais gerais e mais específicos, é considerado apenas um nível da taxonomia da ontologia (tanto no sentido da generalização como na especificação). Se, ainda assim, não forem encontradas descrições candidatas ao “casamento”, o serviço web *Matchmaker* retorna um conjunto solução vazio para o objetivo em análise.

```
Consulta SPARQL para recuperar os ServiceProfiles candidatos ao “casamento”
1: PREFIX
2: owls: <http://www.daml.org/services/owl-s/1.1/Process.owl>
3: isl: <http://www.caiena.net/semanticweb/ontologies/isl.owl>
4:
5:
6:
7: SELECT ?serviceProfile
8: WHERE
9: {
10:
11: ?serviceProfile owls:hasOutput isl:Mongagua .
12: ?serviceProfile a owls:Profile
13:
14: }
```

Figura 20. Consulta SPARQL para selecionar os *ServiceProfiles* candidatos ao procedimento de “casamento”.

Uma vez que tenham sido selecionados os *Service Profiles* candidatos ao “casamento”, o segundo passo consiste em verificar o grau de correspondência entre o objetivo e as descrições dos serviços web selecionados no primeiro passo. As definições de cada grau de correspondência adotadas são as mesmas definidas em Klush e colaboradores (KLUSCH; FRIES; SYCARA, 2006) e

são os seguintes:

- **Exact match** – S corresponde a G $\Leftrightarrow \forall inS \exists inG: inS = inG \wedge \forall outG \exists outS : outG = outS$;
- **Plug-in match** – S pluga em G $\Leftrightarrow \forall inS \exists inG: inS \supseteq inG \wedge \forall outG \exists outS : outS \in LSC(outG)$, onde LSC é o conjunto formado pelas subclasses do conceito outG;
- **Subsumes match** – G subsumes S $\Leftrightarrow \forall inS \exists inG: inS \supseteq inG \wedge \forall outG \exists outS : outG \supseteq outS$;
- **Subsumed-by match** – G subsumed-by S $\Leftrightarrow \forall inS \exists inG: inS \supseteq inG \wedge \forall outG \exists outS : (outS = outG \vee outS \in LGC(outG)) \wedge simIR(S,G) \geq \alpha$, onde LGC(outG) é o conjunto de conceitos mais abstratos e relacionados a outG;
- **Nearest-neighbor match** – S nearest-neighbor G $\Leftrightarrow \forall inS \exists inG: inS \supseteq inG \wedge \forall outG \exists outS : outG \supseteq outS \vee simIR(S,G) \geq \alpha$;
- **Fail** – Não é encontrada nenhuma correspondência entre a descrição do serviço web e o objetivo.

A descrição da operação *Match* é apresentada na tabela 5. Na figura 21 é apresentado o pseudocódigo da operação.

Tabela 5 - Descrição da operação *Match* do serviço web *Matchmaker*.

Serviço web Matchmaker – Operação Match	
Entradas:	Goals (JSONArray[URI]) – Objetivos candidatos ao match. Threshold (Float) – Limiar para ser considerado no nível de match Nearest-neighbor.
Saídas:	WebServices (JSONArray[Objects]) – Lista de serviços web que “casam” com os objetivos. Retorna também o grau de correspondência encontrado.

Pseudocódigo operação Match

```
1: Match ( Goals, threshold )
2: {
3:   foreach ( goals.goal )
4:   {
5:     new gOutput = goal.hasOutput;
6:     new sparql = (SPARQL) gOutput;
7:
8:     new candServiceProfiles = SemanticServer.executeSPARQL(sparql);
9:
10:    foreach ( candServiceProfiles.service )
11:    {
12:      if ( exact( candServiceProfiles.service ) ){ match = EXACT; break; }
13:      if ( plugin( candServiceProfiles.service ) ){ match = PLUGIN; break; }
14:      if ( subsumes( candServiceProfiles.service ) ){ match = SUBSUMES; break; }
15:      if ( subsumedBy( candServiceProfiles.service ) ){ match = SUBSUMEDBY;
16:        break; }
17:      if ( nearestneighbor( candServiceProfiles.service ) >= threshold )
18:        { match = NEAREST; break; }
19:
20:      match = FAIL;
21:    }
22:
23:    if (match != FAIL) { WebServices.add(goals.goal, candServiceProfiles, match,
24:      sim) }
25:  }
26:  WebServices.sort();
27:  return WebServices;
28: }
```

Figura 21. Pseudocódigo da operação *Match* do serviço web *Matchmaker*.

Uma vez que nesta etapa foram obtidos os serviços web e os respectivos graus de correspondência com os objetivos, na próxima etapa, os serviços web são refinados com base em suas características espaciais.

d) Refine Solution

O serviço web *Refine Solution* é o responsável por filtrar o conjunto solução levando em consideração os aspectos espaciais (i.e. *BoundingBox*) relacionados aos serviços web selecionados. Em toda consulta geográfica especificada pelos usuários/agentes é possível determinar uma área geográfica de interesse. Com base nessa área de interesse o conjunto solução pode ser refinado para retornar apenas os serviços web que provêm dados e modelos para a área geográfica específica. O serviço web *Refine Solution* é o responsável por essa funcionalidade. No entanto, esse serviço não será especificado em detalhes.

4.4 Módulo *Semantic Utils*

Um dos requisitos a ser suportado pela arquitetura Pangea consiste em descrever os dados e modelos geográficos utilizando conceitos de ontologias. A anotação semântica é o procedimento responsável por suportar tal requisito e, portanto, deve ser considerado no projeto da arquitetura.

No caso da anotação semântica, como já mencionado, é importante considerar o tipo de recurso que é alvo do processo de anotação. As características associadas ao tipo do recurso podem ser utilizadas como indicativos para o procedimento de anotação. Além do tipo, é fundamental considerar qual o esquema de metadados utilizado para descrever o recurso. O esquema de descrição adotado delimita quais os aspectos que devem ser considerados para descrever o recurso.

Diante do exposto, na Pangea, os recursos alvo da anotação semântica são serviços OGC, conforme os padrões: WMS, WFS e WPS. Considerando o esquema utilizado para descrever os recursos, é empregado o *OWL-S Service Profile*.

Desse modo, o procedimento de anotação semântica adotado na Pangea tem o objetivo de descrever os serviços OGC utilizando conceitos de ontologias de domínio. Os conceitos de domínio devem ser associados aos aspectos que são pertinentes ao padrão de descrição do *OWL-S Service Profile*. Cada *OWL-S Service Profile* gerado pelo processo de anotação deve ser armazenado no *Semantic Server*.

Em decorrência do tipo de dado e do esquema de descrição, o procedimento de anotação semântica adotado na Pangea é baseado nos seguintes aspectos:

- **Extração de metadados** – Serviços OGC possuem metadados associados. Os metadados podem ser recuperados por meio de chamadas às operações *GetCapabilities*, *DescribeProcess* e *DescribeFeatureType*. O procedimento de anotação semântica na Pangea utiliza os metadados associados ao serviço OGC para proceder a anotação semântica;
- **Enriquecimento semântico** – Serviços OGC de dados geográficos sempre estão relacionados à uma área geográfica de interesse. A área geográfica de interesse do serviço OGC é justamente a área a qual são pertinentes as *features* providas pelo serviço OGC. O processo de anotação semântica na Pangea considera uma etapa na qual são recuperados dados geográficos pertinentes à área geográfica do serviço OGC. Os dados são recuperados de provedores de informação geográfica da Web pré-cadastrados na arquitetura. Além de

recuperar os dados geográficos pertinentes à região de interesse, o processo de enriquecimento semântico também procura por triplas associadas aos conceitos da ontologia de domínio sendo utilizada na anotação semântica. Esse processo é realizado utilizando o Swoogle e repositórios internos ao *Semantic Server*. Ao final desse procedimento, se espera ter mais dados geográficos associados aos dados providos pelo serviço OGC. Além disso, se espera ter mais triplas compondo a ontologia de domínio;

- **Validação das anotações** – As anotações semânticas sugeridas devem ser validadas pelo usuário/agente que conduz o processo de anotação;
- **Dedução de novas anotações** – Com base nas anotações semânticas validadas pelo usuário/ agente são inferidas novas anotações. A inferência leva em consideração os axiomas da ontologia de domínio sendo utilizada no processo de anotação semântica.

Embora as anotações semânticas permitam organizar os dados e modelos geográficos com base em uma ontologia de domínio, não é possível considerar que exista uma ontologia que possa ser utilizada e compartilhada por todos os usuários e agentes. A situação real é que há várias ontologias de domínio sendo utilizadas para organizar os dados e modelos, de modo que se fazem necessários mecanismos que permitam encontrar os mapeamentos semânticos entre os conceitos dessas ontologias. A integração de ontologias é o procedimento responsável por encontrar e representar as relações semânticas entre ontologias.

Na Pangea o procedimento de integração de ontologias é multicritério e baseado nos seguintes aspectos:

- **Similaridade léxica** – A medida de similaridade entre duas entidades (classes, instâncias) das ontologias sendo integradas é baseada nos rótulos associados às entidades. Na Pangea são empregadas métricas de similaridade entre cadeias de texto para obter um índice para essa característica;
- **Similaridade semântica** – Nessa perspectiva a similaridade é medida com base no contexto semântico associado às entidades das ontologias. Na Pangea, são consideradas as relações de hierarquia para permitir estabelecer um grau de similaridade;
- **Similaridade espacial** – Nesse caso, a similaridade é obtida considerando a relação espacial

entre as entidades. Conceitos em uma ontologia que representam features geográficas podem ser transformados para as coordenadas geográficas que representam a localização da feature. Na Pangea, a localização espacial das entidades é utilizada para medir a similaridade entre entidades.

O módulo *Semantic Utils* agrupa os pacotes que suportam a **anotação semântica** e a **integração de ontologias**. O módulo é dividido em dois pacotes: (i) *Semantic Annot*; (ii) *Semantic Align*. O pacote *Semantic Annot* reúne os serviços web de anotação semântica de serviços OGC. O pacote *Semantic Align* reúne os serviços web para a integração de ontologias.

4.4.1 Anotação semântica

No contexto deste trabalho, o problema da anotação semântica consiste em encontrar as associações entre os conceitos de um domínio e os dados e modelos geográficos disponibilizados conforme os padrões WFS, WMS e WPS do OGC. O objetivo principal do estabelecimento dessa relação é permitir que os serviços OGC possam ser descobertos e integrados a partir de consultas mais elaboradas, tirando proveito da descrição semântica associada (Ontologia de domínio).

4.4.1.1 Requisitos

Considerando o problema exposto, e os aspectos discutidos no Capítulo 3, os requisitos gerais da anotação e que devem ser suportados pelo pacote de anotação são:

- (i) **Considerar textos em linguagem natural e as palavras-chave no processo de anotação semântica** – Para o procedimento de anotação devem ser considerados textos e palavras-chave utilizados na descrição dos serviços OGC. A associação entre os conceitos de domínio e esses textos deve ser realizada com base em métricas de similaridade entre cadeias de texto;
- (ii) **Considerar as características espaciais e os metadados dos serviços OGC** – Os metadados relacionados aos serviços OGC são obtidos através de chamadas aos métodos

GetCapabilities, *DescribeFeatureType*, *DescribeProcess*. Dos documentos resultantes das chamadas aos métodos devem ser extraídas as informações relevantes para a anotação, incluindo informações geoespaciais;

- (iii) **Considerar as relações topológicas e de distância das features** – A geometria das *features* e suas relações topológicas, e de distância devem ser consideradas no procedimento de anotação semântica;
- (iv) **Utilizar o modelo conceitual do OGC como base** – O modelo de referência do OGC (OGC REFERENCE MODEL 2002) deve ser utilizado como base para o processo de anotação com a finalidade de garantir o mínimo de alinhamento aos padrões OGC;
- (v) **O alvo da anotação devem ser serviços WMS, WFS e WPS** – Os serviços alvo da anotação semântica são aqueles que seguem os padrões WMS, WFS e WPS do OGC;
- (vi) **As ontologias de domínio devem derivar de conceitos mais gerais compartilhados (i.e. Ontologias de topo)** – Deve ser possível associar conceitos mais gerais aos dados e processos nos casos onde não sejam encontradas anotações semânticas no domínio. Portanto, é importante que os conceitos de domínio estejam associados a conceitos mais gerais;
- (vii) **Utilizar inferência para descobrir novas anotações** – Uma vez que um recurso tenha sido anotado, deve ser possível realizar inferência para descobrir mais anotações pertinentes ao recurso.

Os requisitos listados devem ser suportados pelo projeto do componente de anotação semântica da Pangea. Na seção 4.4.1.2 o projeto do anotador semântico é descrito em função dos serviços web que o compõem.

4.4.1.2 Serviços web

O pacote de anotação semântica consiste em um conjunto de serviços web que auxiliam os

usuários/agentes na tarefa de anotar semi-automaticamente um conjunto de dados geográficos utilizando conceitos de um domínio. O esquema do funcionamento geral é apresentado na figura 22.

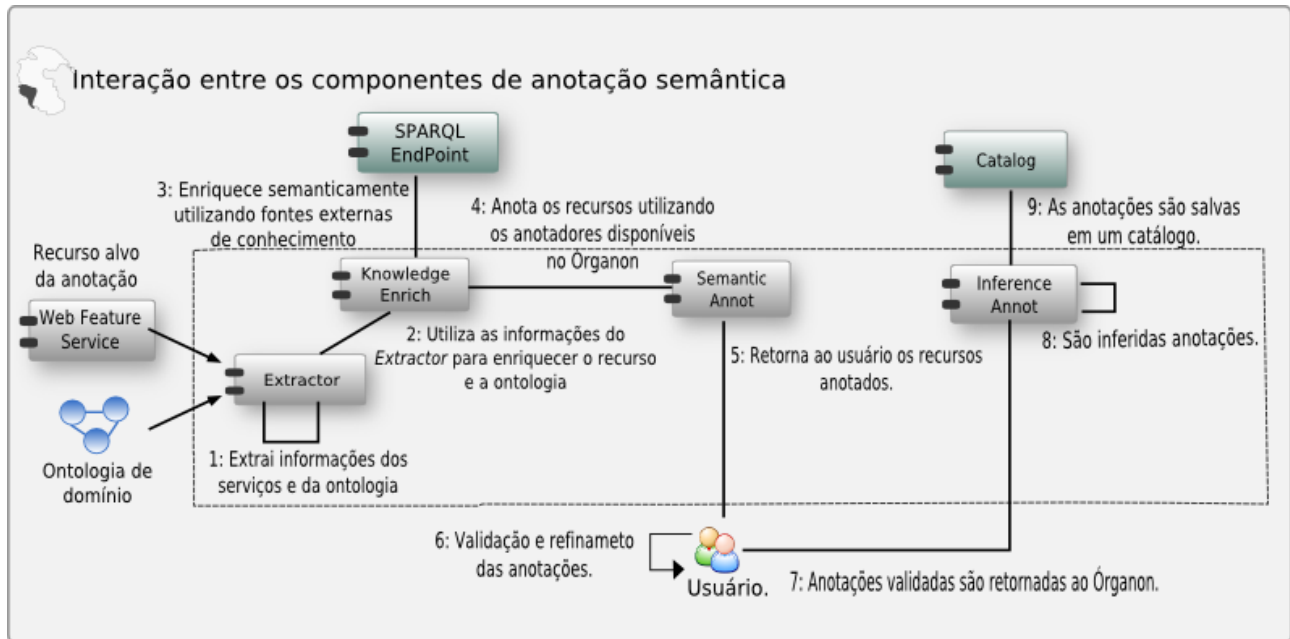


Figura 22. Esquema geral da anotação semântica.

Na tarefa de anotação o primeiro passo (na figura 22: serviço web *Extractor*) consiste em extrair informações do descritor do serviço OGC e da ontologia de domínio, além de aplicar uma transformação sintática para armazenar as informações extraídas em um esquema de representação interno. O segundo passo (na figura 22: serviço web *Knowledge Enrich*) possui dois focos: (i) Enriquecer semanticamente as triplas da ontologia de domínio; (ii) Agregar informações geográficas pertinentes ao limite geográfico de interesse, associado ao serviço OGC sendo anotado. Isso é feito no terceiro passo, através de consultas a *SPARQL EndPoints* pré-estabelecidos. O objetivo é recuperar informação semântica relacionada aos conceitos da ontologia de domínio e dados geográficos associados aos limites geográficos das *features* geográficas que são alvo da anotação.

Após o enriquecimento semântico, há condições necessárias para a execução do quarto passo, o serviço web de anotação (na figura 22: *Semantic Annot*). Há quatro serviços web de anotação semântica que podem ser utilizados e combinados de acordo com a natureza do serviço OGC sendo anotado, ou seja, serviço de dados e mapas (i.e. WFS, WMS) ou serviço de processamento (i.e. WPS).

No quinto passo, as sugestões de anotação são retornadas ao usuário para validação. Após a validação, no sexto passo (na figura 22: *Inference Annot*) são inferidas novas anotações com base nos axiomas da ontologia de domínio. Finalmente, as anotações semânticas produzidas são

instanciadas como OWL-S *Service Profiles* de acordo com o padrão OWL-S. Nas seções seguintes, os passos e os serviços responsáveis pela anotação semântica são descritos em detalhes.

a) **Extractor**

O serviço web *Extractor* é o responsável por extrair os metadados dos serviços OGC e transformar o documento de metadados do serviço OGC em uma ontologia, aplicando apenas uma transformação sintática (XML/GML para XML/OWL-DL). O *Extractor* possui a operação *Extract* que recebe como entrada a referência para o serviço OGC alvo da anotação e a referência para a ontologia de domínio que fornece os conceitos para a anotação. Como saída é esperada a ontologia do serviço (i.e. *Service Ontology*), que contém os metadados extraídos do serviço OGC.

Há duas classes de serviços suportadas pelo *Extractor*: (i) Serviços de dados geográficos e mapas (i.e. WFS e WMS) (na figura 23: considerar linhas 6 até 20); (ii) Serviços de processamento de dados geográficos (i.e. WPS) (na figura 23: considerar linhas 23 até 47). Para os serviços do tipo (i) os metadados são recuperados a partir de chamadas ao método *GetCapabilities* (na figura 23: considerar linha 3) e ao *DescribeFeatureType* (na figura 23: considerar linhas 15 e 16). Para os serviços do tipo (ii) os metadados são recuperados por meio de chamadas aos métodos *GetCapabilities* (na figura 23: considerar linha 3) e *DescribeProcess* (na figura 23: considerar linha 25).

Os metadados retornados contém diversas informações, entre elas: palavras-chave, restrições de acesso, operações disponíveis, formatos aceitos (i.e. GML, KML), *feature types*, processos, entradas e saídas, entre outras. No caso dos serviços WFS, é relevante considerar a lista de *feature types* disponibilizadas pelo serviço. Para cada *feature type*, são extraídas as seguintes informações: seu nome, sistema de referência de coordenadas (SRS) e limites geográficos (i.e. *BoundingBox*). Além disso, é importante considerar a descrição do esquema dos dados (i.e. *feature type schema*). O *feature type schema* é recuperado a partir da chamada ao método *DescribeFeatureType* e contém informações sobre os atributos que descrevem as *features*. Para cada *feature type* o esquema é recuperado.

Considerando serviços WPS, é importante extrair a lista de operações disponíveis no serviço OGC e, para cada operação disponível, suas entradas e saídas. Para cada operação são recuperadas as seguintes informações: identificação, título, resumo, metadados, entradas e saídas suportadas. Na tabela 6 é apresentada a estrutura geral do serviço web *Extractor* e na figura 23 é descrito o pseudocódigo da operação *Extract*, responsável por extrair os metadados dos serviços OGC.

Tabela 6 - Descrição da operação *Extract* do serviço web *Extractor*.

Serviço web Extractor – Operação Extract	
Entradas:	<p>OGC Service (URL) – Referência para um serviço OGC do tipo WMS, WFS ou WPS.</p> <p>DomainOntology (URL) – Ontologia de domínio que fornece os conceitos para a anotação semântica</p>
Saídas:	<p>service_ontology (URL) – Referência para a ontologia que descreve o serviço OGC.</p>

Pseudocódigo operação Extract	
<pre> 1: Extract(OGCSERVICEURI, domainOntology) 2: { 3: capabilities = OGCSERVICEURI.getCapabilities(); 4: service_ontology = null; 5: 6: if (OGCSERVICEURI.equals(WMS OR WFS)) 7: { 8: foreach (capabilities.FeatureType) 9: { 10: new FeatureType; 11: 12: FeatureType.name = capabilities.FeatureType.name; 13: FeatureType.SRS = capabilities.FeatureType.SRS; 14: FeatureType.BBox = capabilities.FeatureType.BoundingBox; 15: FeatureType.schema = 16: OGCSERVICEURI.describeFeatureType(FeatureType.name); 17: 18: service_ontology.add(FeatureType); 19: } 20: } 21: else 22: { 23: if (OGCSERVICEURI.equals(WPS)) 24: { 25: processDescription = OGCSERVICEURI.DescribeProcess() 26: 27: foreach (capabilities.Process) 28: { 29: new Process; 30: 31: Process.id = capabilities.Process.Identifier; 32: Process.title = capabilities.Process.Title; 33: Process.abstract = capabilities.Process.Abstract; 34: 35: foreach(capabilities.Process.Metadata) 36: { 37: new Metadata = capabilities.Process.Metadata; 38: Process.metadata.add(Metadata); 39: } 40: 41: pDesc = processDescription.getProcess(Process.id); 42: Process.inputs = processDescription.getInputs(); 43: Process.outputs = processDescription.getOutputs(); 44: 45: service_ontology.add(Process); 46: } 47: } 48: } 49: result.add (service_ontology); 50: }</pre>	

Figura 23. Pseudocódigo da operação *Extract*.

Considerando os requisitos apresentados na seção 4.4.1.1, o serviço web *Extractor* satisfaz os requisitos ii, iv, v e vi. Na sequência, os dados e metadados são utilizados pelo serviço web *KnowledgeEnrich* com a finalidade de agregar semântica e informações geográficas, respectivamente à ontologia de domínio e aos dados geográficos.

b) KnowledgeEnrich

O serviço web *Knowledge Enrich* é responsável por agregar informações geográficas e semânticas às ontologias de serviço (i.e. *Service Ontology*) e de domínio. Um esquema geral do processo de enriquecimento semântico é apresentado na figura 24.

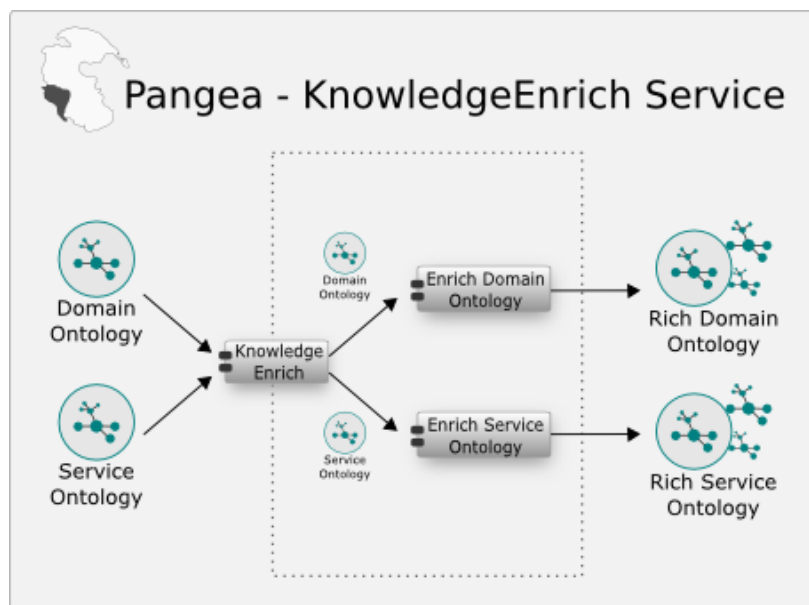


Figura 24. Esquema geral do enriquecimento semântico.

Considerando a ontologia de serviço (i.e. *Service Ontology*) (considerar figura 26), o enriquecimento semântico pode proceder de duas formas distintas, dependendo do tipo de serviço OGC que é alvo da anotação: (i) Serviços de dados e mapas (i.e. WFS e WMS); (ii) Serviços de processamento (i.e. WPS).

No caso dos serviços de dados (na figura 26: considerar linhas 10 até 38), a ontologia de serviços (i.e. *Service Ontology*) contém em seus conceitos as informações referentes aos limites geográficos (i.e. *BoundingBox*) dos dados providos. Com isso, é possível proceder o enriquecimento semântico com base nos limites geográficos do serviço (na figura 26: considerar linhas 17, 19 e 23). Nesse cenário, o objetivo principal é fornecer um contexto geográfico ao usuário que vai validar as anotações semânticas. O contexto geográfico é composto de pontos de interesse, malha viária, dados sócio-econômicos, limites geopolíticos, recursos naturais, entre outros. As informações são

recuperadas de provedores de conteúdo pré-estabelecidos na Pangea, disponíveis através de serviços web. Por outro lado, nos serviços de processamento, o objetivo é associar sua identificação, entradas e saídas aos conceitos de ontologias armazenadas no servidor semântico.

No que concerne à ontologia de domínio (i.e. *Domain Ontology*) (considerar figura 25), o enriquecimento é realizado com base no servidor semântico e em provedores externos. Nesse caso, o enriquecimento consiste na associação de outras descrições semânticas aos conceitos de domínio.

O processo de enriquecimento semântico é realizado pela operação *Enrich* que recebe como entradas: a ontologia de serviço e a ontologia de domínio. Internamente, a ontologia de domínio é passada como parâmetro para o serviço web *EnrichDomainOntology*, enquanto a ontologia de serviço é passada como parâmetro para o serviço web *EnrichServiceOntology*. Como saídas, se tem as ontologias enriquecidas (i.e. *Rich Domain Ontology* e *Rich Service Ontology*) com as informações recuperadas dos provedores de conteúdo pré-estabelecido.

Como já mencionado, o critério para a recuperação das informações geográficas associadas aos serviços de dados geográficos é baseado fundamentalmente no limite geográfico (i.e. *BoundingBox*) da *feature type* sendo anotada. Ou seja, são recuperadas dos provedores de conteúdo, informações geográficas restritas à localização geográfica e/ou vizinhança da *feature type* sendo anotada. As informações geográficas são extraídas dos seguintes serviços:

- **Geonames (GEONAMES, 2009)** – O Geonames consiste em um banco de dados geográfico que contém mais de 8 milhões de nomes geográficos. Sua base pode ser acessada através de diversos serviços web. Para o procedimento de anotação semântica, é relevante recuperar as cidades e os lugares que ocorrem dentro do *BoundingBox* da *FeatureType* sendo anotada e também em sua vizinhança. A contextualização através de locais, cidades, país, vizinhança, facilita o referenciamento para o usuário e contribui com o *feedback* do mesmo durante o procedimento de anotação;
- **DBPedia (DBPEDIA, 2009)** – O DBpedia consiste de um banco de dados semântico que organiza os dados da Wikipedia e permite recuperar estes dados por meio de consultas mais sofisticadas. No DBpedia há cerca de 300.000 *features* geográficas catalogadas e disponibilizadas para acesso via SPARQL EndPoints. Para a anotação semântica é importante recuperar informações semânticas relacionadas com o *BoundingBox* da *feature type* alvo da anotação;
- **OpenStreetMaps (OPENSTREETMAPS, 2009)** – O *OpenStreetMaps* consiste de uma iniciativa da comunidade para disponibilizar de forma livre dados de navegação (i.e. Ruas e

Pontos de Interesse). O *OpenStreetMaps* disponibiliza seus dados por meio de serviços web.

No caso da ontologia de domínio, a recuperação de informação semântica é realizada em repositórios de conceitos previamente cadastrados no *Semantic Server* (apresentado em detalhes na seção 4.5). Os conceitos da ontologia de domínio são utilizados para recuperar informações semânticas relacionadas. As informações semânticas agregadas permitem enriquecer a descrição de domínio previamente fornecida para o procedimento de anotação semântica. Para a anotação da ontologia de domínio são considerados repositórios de duas naturezas:

- **Repositórios internos** – Consistem nos repositórios previamente cadastrados no *Semantic Server* e que possuem alta relevância para o procedimento de anotação. Comumente, estes repositórios contém os conceitos de topo com os quais os conceitos de domínio se relacionam. Os aspectos relacionados aos repositórios semânticos são apresentadas em detalhes na seção 4.5;
- **Repositórios externos** – São serviços web que permitem recuperar descrição semântica. Um serviço dessa natureza bastante utilizado é o Swoogle (SWOOGLE, 2009).

Ambos os repositórios são utilizados para descobrir novas relações semânticas para os conceitos do domínio. Essas relações são exploradas por outros componentes deste módulo para estabelecer as anotações semânticas.

Considerando a ontologia de serviços WPS, o enriquecimento é realizado apenas levando em conta os repositórios internos. Neste caso, são associados conceitos aos termos que descrevem os seguintes aspectos do processo: identificação, entradas e saídas. Os conceitos são associados por cálculo de similaridade através da distância de edição. Diante do exposto, nas Tabelas 7 e 8 são apresentados, respectivamente, detalhes sobre a operação *Enrich* dos serviços *EnrichDomainOntology* e *EnrichServiceOntology*. Nas figuras 25 e 26 são apresentados o pseudocódigo das operações.

Tabela 7 - Descrição da operação *Enrich* do serviço web *Enrich Domain Ontology*.

Serviço web Enrich Domain Ontology – Operação Enrich	
Entradas:	Domain_Ontology (URI) – Referência para a ontologia de domínio.
Saídas:	rich_domain_ontology (URI) – Ontologia de domínio enriquecida semanticamente.

Tabela 8 - Descrição da operação *Enrich* do serviço web *Enrich Service Ontology*.

Serviço web Enrich Service Ontology – Operação Enrich	
Entradas:	Service_Ontology(URI) – Ontologia do serviço.
Saídas:	rich_service_ontology (URI) – Ontologia de serviços enriquecida semanticamente.

Pseudocódigo operação Enrich – Serviço web EnrichDomainOntology	
<pre> 1: Enrich(Domain_Ontology) 2: { 3: new richDomainOntology; 4: new semanticRepository; 5: new result; 6: 7: foreach (Domain_Ontology.class) 8: { 9: richDomainOntology.add(10: SemanticRepository.getSemanticEnrich(Domain_Ontology.class) 11:); 12: } 13: 14: result.add(richDomainOntology); 15: 16: return result; 17: }</pre>	

Figura 25. Pseudocódigo da operação *Enrich* do serviço web *EnrichDomainOntology*.

Pseudocódigo operação Enrich – Serviço web Enrich Service Ontology

```
1: Enrich( Service_Ontology )
2: {
3:   new Geonames;
4:   new OpenStreetMaps;
5:   new Dbpedia;
6:   new Swoogle;
7:   new SemanticRepository;
8:   new richServiceOntology;
9:
10:  if ( Service_Ontology.service.equals( WFS ) )
11:  {
12:
13:    foreach ( Service_Ontology.FeatureType )
14:    {
15:      new FeatureType = Service_Ontology.FeatureType;
16:      new BoundingBox = FeatureType.bbox;
17:      new geonamesData = Geonames.getData( BoundingBox, 'LIMIT 100');
18:      new openStrData =
19:        OpenStreetMaps.getData(BoundingBox, 'LIMIT 100');
20:
21:      foreach( geonamesData.locale )
22:      {
23:        new dbpediaData = Dbpedia.getData( geonamesData.locale ,
24:          'LIMIT 100' );
25:        richServiceOntology.add( dbpediaData );
26:      }
27:
28:      foreach( openStrData.POI )
29:      {
30:        new dbpediaData = Dbpedia.getData( openStrData.POI, 'LIMIT
31:          100' );
32:        richServiceOntology.add( dbpediaData );
33:      }
34:
35:      richServiceOntology.add( geonamesData );
36:      richServiceOntology.add( openStrData );
37:    }
38:  }
39:
40:  if ( Service_Ontology.service.equals( WPS ) )
41:  {
42:    foreach ( Service_Ontology.Process )
43:    {
44:      new Process = Service_Ontology.Process;
45:      new idConcept =
46:        SemanticRepository.getSimilarEntity( Process.id );
47:
48:      richServiceOntology.add(idConcept);
49:
50:      foreach ( Process.inputs )
51:      {
52:        new inputConcept =
53:          SemanticRepository.getSimilarEntity( Process.input );
54:        richServiceOntology.add(inputConcept);
55:      }
56:
57:      foreach ( Process.outputs )
58:      {
59:        new outputConcept =
60:          SemanticRepository.getSimilarEntity( Process.output );
61:        richServiceOntology.add(outputConcept);
62:      }
63:    }
64:  }
65: }
```

Figura 26. Pseudocódigo da operação *Enrich* do serviço web *EnrichServiceOntology*.

Considerando os requisitos apresentados na seção 4.4.1.1, o serviço *KnowledgeEnrich* satisfaz os requisitos i, ii, iv, v e vi. Finalizada esta etapa há condições necessárias para a execução

dos serviços de anotação. Os serviços de anotação são apresentados nos itens (c), (d), (e) e (f).

c) Semantic Annot

O *Semantic Annot* é composto por quatro serviços web de anotação semântica: (i) *Process Annot*; (ii) *Spatial Output Annot*; (iii) *Text Similarity Annot*; (iv) *Inference Annot*. O serviço web *Process Annot* é responsável por anotar semanticamente os serviços WPS. A anotação é realizada a partir do alinhamento entre os conceitos da ontologia de domínio e da ontologia de serviço enriquecidas (verificar serviço web *Knowledge Enrich*). O serviço web *Spatial Output Annot* realiza a anotação semântica das saídas dos serviços WFS. Para tanto, considera características e relações espaciais expressas na ontologia de domínio e verificadas nos dados sendo anotados. Para os casos onde não há características espaciais a serem consideradas, é proposto o serviço web de anotação *Text Similarity Annot*. Este emprega a métrica de distância de edição para anotar semanticamente serviços WFS. Finalmente, o serviço web *Inference Annot* aplica processos de dedução lógica nos axiomas da ontologia de domínio para obter novas anotações semânticas. Nas próximas seções os serviços web de anotação são detalhados.

c.1) ProcessAnnot

O serviço web *ProcessAnnot* é o responsável por anotar semanticamente os serviços web de processamento (WPS) conforme o padrão do *framework* OWL-S. O OWL-S organiza a descrição de um serviço web a partir de quatro perspectivas: (i) *Process Model*; (ii) *Profile*; (iii) *Grounding*; (iv) *Service* (OWL-S, 2004).

A escolha de qual perspectiva utilizar para descrever o serviço web está intimamente relacionada aos mecanismos de descobrimento que serão suportados pela anotação semântica realizada (KLUSCH, 2008). Uma anotação semântica baseada em processo (i.e. *Process Model*), permite descobrir serviços por meio das entradas, saídas e das relações entre elas. Por outro lado, uma anotação baseada em *Profile*, permite apenas verificar entradas e saídas, para determinar se um serviço web satisfaz uma consulta. O procedimento de anotação semântica adotado no *Process Annot* é baseado apenas nas entradas e saídas do serviço web. Portanto, a anotação semântica cria um OWL-S *Service Profile* do serviço WPS sendo anotado.

Para suportar essa funcionalidade, o serviço web *ProcessAnnot* contém a operação *createServiceProfile*. Essa operação recebe como entradas a ontologia de serviços enriquecida (resultante do processo de enriquecimento semântico) (*Rich_Service_Ontology*), a ontologia de

domínio (*Rich_Domain_Ontology*) e um limiar (*threshold*) que permite decidir se uma anotação deve ou não ser sugerida. A partir da ontologia de serviço enriquecida, é criado o *ServiceProfile* conforme o padrão OWL-S. A associação entre os conceitos da ontologia de serviço e os conceitos da ontologia de domínio é realizada com base na métrica proposta por Stoilos, Stamou e Kollias (2005). De modo geral, o problema da criação do *ServiceProfile*, nesse caso, é encarado como o alinhamento entre os conceitos da ontologia de serviços e da ontologia de domínio. A descrição da operação *createServiceProfile* é apresentada na tabela 9 e o pseudocódigo é descrito na figura 27.

Tabela 9 - Descrição da operação *createServiceProfile* do serviço web *Process Annot*.

Serviço web Process Annot – Operação createServiceProfile	
Entradas:	<p>Rich_Service_Ontology(URL) – Ontologia do serviço WPS enriquecida semanticamente.</p> <p>Rich_Domain_Ontology(URL) – Ontologia de domínio WPS enriquecida semanticamente.</p> <p>Threshold (Float) – Limiar considerado para que uma anotação semântica seja materializada.</p>
Saídas:	<p>Service_Profile (ServiceProfile) – Profile do serviço WPS conforme o padrão OWL-S.</p>

Pseudocódigo operação createServiceProfile

```
1: CreateServiceProfile( RichServiceOntology, RichDomainOntology, Threshold )
2: {
3:   new LinkedKnowledge = Pangea.SemanticMiddleware.LinkedKnowledge;
4:   new Annot;
5:   new result;
6:
7:   foreach ( RichServiceOntology.INPUT )
8:   {
9:     foreach( RichDomainOntology.CONCEPT )
10:    {
11:      new sim = LinkedKnowledge.getSimilarity(RichServiceOntology.INPUT,
12:                                             RichDomainOntology.CONCEPT );
13:      if ( sim > Threshold )
14:      {
15:        Annot.add( RichServiceOntology.INPUT, RichDomainOntology.CONCEPT );
16:        result.add( Annot );
17:      }
18:    }
19:  }
20:
21:  foreach ( RichServiceOntology.OUTPUT )
22:  {
23:    foreach( RichDomainOntology.CONCEPT )
24:    {
25:      new sim = LinkedKnowledge.getSimilarity(RichServiceOntology.OUTPUT,
26:                                             RichDomainOntology.CONCEPT );
27:      if ( sim > Threshold )
28:      {
29:        Annot.add(RichServiceOntology.INPUT, RichDomainOntology.CONCEPT);
30:        result.add( Annot );
31:      }
32:    }
33:  }
34:
35:  return result;
36: }
```

Figura 27. Pseudocódigo da operação *createServiceProfile* do serviço web *Process Annot*.

Considerando os requisitos apresentados na seção 4.4.1.1, o serviço web *Process Annot* satisfaz os requisitos i, ii, iv, v e vi. Após a criação do *ServiceProfile*, a próxima etapa é relacionada aos serviços WFS e consiste na anotação das saídas do serviço com base na identificação e verificação de relações espaciais topológicas e de distância.

c.2) SpatialOutputAnnot

O serviço web *Spatial Output Annot* é utilizado especialmente para descobrir as saídas pertinentes a um serviço de dados geográficos. Para isso, leva em consideração as relações espaciais encontradas na ontologia de domínio.

Para suportar o procedimento de anotação, o serviço web *Spatial Output Annot* possui a operação *Spatial Annot*, responsável pela funcionalidade de anotar os recursos com os conceitos de domínio. Essa operação recebe como entradas: a ontologia que representa as *feature types* que são alvo da anotação (i.e. *Service Ontology*), a ontologia de domínio (i.e. *Domain Ontology*) e a ontologia de relações espaciais (ver figura 28). Como saídas são esperadas as anotações semânticas, relacionando as *feature types* aos conceitos de domínio.

Para proceder a anotação semântica, o serviço web *Spatial Output Annot* busca por propriedades na ontologia de domínio que indicam algum tipo de relação espacial entre os conceitos. Uma relação espacial é descoberta comparando cada propriedade da ontologia de domínio com conceitos em uma Ontologia de Relações Espaciais – ORE (ver figura 28). Nessa ontologia estão definidas as primitivas que exprimem as principais classes de relações espaciais conhecidas: (i) topológicas; (ii) direcionais; (iii) distância. As relações espaciais na ORE foram definidas a partir das operações espaciais disponíveis no PostGIS. Para cada relação espacial identificada, o *Spatial Output Annot* recupera as instâncias da *feature type* sendo anotada e aplica as operações espaciais correspondentes à relação identificada. Caso as instâncias satisfaçam a condição espacial, então são anotadas com o conceito de domínio em análise.

Para ilustrar o funcionamento do *Spatial Output Annot*, no caso de um conceito “Manguezal” ser identificado na ontologia de domínio, este tem associada uma propriedade que define o “Manguezal” como uma “*região costeira, associada às margens de baías*”. A relação espacial “às margens” está associada à uma ou mais operações espaciais primitivas. O *Spatial Output Annot* recupera as instâncias necessárias e realiza a verificação da relação espacial. Caso a relação espacial seja verificada, a anotação semântica é sugerida. Com isso, na tabela 10 é apresentada a operação *Spatial Annot* do serviço web *Spatial Output Annot*. Na figura 29 o pseudocódigo é apresentado.

Tabela 10 - Descrição da operação *Spatial Annot* do serviço web *Spatial Output Annot*.

Serviço web Spatial Output Annot – Operação Spatial Annot	
Entradas:	<p>FeatureType_Ontology(URL) – Consiste da lista de <i>features types</i> e seus metadados.</p> <p>Domain_Ontology (URL) – Referência para a ontologia de domínio.</p> <p>Spatial_Relation_Ontology (URL) – Referência para a ontologia que especifica as relações espaciais primitivas.</p>
Saídas:	<p>annotated_data (Object) – Associação entre <i>feature types</i> e conceitos.</p>

O procedimento de anotação semântica realizado pelo serviço web *SpatialOutputAnnot* é similar ao processo de validação de anotações descrito por Klien (KLIEN, E.M. 2008). A diferença em relação ao trabalho de Klien (KLIEN, 2008) consiste no mecanismo de identificação das relações espaciais, que é feita com base na Ontologia de Relações Espaciais – ORE. Além disso, as relações espaciais são utilizadas para encontrar anotações e não apenas para validar anotações já realizadas. A taxonomia da ontologia ORE é apresentada na figura 28.

- ▼ ● GeometryRelationship
 - Distance
 - Equals
 - Disjoint
 - Intersects
 - Touches
 - Crosses
 - Within
 - Overlaps
 - Contains
 - Relate
- ▼ ● GeometryProcessing
 - Centroid
 - Area
 - Length
 - PointOnSurface
 - Boundary
 - Buffer
 - ConvexHull
 - Intersection
 - SymDifference
 - Difference
 - GeomUnion
- ▼ ● GeometryAccessors
 - AsText
 - AsBinary
 - SRID
 - Dimension
 - Envelope
 - IsEmpty
 - IsSimple
 - IsClosed
 - IsRing
 - NumGeometries
 - GeometryN

Figura 28. Estrutura geral da ontologia ORE.

Pseudocódigo operação SpatialAnnot

```
1: SpatialAnnot( FeatureType_Ontology, Domain_Ontology, Spatial_Rel_Ontology )
2: {
3:
4:   new annotated_data;
5:
6:   foreach ( Domain_Ontology.Attribute )
7:   {
8:     if ( isSpatial( Domain_Ontology.Attribute ) )
9:     {
10:      foreach ( FeatureType_Ontology.feature_type )
11:      {
12:        new flag = testSpatialOperation( Domain_Ontology.Attribute,
13:                                         FeatureType_Ontology.feature_type 14: );
15:        if (flag)
16:        {
17:          annotated_data.add(Domain_Ontology.class,
18:                             FeatureType_Ontology.feature_type );
19:        }
20:      }
21:    }
22:  }
23: }
24:
25: result.add(annotated_data);
26:
27: return result;
28: }
```

Figura 29. Pseudocódigo da operação *Spatial Annot* do serviço web *Spatial Output Annot*.

Considerando os requisitos apresentados na seção 4.4.1.1, o serviço web *Spatial Output Annot* satisfaz os requisitos i, iii, iv e v. Na sequência, é apresentado o anotador baseado em métricas de recuperação de informação: *Text Similarity Annot*.

c.3) TextSimilarityAnnot

A anotação feita com base nas relações espaciais (procedida pelo serviço *Spatial Output Annot*) pode não resultar na anotação de algumas *features types*. Além disso, a anotação semântica procedida pelo *Spatial Output Annot* resulta apenas em anotações das saídas providas pelo serviço, sendo que as entradas aceitas ficam em aberto. Em ambas as situações é necessário proceder um mecanismo de anotação que utiliza a similaridade textual para associar semântica tanto às entradas como às saídas do serviço.

Neste sentido, o serviço web *Text Similarity Annot* tem como finalidade associar conceitos às entradas e saídas que ainda não foram anotadas pelo serviço *Spatial Output Annot*. Antes de descrever o mecanismo através do qual a anotação das entradas é realizada, é importante verificar o meio de acesso às *features* disponibilizado por serviços WFS. Serviços WFS disponibilizam acesso às *features* por meio da operação *getFeature*. A operação *getFeature* permite recuperar as informações geográficas por meio da aplicação de filtros nos atributos alfanuméricos e espaciais pertinentes às *features types*. Com isso, é possível modelar um serviço WFS de acordo com entradas

e saídas, a partir dos atributos presentes no esquema de dados.

No exemplo anterior é possível verificar que um serviço WFS pode ser modelado semanticamente de diferentes maneiras a partir dos atributos presentes no esquema de dados, e portanto, é fundamental proceder a anotação semântica das entradas (i.e. dos atributos presentes no esquema) para possibilitar o descobrimento a partir de variadas consultas.

A anotação semântica das entradas é realizada por meio da associação de conceitos do domínio às entradas do esquema de dados da *feature_type* disponibilizada pelo WFS. A associação é realizada pela métrica de Levenshtein entre a descrição textual do atributo da *feature_type* e os conceitos da ontologia de domínio. Caso não haja correspondência para determinado *threshold*, é consultado o dicionário wordnet na tentativa de encontrar termos sinônimos ao atributo. Uma vez que a associação de conceitos às entradas e, eventualmente, saídas foi realizada, é criado um *Service Profile* conforme o padrão OWL-S. O *Service Profile* é a saída da operação *Text Annot*.

Para suportar a anotação semântica conforme descrita, o serviço *TextSimilarityAnnot* contém a operação *Text Annot*. A descrição do serviço web *Text Similarity Annot* é apresentada na tabela 11, enquanto na figura 30 é descrito o pseudocódigo da operação *Text Annot*.

Tabela 11 - Descrição da operação *Text Annot* do serviço web *Text Similarity Annot*.

Serviço web Text Similarity Annot – Operação Text Annot	
Entradas:	OGC WFS Service (URL) – Referência para um serviço OGC do tipo WFS. DomainOntology (URL) – Referência para a ontologia de domínio.
Saídas:	service_profile (Object) – ServiceProfile do serviço WFS.

Pseudocódigo operação Text Annot

```
1: TextAnnot( ServiceOntology, DomainOntology, Threshold )
2: {
3:   new LinkedKnowledge = SemanticMiddleware.LinkedKnowledge;
4:
5:   foreach ( ServiceOntology.INPUT )
6:   {
7:     foreach ( DomainOntology.CLASS )
8:     {
9:       new sim = LinkedKnowledge.Levenshtein(ServiceOntology.INPUT,
10:                                             DomainOntology.CLASS);
11:       if (sim > threshold)
12:       {
13:         Annot.add(ServiceOntology.INPUT, DomainOntology.CLASS)
14:       }
15:     }
16:   }
17:
18:   foreach ( ServiceOntology.OUTPUT )
19:   {
20:     foreach ( DomainOntology.CLASS )
21:     {
22:       new sim = LinkedKnowledge.Levenshtein(ServiceOntology.OUTPUT,
23:                                             DomainOntology.CLASS);
24:       if (sim > threshold)
25:       {
26:         Annot.add(ServiceOntology.OUTPUT, DomainOntology.CLASS)
27:       }
28:     }
29:   }
30:
31:   result.add(Annot);
32: }
```

Figura 30. Pseudocódigo da operação *Text Annot* do serviço web *Text Similarity Annot*.

c.4) InferenceAnnot

O serviço web *Inference Annot* é responsável por inferir anotações semânticas a partir das anotações validadas pelos usuários e, também, a partir do conhecimento agregado pelo serviço web *Knowledge Enrich* na etapa de enriquecimento semântico (item b). Por meio de métodos de dedução lógica, novos fatos podem ser derivados da ontologia de domínio.

A operação *Infer Annot* do serviço web *Inference Annot* é a responsável por realizar a inferência e descobrir novas anotações. Esta recebe como entradas: as anotações validadas pelos usuários (*Annots*) e uma referência para o catálogo (*Catalog*) onde as anotações devem ser salvas ao final do procedimento de inferência (ver campo “Entradas” na tabela 11). Como saída é esperado o conjunto de anotações inferidas (ver campo “Saídas” na tabela 11).

No contexto da anotação semântica, são aplicados dois métodos de dedução para deduzir novas anotações: (i) Dedução para classificação a partir das definições formais dos conceitos; (ii) Dedução com base em regras, especificadas conforme o padrão *Semantic Web Rule Language – SWRL*. Ambos os tipos estão apresentados a seguir:

- **Dedução de classificação** – Infere as relações de classe-subclasse a partir da descrição das classes da ontologia (i.e. sua definição formal a partir das restrições de pertinência) (ver linha 7 da figura 31) ;
- **Dedução baseado em regras SWRL** – Infere novas relações semânticas a partir do processamento de regras SWRL (ver linha 8 da figura 31).

As anotações semânticas inferidas são armazenadas em um componente catálogo (ver linha 10 da figura 31). Na tabela 12 é apresentado o método *Infer Annot* e na figura 31 o pseudocódigo.

Tabela 12 - Descrição da operação *Infer Annot* do serviço web *Inference Annot*.

Serviço web <i>InferenceAnnot</i> – Operação <i>Infer Annot</i>	
Entradas:	Annots (URL) – Referência para as anotações validadas. Catalog (URL) – Referência para o catálogo.
Saídas:	annots_infer (Object) – Anotações inferidas.

Pseudocódigo operação <i>Infer Annot</i>
<pre> 1: Infer(Annots, Catalog) 2: { 3: 4: new annots = Annots; 5: new catalog = Catalog; 6: 7: annots_infer.add(subsumption_infer(annots)); 8: annots_infer.add(swrl_infer(annots)); 9: 10: catalog.save(annots); 11: 12: result.add(annots_infer); 13: 14: return result; 15: 16: }</pre>

Figura 31. Pseudocódigo da operação *Infer* do serviço web *Inference Annot*.

As anotações originais (Annots) e as inferidas (annots_infer) são armazenadas em um catálogo (Catalog) separadamente das ontologias de domínio e dos serviços. As anotações são materializadas no catálogo como triplas utilizando relações de equivalência (owl:sameAs) entre os conceitos da ontologia de domínio e os conceitos da *Feature Type Ontology*. Considerando os requisitos apresentados na seção 4.4.1.1, o serviço web *Inference Annot* satisfaz os requisitos i, iii, iv, v e vi.

Na seção 4.4.2 é apresentado o pacote responsável por prover mecanismos de integração de ontologias.

4.4.2 Integração de ontologias

No contexto desse trabalho, o problema da integração de ontologias consiste em encontrar e materializar as relações semânticas entre as entidades de ontologias distintas. O termo integração é utilizado nesse capítulo frequentemente, portanto, é importante definir a semântica desse termo, e diferencia-lo dos termos *matching* e mapeamento. O termo integração é empregado para fazer referência ao estabelecimento de uma nova ontologia que materializa as correspondências encontradas entre duas ontologias distintas. Por sua vez, o termo *matching* é adotado para denotar o procedimento de cálculo de similaridade entre as entidades de uma ontologia. O termo mapeamento é empregado para denotar funções que fazem o mapeamento das entidades de uma ontologia em outra.

4.4.2.1 Requisitos

Os requisitos que devem ser suportados pelo mecanismo de integração de ontologias adotado na Pangea são:

- i. Identificar as relações entre classes-classes, instâncias-instâncias e classes-instâncias** – O mecanismo de integração deve procurar por relações semânticas entre classes-classes, classes-instâncias, instâncias-instâncias. As relações a serem procuradas devem ser de equivalência;
- ii. Identificar as características léxicas e linguísticas dos conceitos e instâncias** – Durante o procedimento de *matching* devem ser levados em consideração os aspectos léxicos e linguísticos dos conceitos. As propriedades *rdfs:label* e *rdfs:comments*, bem como o próprio nome do conceito podem ser utilizados para essa finalidade;
- iii. O contexto semântico dos conceitos sendo alinhados deve ser investigado** – Durante o procedimento de integração de ontologias é importante considerar o contexto semântico no qual as entidades da ontologia estão inseridas. Dado um parâmetro de profundidade de investigação, o contexto deve ser analisado e levado em conta para definir a medida de similaridade;

- iv. **A integração deve considerar conhecimento externo relacionado aos conceitos** - Esse requisito faz referência à etapa de *Anchoring* descrita em (SABOU; D'AQUIM; MOTTA, 2006). Essa etapa consiste em associar descrição semântica externa aos conceitos sendo alinhados. Na Pangea, deve ser conduzida uma etapa de *Anchoring*;
- v. **Devem ser considerados conhecimentos provenientes da iniciativa OpenLinkedData** – A etapa de *anchoring* é muito importante para o procedimento de integração conduzido na Pangea. Com isso, é importante considerar algumas das fontes de ontologias disponíveis na iniciativa *OpenLinkedData*;
- vi. **Avaliar o conhecimento associado aos termos na etapa de *anchoring*, seguindo medidas de similaridade baseadas em análise de sinônimos** – Semelhante ao trabalho (GRACIA; LOPEZ; D'AQUIN, 2007), as associações encontradas durante a etapa de *anchoring* devem ser validadas por meio da exploração do contexto semântico e de medidas de similaridade baseadas em análise de sinônimos;
- vii. **Considerar aspectos geoespaciais na integração entre conceitos** – Em alguns casos é importante considerar aspectos geoespaciais relacionados aos conceitos sendo alinhados. É possível extrair evidências geoespaciais que permitem associar semanticamente os conceitos.

Os requisitos listados devem ser suportados pelo projeto do integrador semântico da Pangea. Nas próximas seções o projeto do integrador é descrito em função dos serviços web que o compõem.

4.4.2.2 Serviços web

A integração de ontologias da Pangea é baseada em múltiplos critérios, e é composta por três serviços web: (i) *Lexical Similarity*; (ii) *Semantic Similarity*; (iii) *Spatial Similarity*.

O serviço web *Lexical Similarity* é responsável por medir as semelhanças léxicas/sintáticas entre os conceitos candidatos ao *matching*. Para tanto, são empregadas métricas baseadas em

similaridade entre palavras isoladas, textos e também análise de sinônimos.

Entretanto, é possível que conceitos semanticamente relacionados possam ter diferenças léxicas bastante evidentes (como é o caso, por exemplo, dos conceitos *Food* e *Meat*). Nesse caso é relevante considerar outras características para estabelecer a relação semântica entre os conceitos. Com isso, é proposto também o módulo *Semantic Similarity*. Esse módulo faz o cálculo de similaridade com base em conhecimento extraído a partir da Web Semântica, considerando repositórios pré-cadastradas na Pangea.

Além dos serviços web já apresentados, há também a possibilidade de relacionar conceitos por meio dos aspectos espaciais relacionados aos mesmos. O serviço web *Spatial Similarity* utiliza as características espaciais das instâncias para medir a similaridade entre dois conceitos. Nas próximas seções, são apresentados os serviços web descritos anteriormente em linhas gerais.

a) Lexical Similarity

O serviço web *Lexical Similarity* provê operações que permitem mensurar a similaridade entre duas entidades por meio da análise das cadeias de textos relacionadas às mesmas. Nesse contexto, há três abordagens utilizadas para a análise de similaridade entre duas entidades: (i) Baseada na similaridade entre palavras isoladas; (ii) Baseada na similaridade entre conjuntos de palavras; (iii) Baseados em métricas de sinônimos.

Na literatura há diversas métricas de similaridade entre palavras isoladas (abordagem (i)), porém, o estudo conduzido por Stoilos, Stamou e Kollias (2005) demonstra que elas não são adequadas ao contexto da integração de ontologias. Neste sentido, Stoilos, Stamou e Kollias propõem uma nova métrica, apresentada na Equação (1), que demonstra ter melhores resultados quando comparada às métricas tradicionais. Para estabelecer um grau de similaridade com base em palavras isoladas, o serviço web *Lexical Similarity* emprega a métrica de similaridade descrita em (STOILLOS; STAMOU; KOLLIAS, 2005).

Eq.(1)

$$Sim(s_1, s_2) = Comm(s_1, s_2) - Diff(s_1, s_2) + winkler(s_1, s_2)$$

Onde: s1 e s2 são as palavras.

Na métrica de similaridade, a função $Comm(s_1, s_2)$, apresentada na Equação (2), implementa a métrica baseada no casamento de subtópicos. Essa métrica faz o cálculo do maior

subtexto contido em ambas as palavras s_1 e s_2 . Uma vez que esse subtexto é encontrado, ele é removido das palavras e o procedimento é repetido até que não haja subtextos em comum. O tamanho desses subtextos em comum é então utilizado para chegar ao resultado da função $Comm$.

$$Comm(s_1, s_2) = \frac{2 * \sum_i length(maxComSubString_i)}{length(s_1) + length(s_2)} \quad \text{Eq.(2)}$$

A função de diferença (i.e. $Diff$), apresentada na Equação (3), é baseada no tamanho das *substrings* que não foram casadas. De acordo com Stoilos (STOILLOS 2005), o parâmetro p pode ser ajustado, sendo o valor 0.6 observado como ideal em experimentos. A função $Winkler(s_1, s_2)$, é baseada na métrica descrita por Winkler (1999).

$$Diff(s_1, s_2) = \frac{uLen_{s_1} * uLen_{s_2}}{p + (1 - p) * (uLen_{s_1} + uLen_{s_2} - uLen_{s_1} * uLen_{s_2})} \quad \text{Eq.(3)}$$

Na abordagem (ii) utilizada para o cálculo da similaridade entre textos, é empregada a técnica TF-IDF (GARCIA, 2006), de forma similar ao descrito em Cruz, Antonielli e Stroe (2009). Tais autores (CRUZ; ANTONIELLI; STROE 2009) observam que comumente os métodos de alinhamentos procuram por associações entre entidades isoladas, classes são comparadas com classes, instâncias com instâncias, o que acaba por desconsiderar o contexto no qual os conceitos estão inseridos, suas propriedades, *labels*, descrições, etc. Com isso, na abordagem baseada em TF-IDF, os conceitos e suas informações correspondentes são transformadas em vetores TF-IDF e a distância entre cossenos é empregada para calcular a similaridade.

Com relação à abordagem (iii) é utilizada uma métrica baseada em sinônimos, que faz uso de consultas ao WordNet, de forma similar, porém menos abrangente, ao descrito por Gracia, Lopez e D'Aquin (2007). O WordNet é consultado para verificar os sinônimos relacionados aos termos e então é empregada a métrica adotada na abordagem (i) para analisar a similaridade entre os sinônimos.

Além das métricas utilizadas no cálculo da similaridade, é importante considerar o formato no qual são expressas as ontologias que são alvo do processo de integração. As ontologias são

especificadas em OWL, portanto, há algumas propriedades relacionadas aos conceitos que podem ser utilizadas no processo de cálculo da similaridade: *rdfs:comment*, *rdfs:label*, *rdfs:isDefinedBy* e *rdfs:seeAlso*. O conteúdo dessas propriedades, bem como a denominação do próprio conceito, são utilizados para calcular a similaridade.

Para suportar as funcionalidades mencionadas, o serviço web *Lexical Similarity* contém a operação *Align Ontology*. Essa operação recebe como entradas duas ontologias e retorna a similaridade calculada entre os seus elementos. São realizadas comparações entre classes-classes, instâncias-instâncias, classes-instâncias.

Antes de iniciar a análise de similaridade propriamente dita, a primeira etapa consiste em normalizar os termos. A normalização consiste em extrair espaços em branco e símbolos especiais (_ , * , (), entre outros). Somente após a normalização são conduzidas as análises de similaridade evidenciadas anteriormente.

A primeira análise é com base na métrica proposta por Stoilos (2005), e envolve a análise dos rótulos e também dos nomes propriamente ditos dos conceitos e instâncias. Após, é conduzida uma análise utilizando a técnica TF-IFD. Para tanto, são recuperados além dos nomes e rótulos dos conceitos, os comentários (i.e. *rdfs:comment*), definições (i.e. *rdfs:isDefinedBy*), as referências externas (i.e. *rdfs:seeAlso*), propriedades, e também o contexto semântico (i.e. superclasse direta e subclasses). Os vetores com as palavras são montados com base nesses fragmentos de textos e então é avaliada a distância de cosenos. A etapa final de cálculo de similaridade emprega medida de sinônimos para determinar a similaridade entre as entidades. Após realizados os três cálculos de similaridade, os mesmos são combinados, considerando o peso definido pelo usuário/agente quando da invocação do serviço web. Como saída, tem-se o índice de similaridade léxico. A operação *Align Ontology* é descrita em detalhes na tabela 13. O pseudocódigo da operação é apresentado na figura 32.

Tabela 13 - Descrição geral da operação *Align Ontology*.

Serviço web Lexical Similarity – Operação Align Ontology	
Entradas:	<p>OntologyA (URL) – Referência para uma das ontologias a ser alinhada.</p> <p>OntologyB (URL) – Referência para uma das ontologias a ser alinhada.</p> <p>Weight1 (Float) – Peso para abordagem 1.</p> <p>Weight2 (Float) – Peso para abordagem 2.</p> <p>Weight3 (Float) – Peso para abordagem 3.</p>
Saídas:	<p>sim_idx (Object) – Índices de similaridade.</p>

Pseudocódigo operação Align Ontology

```
1: AlignOntology( OntologyA, OntologyB, weight1, weight2, weight3 )
2: {
3:   new stoilos = SemanticMiddleware.LinkedKnowledge.TextSimilarity.Stoilos;
4:   new tfidf = SemanticMiddleware.LinkedKnowledge.TextSimilarity.TFIDF;
5:   new sinon = SemanticMiddleware.LinkedKnowledge.TextSimilarity.Sinon;
6:
7:   new ClassPackage;
8:   new InstancePackage;
9:   new InstanceClassPackage;
10:
11:   new return;
12:
13:   foreach ( OntologyA.CLASS )
14:   {
15:     foreach ( OntologyB.CLASS )
16:     {
17:       var simLex = stoilos.getSim( OntologyA.CLASS, OntologyB.CLASS );
18:       var tfidf = tfidf.getSim( OntologyA.CLASS, OntologyB.CLASS );
19:       var sinon = sinon.getSim( OntologyA.CLASS, OntologyB.CLASS );
20:
21:       ClassPackage.add(OntologyA.CLASS, OntologyB.CLASS, simLex, tfidf, sinon);
22:     }
23:   }
24:
25:   foreach ( OntologyA.INSTANCE )
26:   {
27:     foreach ( OntologyB.INSTANCE )
28:     {
29:       var simLex = stoilos.getSim( OntologyA.INSTANCE, OntologyB.INSTANCE );
30:       var tfidf = tfidf.getSim( OntologyA.INSTANCE , OntologyB.INSTANCE );
31:       var sinon = sinon.getSim( OntologyA.INSTANCE , OntologyB.INSTANCE );
32:
33:       InstancePackage.add(OntologyA.INSTANCE, OntologyB.INSTANCE,
34:                           simLex, tfidf, sinon);
35:     }
36:   }
37:
38:   foreach ( OntologyA.CLASS )
39:   {
40:     foreach ( OntologyB.INSTANCE )
41:     {
42:       var simLex = stoilos.getSim( OntologyA.CLASS, OntologyB.INSTANCE );
43:       var tfidf = tfidf.getSim( OntologyA.CLASS , OntologyB.INSTANCE );
44:       var sinon = sinon.getSim( OntologyA.CLASS , OntologyB.INSTANCE );
45:
46:       InstanceClassPackage.add( OntologyA.CLASS, OntologyB.INSTANCE,
47:                               simLex, tfidf, sinon);
48:     }
49:   }
50:
51:   foreach ( OntologyB.CLASS )
52:   {
53:     foreach ( OntologyA.INSTANCE )
54:     {
55:       var simLex = stoilos.getSim( OntologyB.CLASS, OntologyA.INSTANCE );
56:       var tfidf = tfidf.getSim( OntologyB.CLASS , OntologyA.INSTANCE );
57:       var sinon = sinon.getSim( OntologyB.CLASS , OntologyA.INSTANCE );
58:
59:       InstanceClassPackage.add(OntologyB.CLASS, OntologyA.INSTANCE,
60:                               simLex, tfidf, sinon);
61:     }
62:   }
63:
64:   result.add( ClassPackage, InstancePackage, InstanceClassPackage );
65:   return result;
66: }
```

Figura 32. Pseudocódigo da operação *Align Ontology*.

O método suportado pelo *Lexical Similarity* é ideal para ontologias que possuem semelhanças léxicas, entretanto, falha em situações onde há heterogeneidades nesse aspecto. Deste

modo, o serviço web *Semantic Similarity* tem por objetivo suplantar as limitações que ocorrem nos casos onde as diferenças léxicas são predominantes. O serviço web *Semantic Similarity* é apresentado na próxima seção.

b) Semantic Similarity

O serviço web *Semantic Similarity* provê operações para gerar mapeamentos a partir do enriquecimento semântico dos conceitos (etapa denominada *Anchoring*) e de dedução lógica. O método é similar ao empregado em (SABOU; D'AQUIN; MOTTA, 2006), considerando a estratégia S2.

O processo de *matching* dos conceitos é dividido em duas etapas: (i) *Anchoring* semântico; (ii) Dedução. Na etapa (i), é relacionado conhecimento externo aos conceitos sendo alinhados. Após, na etapa (ii), o conhecimento externo adicionado aos conceitos é utilizado para inferir a relação semântica entre os conceitos.

Para suportar esse processo, o serviço web *Semantic Similarity* contém a operação *Semantic Align Ontology*. Essa operação recebe como entrada as duas ontologias candidatas ao alinhamento e regras especificadas em *Semantic Web Rule Language – SWRL*, utilizadas para inferir os alinhamentos. Como saída é gerada a relação semântica entre os conceitos das ontologias fornecidas como entrada.

Os termos candidatos ao *matching* são normalizados (eliminados caracteres especiais como *, (), _, entre outros) antes de iniciar o procedimento de cálculo da similaridade. Na primeira etapa, são recuperadas as ontologias que fazem referência aos termos normalizados. A busca pelas ontologias de referência é realizada em duas classes de repositórios: (i) Repositórios Internos; (ii) Repositórios externos. Os repositórios externos são representados pelo Swoogle (SWOOGLE, 2010) e por repositórios da iniciativa *OpenLinkedData*. Por outro lado, os repositórios internos são representados pelas ontologias armazenadas no *Semantic Server*.

A segunda etapa consiste em aplicar as regras fornecidas como entrada sobre as ontologias recuperadas dos repositórios internos e externos, de modo a inferir o relacionamento semântico entre os termos. Os relacionamentos podem ser de equivalência, generalização (especialização), disjunção. Como saída dessa operação, se tem as relações semânticas entre os termos candidatos ao *matching*.

A operação *Semantic Align Ontology* é descrita na tabela 14. O pseudocódigo da operação é apresentado na figura 33.

Tabela 14 - Descrição geral da operação *Semantic Align Ontology*.

Serviço web Semantic Similarity – Operação Semantic Align Ontology	
Entradas:	<p>OntologyA (URL) – Referência para uma das ontologias a ser alinhada.</p> <p>OntologyB (URL) – Referência para uma das ontologias a ser alinhada.</p> <p>Rules (URL) – Regras utilizadas para inferir os alinhamentos.</p>
Saídas:	sim_idx (Object) – Índices de similaridade.

Pseudocódigo operação Semantic Align Ontology
<pre> 1: SemanticAlignOntology(OntologyA, OntologyB, rules) 2: { 3: new SWOOGLE; 4: 5: foreach (OntologyA.CLASS) 6: { 7: anchorPkgClassA = SWOOGLE.anchoring(OntologyA.CLASS); 8: } 9: 10: foreach (OntologyB.CLASS) 11: { 12: anchorPkgClassB = SWOOGLE.anchoring(OntologyB.CLASS); 13: } 14: 15: foreach (OntologyA.CLASS) 16: { 17: foreach (OntologyB.CLASS) 18: { 19: new semRelations = applyRules(anchorPkgClassA, anchorPkgClassB, rules, 20: OntologyA.CLASS, OntologyB.CLASS); 21: result.add(semRelations); 22: } 23: } 24: 25: return result; 26: }</pre>

Figura 33. Pseudocódigo da operação *Semantic Align Ontology* do serviço web *Semantic Similarity*.

A operação de alinhamento com base em semântica permite estabelecer relacionamentos semânticos mesmo onde ocorrem diferenças léxicas entre as ontologias. O terceiro método de alinhamento descrito nesse trabalho considera a relação espacial entre os conceitos para estabelecer uma ligação semântica entre os termos. O serviço web *Spatial Similarity* é descrito no próximo item.

c) Spatial Similarity

O *Spatial Similarity* é o serviço web que provê operação para o *match* de ontologias considerando aspectos espaciais relacionados aos conceitos e às instâncias. A abordagem adotada é

similar a descrita por Hess (2008) para medir a similaridade de conceitos geográficos.

Para suportar o *matching* com base em aspectos espaciais, o serviço web *Spatial Similarity* contém a operação *Spatial Align Ontology*. Essa operação recebe como entrada as ontologias candidatas ao alinhamento e retorna a medida de similaridade espacial entre os termos.

A primeira etapa consiste em verificar se os conceitos candidatos ao *match* espacial possuem instâncias. No caso de não haver instâncias para os conceitos sendo analisados, os repositórios do *Semantic Server* (apresentado na seção 4.5) são consultados para recuperar instâncias. A partir das instâncias é realizado o cálculo da *Geographic Context Region Similarity* (HESS, 2008). Essa região é calculada com base na sobreposição entre os *BoundingBoxs* geográfico que contemplam as instâncias de ambos os conceitos. Caso a sobreposição seja vazia, então os conceitos são relacionados como disjuntos.

No caso de haver a sobreposição, então é realizado o cálculo da métrica espacial nas instâncias que participam da sobreposição geográfica. Para o cálculo da métrica, a primeira etapa consiste em transformar a geometria das instâncias para pontos (para geometrias do tipo polígono o centróide é adotado. Para geometrias do tipo linha o centróide do *BoundingBox* é adotado). Então, a medida de similaridade é calculada com base na distância euclidiana inversa entre as coordenadas, de forma análoga ao descrito por Hess (2008). Como saída, é obtida a similaridade espacial. A operação *Spatial Align Ontology* é descrita em detalhes na tabela 15. O pseudocódigo da operação é apresentado na figura 34.

Tabela 15 - Descrição geral da operação *Spatial Align Ontology*.

Serviço web Spatial Similarity – Operação Spatial Align Ontology	
Entradas:	OntologyA (URL) – Referência para uma das ontologias a ser alinhada. OntologyB (URL) – Referência para uma das ontologias a ser alinhada.
Saídas:	sim_package (Object) – Conceitos relacionados.

Pseudocódigo operação Spatial Align Ontology

```
1: SpatialAlignOntology ( OntologyA, OntologyB )
2: {
3:   foreach ( OntologyA.CLASS )
4:   {
5:     foreach ( OntologyB.CLASS )
6:     {
7:       instancesA = SemanticServer.getInstances( OntologyA.CLASS )
8:       instancesB = SemanticServer.getInstances( OntologyB.CLASS )
9:
10:      if ( instancesA == null )
11:      {
12:        instancesA = SemanticServer.getInstances( OntologyA.CLASS )
13:      }
14:
15:      if ( instancesB == null )
16:      {
17:        instancesB = SemanticServer.getInstances( OntologyB.CLASS )
18:      }
19:
20:      if ( ( instancesA && instancesB ) != null )
21:      {
22:        bboxA = instancesA.getBoundingBox();
23:        bboxB = instancesB.getBoundingBox();
24:
25:        intersection = Intersects( bboxA, bboxB );
26:
27:        if ( intersections != null )
28:        {
29:          instA = instancesA.getInstances( intersection );
30:          instB = instancesB.getInstances( intersection );
31:
32:          foreach( instA.inst )
33:          {
34:            foreach( instB.inst )
35:            {
36:              distArr = inverseEuclidesDist( instA, instB );
37:              semanticSimilarity.add( instA.inst, instB.inst, distArr );
38:            }
39:          }
40:
41:        }
42:      }
43:
44:    }
45:  }
46:
47:  result.add( semanticSimilarity );
48: }
```

Figura 34. Pseudocódigo da operação *Spatial Align Ontology* do serviço web *Spatial Align Ontology*.

4.5 Semantic Server

Ontologias são artefatos centrais ao funcionamento da arquitetura Pangea. Nesse contexto, um dos requisitos que deve ser suportado pela Pangea consiste em prover ontologias aos componentes da arquitetura para que executem as tarefas de anotação, descobrimento, entre outras.

Assim, é importante considerar os seguintes aspectos para o projeto de um módulo que

suporte os requisitos mencionados:

- **As funcionalidades expostas aos usuários/agentes** – Consiste em definir as funcionalidades disponibilizadas aos usuários/agentes. No contexto da Pangea é importante considerar um módulo que permita armazenar, recuperar e realizar dedução lógica em ontologias. Na Pangea, tais funcionalidades são fundamentais, uma vez que as ontologias são empregadas durante vários processos chave para a arquitetura, entre eles: a anotação semântica e o descobrimento de recursos geográficos;
- **Organização arquitetural** – Consiste em definir as camadas que definem a arquitetura do módulo. No contexto apresentado, são relevantes quatro camadas: (i) Camada de persistência; (ii) Camada de API; (iii) Camada de serviços web; (iv) Camada de aplicação. Na Camada de Persistência devem ser disponibilizados os mecanismos de persistência onde as ontologias podem ser armazenadas. Para permitir independência do mecanismo de persistência adotado, a camada de API foi projetada. Essa camada permite estabelecer interfaces padrão para armazenar, atualizar e consultar as ontologias na camada de repositórios. A camada de serviços web disponibiliza acesso às funcionalidades da API para as aplicações na camada de aplicação.

Outro aspecto importante a ser considerado no projeto do módulo é a divisão conceitual na camada de repositórios: repositórios internos e externos. Os repositórios internos consistem em ontologias mantidas por usuários/agentes que são cadastrados na Pangea. Os repositórios externos consistem em serviços mantidos por terceiros que permitem acesso à ontologias na Web (ex. Swoogle).

Considerando o exposto, o *Semantic Server* é o módulo responsável por catalogar as ontologias utilizadas na arquitetura e prover serviços web que permitam localizar, recuperar e fazer dedução lógica nas ontologias. O *Semantic Server* é composto pelo pacote *Semantic Server Utils* que reúne os serviços web que suportam os requisitos apresentados.

O *Semantic Server* é o módulo responsável por prover facilidades para armazenar, organizar e recuperar ontologias na Web. Ontologias são relevantes em diversos contextos, considerando sistemas de informação. Na Pangea, as ontologias são utilizadas, principalmente, para organizar semanticamente os repositórios de dados geográficos. Em todos os contextos onde ontologias são empregadas, é importante garantir meios de armazenar, organizar, recuperar, e realizar dedução lógica utilizando as ontologias. O *Semantic Server* tem o objetivo de disponibilizar

essas funcionalidades às aplicações.

O armazenamento de ontologias pode ser realizado em diversos meios, dependendo das necessidades das aplicações que consomem as ontologias. Desse modo, o *Semantic Server* permite o armazenamento em bancos de dados relacionais, arquivos e na memória.

Com relação à organização das ontologias, o *Semantic Server* permite organizar as ontologias com base em dois aspectos: (i) Generalidade da ontologia; (ii) Utilidade da ontologia. O aspecto (i) permite classificar as ontologias como sendo de topo, domínio ou aplicação. O aspecto (ii) permite classificar as ontologias como sendo *profile*, *trilhas* e modelos geoespaciais.

Os mecanismos de acesso e recuperação de ontologias são disponibilizados às aplicações como serviços web. Os serviços web são baseados em uma camada de API que provê funcionalidades para acesso e manipulação das ontologias.

Considerando os métodos de dedução, o *Semantic Server* disponibiliza às aplicações serviços web para realizar dedução lógica sobre ontologias.

Os aspectos de armazenamento, organização, recuperação e dedução lógica (apresentados em linhas gerais) são agrupados no pacote *Semantic Server Utils*. Esse pacote reúne os serviços web responsáveis por prover as funcionalidades mencionadas.

Na seção 4.5.1 são apresentados os detalhes de projeto do *Semantic Server*. Inicialmente são descritos os requisitos que o *Semantic Server* deve suportar. Após, com base nos requisitos, são apresentados a arquitetura geral do *Semantic Server* e os serviços web que o compõem.

4.5.1 Semantic Server Utils

Aplicações semanticamente habilitadas são aquelas que utilizam ontologias em suas operações. Na Pangea as ontologias são utilizadas principalmente para anotar os repositórios de dados e modelos geográficos.

Nas aplicações semânticas, é comum que haja a necessidade de localizar e consumir ontologias sob demanda. Desse modo, é necessário um componente capaz de suportar as aplicações semânticas no descobrimento, manutenção e dedução lógica sobre ontologias.

Portanto, o cenário onde é inserido o *Semantic Server* é caracterizado por aplicações semânticas e ontologias diversas. O problema principal é estabelecer meios para que as aplicações semânticas possam localizar, consumir e manter as descrições semânticas, utilizadas em suas operações.

Uma vez exposto o cenário do problema, na seção 4.5.1.1 são listados os requisitos que devem ser suportados pelo *Semantic Server*. Além disso, é descrita a arquitetura geral do *Semantic Server*. Com base nos requisitos e na arquitetura geral, na seção 4.5.1.1 são apresentados os serviços web que compõem o módulo *Semantic Server*.

4.5.1.1 Requisitos

Na Pangea há diversas ocasiões onde são requeridas ontologias. Desse modo, é importante considerar os requisitos que devem ser suportados por um componente responsável por prover interfaces às aplicações para localizar, consumir, manter e realizar dedução lógica em ontologias.

Assim, considerando os trabalhos relacionados apresentados na seção 3.3.1, sobretudo as questões discutidas por Ahmad et al. (AHMAD; COLOMB, 2007), os requisitos gerais que devem ser suportados pelo *Semantic Server* são:

- i. **Disponibilizar interfaces de consulta para o descobrimento de ontologias** – A principal funcionalidade do *Semantic Server* é prover interfaces de consulta que permitam o acesso das aplicações às ontologias armazenadas;
- ii. **Permitir dedução lógica em ontologias** – O *Semantic Server* deve permitir realizar dedução lógica com base nos axiomas das ontologias armazenadas;
- iii. **Disponibilizar interfaces para publicar e manter ontologias** – O *Semantic Server* deve permitir que as aplicações possam publicar e manter ontologias;
- iv. **Suportar as principais linguagens e padrões da Web Semântica** – Deve ser possível armazenar descrições semânticas conforme os padrões RDF, RDFS, OWL. Além disso, deve suportar o padrão para a especificação de regras *Semantic Web Rule Language* – *SWRL*;
- v. **Suportar a linguagem de consulta SPARQL** – O *Semantic Server* deve processar consultas especificadas segundo o padrão SPARQL;

- vi. **Recuperar ontologias por contexto** – Deve ser possível agrupar e recuperar ontologias por contexto. Os contextos permitem diferenciar e agrupar as ontologias, possibilitando estabelecer espaços de busca menores e, desse modo, permitindo respostas mais eficientes;
- vii. **Descobrir ontologias por palavras-chave** – O *Semantic Server* deve permitir descobrir ontologias por meio de palavras-chave;

Os requisitos listados devem ser suportados pelo projeto do *Semantic Server*. Na seção 4.5.1.2 o projeto do *Semantic Server* é descrito em função das camadas e serviços web que o compõem.

4.5.1.2 Arquitetura geral

Visando atender os requisitos apresentados, o *Semantic Server* é composto de 4 camadas principais: (i) Repositórios; (ii) API; (iii) Serviços Web; (iv) Aplicações. Essas camadas são apresentadas na figura 35 e descritas em mais detalhes nos itens (a), (b), (c) e (d).



Figura 35. Arquitetura geral do Semantic Server.

a) Camada de repositórios

A camada de repositórios é a responsável por prover os mecanismos de armazenamento às ontologias. No *Semantic Server*, é possível armazenar ontologias em arquivos, na memória e em bancos de dados relacionais. Os repositórios de ontologias são divididos conceitualmente em:

- i. **Repositórios Internos** – Consistem nos repositórios de ontologias mantidos por usuários e agentes devidamente registrados no *Semantic Server*. As ontologias armazenadas nos repositórios internos são organizadas com base em sua generalidade (i.e. Topo, Domínio, Aplicação) e função no contexto da Pangea (i.e. Modelos geoespaciais, trilhas e *profiles*). As ontologias disponíveis nos repositórios internos possuem permissão de leitura para todos os usuários e de escrita para aqueles que possuem tal permissão ;
- ii. **Repositórios Externos** – Consistem nos repositórios de ontologias previamente disponíveis na Web e que não são publicados no *Semantic Server* por um usuário/agente devidamente registrado. Os repositórios externos podem apontar para qualquer *SPARQL EndPoint* distribuído na Web. De fato, há uma quantidade bastante grande de repositórios externos (somente o projeto *OpenLinkedData* cataloga mais de 80 repositórios). Como exemplo de repositórios externos é possível citar o Swoogle e o Geonames. As ontologias disponíveis nos repositórios externos permitem apenas o acesso de leitura. Caso as ontologias dos repositórios externos sejam recuperadas e publicadas como um repositório interno, então as permissões de escrita ficam disponíveis para os usuários que as possuem.

Uma vez que as ontologias são armazenadas na camada de repositórios, é importante prover mecanismos para recuperar e manipular as ontologias. Para tanto, há uma interface de programação que permite acessar funcionalidades de armazenamento, recuperação e dedução lógica sobre as ontologias armazenadas nos repositórios. Essa interface de programação (*API - Application Programming Interface*) é apresentada no item (b).

b) Camada de API

A camada de API consiste em padronizações de bibliotecas e rotinas para a manipulação das ontologias disponíveis na camada de armazenamento. As bibliotecas disponíveis nessa camada permitem acesso às funcionalidades relacionadas com o armazenamento, recuperação e dedução

lógica em ontologias. As funcionalidades mencionadas são agrupadas nos seguintes pacotes:

- i. **Armazenamento** – Esse pacote permite aos usuários/agentes previamente registrados no *Semantic Server* armazenar e atualizar ontologias nos repositórios internos. As classes desse pacote abstraem os detalhes dos diversos mecanismos de armazenamento (i.e. Arquivos, Bancos de dados relacionais, memória) suportados no *Semantic Server*;
- ii. **Recuperação** – O pacote de recuperação consiste em funcionalidades que permitem aos usuários/agentes recuperar ontologias. A recuperação de ontologias se dá em três formas: por meio de consultas estruturadas em SPARQL; Através de palavras-chave; Por meio do contexto no qual a ontologia está inserida (Cada ontologia inserida em um repositório interno é associada a um contexto, que identifica e discrimina as triplas dessa ontologia);
- iii. **Dedução lógica** - As funcionalidades disponíveis nesse pacote possibilitam que os usuários/ aplicativos executem dedução lógica com base nos axiomas da ontologia.

Com a adoção da camada de API os detalhes técnicos e específicos de plataforma dos mecanismos de armazenamento são suprimidos. Além disso, as interfaces de consulta, recuperação e dedução lógica são padronizadas. Apesar de tornar mais simplificado o acesso e manipulação das ontologias, a API não pode ser acessada externamente por usuários e agentes. Desse modo, é necessário criar um mecanismo que possibilite acessar as funcionalidades da API por meio da Web. O mecanismo adotado no *Semantic Server* consiste em publicar interfaces de serviços web que permitem acessar as funcionalidades da API. A camada de serviços web é apresentada em detalhes no item (c).

c) Camada de serviços web

A camada de serviços web possibilita que algumas das funcionalidades da API sejam disponibilizadas como serviços web. Essa camada permite que usuários e agentes façam uso das funcionalidades da API por meio da invocação de serviços web.

Há três serviços web disponíveis nessa camada:

- i. **Publisher** – O *Publisher* disponibiliza interfaces para acesso às bibliotecas relacionadas com o armazenamento e atualização das ontologias. Esse serviço web permite acessar as

funcionalidades do pacote *Armazenamento* na camada de API;

- ii. **Oracle** – O *Oracle* é o responsável por disponibilizar as interfaces de acesso às funcionalidades de recuperação de ontologias. Esse serviço web permite acessar as funcionalidades do pacote *Recuperação* na camada de API;
- iii. **Córtex** – O *Cortex* é o responsável por disponibilizar as interfaces de acesso às funcionalidades de dedução lógica. Considerando a camada de API, esse serviço web permite acessar as funcionalidades do pacote Dedução lógica.

Os serviços web mencionados compõem as interfaces a partir das quais as aplicações podem interagir com o *Semantic Server*. No item (d), é apresentada a camada de aplicações.

d) Camada de aplicações

A camada de aplicações é composta dos aplicativos que utilizam ontologias em suas tarefas. As aplicações pode ser autônomas (i.e. Agentes) ou então guiadas por um usuário. No contexto da Pangea é importante considerar se as aplicações (i.e. Agentes ou guiadas por usuários) são registradas no *Semantic Server* ou se são anônimas. Esse quesito determina quais serão as funcionalidades que as aplicações poderão utilizar no *Semantic Server*.

Aplicações registradas podem publicar e alterar ontologias. O esquema de permissões de acesso para tanto é baseado naquele adotado nos sistemas Unix, ou seja, cada recurso (no caso específico: ontologia) é associado a um dono e grupo. Para cada recurso são associados três grupos de permissões: (i) Associadas ao dono do recurso; (ii) Associadas ao grupo; (iii) Associadas aos outros usuários (i.e. Aqueles que não são dono e que também não fazem parte do grupo). Para cada grupo de permissões, são associadas regras que definem as permissões para leitura, escrita e dedução lógica.

Aplicações anônimas não podem publicar ontologias no *Semantic Server* e, no que se refere às ontologias já publicadas, devem respeitar as regras de permissões associadas aos usuários que não são donos do recurso e que não fazem parte do grupo associado ao recurso.

Uma vez que o *Semantic Server* foi apresentado de forma geral, em função de suas camadas, na seção 4.5.1.3 é apresentada em detalhes a camada de serviços web.

4.5.1.3 Serviços web

O *Semantic Server* consiste de um conjunto de serviços web que suportam os requisitos de publicar, localizar, consumir e realizar dedução lógica sobre ontologias na Web. Os serviços web são descritos nos itens (a), (b) e (c).

a) Publisher

O serviço web *Publisher* é o responsável por permitir que os usuários/agentes publiquem e/ou atualizem ontologias presentes na camada de repositórios do *Semantic Server*. Para executar essa responsabilidade o serviço web *Publisher* disponibiliza a operação *Publish*. Tal operação recebe como entradas o contexto, a ontologia e o tipo de ontologia.

A ontologia deve ser representada em notação XML e ser especificada conforme os padrões RDF, RDFS ou OWL. Além disso, são aceitas regras especificadas no padrão *Semantic Web Rule Language – SWRL*.

O “Contexto” é um meio de agrupar logicamente as ontologias. A principal vantagem da utilização de “Contextos” é relacionada à execução de consultas para recuperar as ontologias. A adoção dos “Contextos” pode auxiliar a diminuir o espaço de busca (uma vez que o contexto tenha sido pré-definido), permitindo recuperar ontologias de forma mais eficiente.

Na tabela 16 é apresentada a estrutura geral do serviço web e na figura 36 é descrito o pseudocódigo do método *Publish*, responsável por publicar ou atualizar uma ontologia.

Tabela 16 - Descrição da operação *Publish* do serviço *Publisher*.

Serviço Publisher – Operação Publish	
Entradas:	Context (String) – Contexto no qual as descrições semânticas devem ser inseridas. Semantic_Description (URI) – Descrição semântica a ser inserida. Type (String) – Indica o tipo de descrição semântica sendo inserida, ou seja, descrição de topo, domínio, aplicação, modelo geoespacial, trilha e profile.
Saídas:	status (Integer) – Status da operação realizada.

Pseudocódigo operação Publish	
<pre> 1: Publish(Context, Semantic_Description, Type) 2: { 3: int status = -1; 4: 5: status = SemanticServer.add(Context, Semantic_Description, 6: Type); 7: 8: result.add(status); 9: 10: return result; 11:}</pre>	

Figura 36. Pseudocódigo da operação *Publish*.

Considerando os requisitos apresentados na seção 4.5.1.1, o serviço web *Publish* satisfaz os requisitos iii e iv.

b) Oracle

O serviço web *Oracle* é o responsável por processar as consultas dos usuários/agentes e retornar as ontologias que satisfazem os critérios da consulta.

No *Oracle* há duas operações que permitem recuperar ontologias: *GetKnowledgeByContext* e *GetKnowledge*.

A operação *GetKnowledgeByContext* recebe como entrada um “Contexto” e, a partir do “Contexto” fornecido, retorna as ontologias pertinentes ao contexto. A operação *GetKnowledge* recupera ontologias a partir de uma consulta especificada em SPARQL. Ambas as operações suportam as aplicações quando estas estão engajadas no processo de descobrir e recuperar ontologias. A seguir são apresentadas as operações *GetKnowledgeByContext* e *GetKnowledge*.

Tabela 17 - Descrição da operação *GetKnowledgeByContext*.

Serviço Oracle – Operação <i>GetKnowledgeByContext</i>	
Entradas:	Context (String) – Evidencia o contexto a partir do qual serão recuperadas as descrições semânticas.
Saídas:	Semantic_Description (Object) – Descrições semânticas pertinentes ao contexto.

Pseudocódigo operação GetKnowledgeByContext

```

1: GetKnowledgeByContext( Context )
2: {
3:   if ( SemanticServer.existContext( Context ) )
4:   {
5:     new semantic_description =
6:       SemanticServer.getSemanticByContext( Context )
7:
8:     result.add( semantic_description );
9:   }
10:  else
11:  {
12:    result.add("status", -1)
13:    result.add("message", "Context don't exist");
14:  }
15:  return result;
16: }
17: }
```

Figura 37. Pseudocódigo da operação GetKnowledgeByContext.

Tabela 18 - Descrição da operação *GetKnowledge*.

Serviço Oracle – Operação GetKnowledge	
Entradas:	<p>Pattern (SPARQL/Keyword) – Padrão utilizado para recuperar conhecimento.</p> <p>Context (String) – Contexto utilizado na recuperação.</p>
Saídas:	<p>Semantic_Description (Object) – Descrições semânticas pertinentes ao padrão.</p>

Pseudocódigo operação GetKnowledge

```

1: GetKnowledge( Pattern, Context )
2: {
3:   new semantic = SemanticServer.getSemantic( Pattern, Context );
4:
5:   if ( semantic )
6:   {
7:     result.add( semantic );
8:   }
9:   else
10:  {
11:    result.add("status", -1);
12:    result.add("message", "Knowledge not found");
13:  }
14:  return result;
15: }
16: }
```

Figura 38. Pseudocódigo da operação *GetKnowledge*.

Considerando os requisitos apresentados na seção 4.5.1.1, o serviço web *Oracle* satisfaz os requisitos iii, iv, v, vi e vii.

c) Cortex

O serviço web *Cortex* é o responsável por realizar dedução lógica em ontologias publicadas no *Semantic Server*. O procedimento de dedução permite descobrir relações semânticas que não foram expressas de maneira explícita na ontologia.

Para suportar o requisito de dedução, o serviço web Cortex possui a operação *Reasoner*. Há três tipos de dedução lógica suportados pelo Cortex:

- i. **Checagem de taxonomia** – Checa a taxonomia de classes e infere novas classificações a partir dos axiomas formalizados;
- ii. **Checagem de instâncias** – Classifica as instâncias de acordo com os axiomas formalizados;
- iii. **Checagem de regras SWRL** – Classifica instâncias e classes de acordo com regras SWRL.

Desse modo, a operação *Reasoner* recebe como entradas a ontologia (alvo do processo de dedução lógica), os tipos de dedução a serem executados e a opção de persistir ou não os fatos inferidos. Como saída, se espera a ontologia acrescida dos fatos inferidos. Na tabela 19 é apresentada a descrição geral da operação *Reasoner*, enquanto na figura 39 é descrito o pseudocódigo.

Tabela 19 - Descrição da operação Reasoning.

Serviço Cortex – Operação Reasoning	
Entradas:	Semantic_Description (URI/SPARQL/SERQL/CONTEXT) – Conhecimento alvo do processo de dedução lógica. Types (String[]) – Tipos de inferência a serem realizados. Flag_Persist (Boolean) – Opção para persistir ou não o conteúdo inferido.
Saídas:	Semantic_Description (Object) – Descrições semânticas acrescidas do processo de inferência.

Pseudocódigo operação Reasoning

```
1: Reasoning( Semantic_Description, Types[], Flag_Persist )
2: {
3:     foreach ( Types[] )
4:     {
5:         new res_infer = SemanticServer.infer(Semantic_Description, Type);
6:         result.add( res_infer );
7:
8:         if (Flag_Persist )
9:         {
10:            Semantic_Server.persist( res_infer );
11:        }
12:    }
13:
14:    return result;
15: }
```

Figura 39. Pseudocódigo da operação *Reasoning*.

Considerando os requisitos apresentados na seção 4.5.1.1, o serviço *web Cortex* satisfaz o requisito ii.

4.6 Personalization Utils

Um dos objetivos gerais da Web Semântica consiste em permitir que agentes de *software* sejam capazes de processar o grande volume de informações disponível na Web de modo a auxiliar os usuários na execução de tarefas. Em grande parte, a tarefa dos agentes é sintetizada em descobrir, processar e filtrar informações para os usuários. Em muitos casos, se torna interessante que o procedimento de descobrir e filtrar informações seja realizado de forma personalizada para cada usuário. Com essa necessidade em foco, é considerado no projeto da Pangea um módulo que agrupa funcionalidades relacionadas com a personalização: *Personalization Utils*.

O módulo *Personalization Utils* é composto por três pacotes: (i) *Profiles*; (ii) *Trails*; (iii) *Preferences*.

Os pacotes são descritos em mais detalhes a seguir:

- **Profile** – O pacote *Profile* reúne serviços web e ontologias que descrevem e manipulam estereótipos de usuários. Os estereótipos são modelos formalizados em OWL que permitem associar informações básicas dos usuários (ex. Idade, formação, entre outros) a perfis pré-determinados;
- **Trails** – O pacote *Trails* reúne serviços web e ontologias que descrevem e manipulam as

interações dos usuários/agentes com os componentes da Pangea, quando engajados em atividades de descobrimento de recursos. O conhecimento acerca das necessidades dos usuários/agentes (expressos por meio de suas necessidades de informação) podem ser utilizados futuramente para personalizar a recuperação de informações;

- ***Preferences*** – O pacote *Preferences* consiste em uma ontologia e serviços Web que representam e manipulam um esquema de preferências. As preferências são representadas de modo declarativo, permitindo que sejam processadas e interpretadas por máquinas. Quando associadas a conceitos de domínio, as preferências devem permitir recomendar informações aos usuários.

Ainda que os pacotes de personalização constituam um diferencial para a proposta da Pangea, eles não são tratados em detalhes neste trabalho. O objetivo dessa seção é apresentar em linhas gerais um pacote de personalização e as funcionalidades que podem ser úteis no contexto do descobrimento, integração e recomendação de dados e modelos geográficos. Nesse sentido, são apresentadas idéias gerais que podem ser consideradas em trabalhos futuros que possam vir a ser realizados no contexto da Pangea.

4.7 Language

Em sistemas de informação a linguagem natural tem papel importante na formulação da consulta por parte do consumidor de informações. Em muitos casos, é através da linguagem que o usuário expressa suas necessidades de informação. Nesse sentido, é importante contar com funcionalidades de processamento de linguagem para o processamento da consulta. Na Pangea, o pacote *Language* disponibiliza recursos que podem ser úteis no processamento das consultas dos usuários, expressas em linguagem natural. Os pacotes que compõem o módulo *Language* são descritos em mais detalhes a seguir:

- ***Tagging*** – O pacote *Tagging* reúne funcionalidades que permitem associar classes léxicas à uma sequência de palavras fornecida como entrada. A consulta do usuário na Pangea pode ser expressa em linguagem natural. Desse modo, pode ser importante em uma etapa de pré-processamento rotular as palavras com sua classe léxica. A partir desse tipo de pré-

processamento pode ser possível identificar se a palavra é um Ponto de Interesse, um operador especial, entre outros;

- ***TextClassify*** – A consulta do usuário em linguagem natural pode ser pré-processada e associada à uma hierarquia de conceitos que categorizam tipos de consultas. Pode haver consultas geográficas que requerem pontos de interesse como saída, ou então que necessitam de análise topológica. O *TextClassify* pode ser utilizado com o intuito de extrair características da consulta em linguagem natural e classifica-la para facilitar o processamento posterior da consulta;
- ***Text Information Extractor*** – Em uma consulta geográfica o usuário pode expressar nomes de lugares ou, de modo mais geral, fenômenos geográficos conhecidos. O pacote *Text Information Extractor* pode auxiliar no pré-processamento da consulta, disponibilizando serviços responsáveis por permitir extrair informações relevantes da consulta em linguagem natural.

Ainda que os pacotes de linguagem sejam de extrema importância, eles não são tratados em detalhes neste trabalho. O objetivo dessa seção é apresentar em linhas gerais um pacote de linguagem e as funcionalidades que podem ser úteis no contexto do descobrimento e integração de dados geográficos. Nesse sentido, é apresentado um modelo geral que pode ser considerado em trabalhos futuros que possam vir a ser realizados no contexto da Pangea.

Capítulo 5

Aspectos de implementação

Neste capítulo são apresentados os detalhes da implementação de parte dos módulos previstos para a arquitetura Pangea. Além disso, é apresentado um estudo de caso. Dentre os módulos e serviços web da arquitetura Pangea, apresentados no Capítulo 4, foram implementados os seguintes:

- **Camada de serviços OGC** – Foram disponibilizados serviços OGC de dados geográficos conforme os padrões WFS e WMS;
- **Semantic Server** – Foram implementadas as quatro camadas (figura 35) previstas no projeto do *Semantic Server*;
- **Semantic Middleware** – Foram implementadas as funcionalidades referentes aos serviços web: *GoalDiscovery* (item b da seção 4.3.2.2) e *Matchmaker* (item c da seção 4.3.2.2).

Como estudo de caso da integração de informação geográfica, foi definido o problema a ser tratado, que consiste em integrar dados visando dar suporte às tomadas de decisão no contexto de acidentes de derramamento de petróleo no litoral.

5.1 Processo adotado na Implementação

No desenvolvimento dos componentes da arquitetura e em decorrência das particularidades inerentes à integração de dados geográficos, as seguintes atividades de implementação foram definidas e executadas:

- **Disponibilização dos dados geográficos como serviços OGC** – O objetivo desta etapa é disponibilizar os dados geográficos utilizados no estudo de caso como serviços nos padrões WFS e WMS do OGC;
- **Implementação das ontologias de domínio** – O objetivo desta etapa é implementar as ontologias de domínio que serão utilizadas para anotar os serviços OGC. Com esse propósito, foram implementadas duas ontologias de domínio: (i) Sensibilidade Ambiental ao

derramamento de Óleo – SAO; (ii) PONTos de Interesse – POI. A descrição de cada domínio é apresentada na seção 5.2;

- **Implementação do *Semantic Server*** – Nesta etapa são implementadas as funcionalidades do *Semantic Server*. O *Semantic Server* contém os mecanismos de armazenamento, recuperação e dedução lógica em ontologias. As anotações semânticas armazenadas no *Semantic Server* são utilizadas pelo *Semantic Middleware* na resolução das consultas;
- **Implementação do *Semantic Middleware*** - Nesta etapa são implementadas as funcionalidades do *Semantic Middleware*. O *Semantic Middleware* é o componente responsável por processar as consultas e retornar os serviços web que satisfazem as consultas;
- **Disponibilização de algumas funcionalidades como serviços web** – Algumas das funcionalidades implementadas são disponibilizadas como serviços web, seguindo as convenções para serviços web Rest.

Considerando as atividades de implementação dos módulos, foram executadas as quatro etapas apresentadas a seguir:

- i. **Análise dos requisitos levantados para o componente sendo implementado** – A primeira etapa consiste em analisar os requisitos levantados para cada módulo da Pangea que é alvo do processo de implementação. Alguns dos requisitos centrais de cada módulo são adotados para guiar o processo de desenvolvimento;
- ii. **Implementação das funcionalidades e bibliotecas de suporte aos requisitos** – São implementadas bibliotecas de classes com base na análise dos requisitos realizada na etapa (i). As bibliotecas implementadas devem suportar os requisitos selecionados na etapa (i);
- iii. **Teste das bibliotecas implementadas** – Nesta etapa são realizados os testes e a integração das funcionalidades implementadas nas bibliotecas de classes. O teste final é realizado com base na execução de um estudo de caso. O estudo de caso é apresentado na seção 5.6.;
- iv. **Disponibilização de algumas funcionalidades das bibliotecas como serviços web** – São disponibilizadas como serviços web algumas das funcionalidades das bibliotecas de classes implementadas na etapa (ii) e testadas na etapa (iii);

Na execução das atividades de implementação foram utilizadas bibliotecas, aplicativos e infraestruturas de desenvolvimento de software conforme descritos a seguir:

- **Ambiente de desenvolvimento integrado** – Consiste em um aplicativo que reúne um

conjunto de ferramentas, de modo a permitir agilizar o desenvolvimento de software. Entre as ferramentas disponibilizadas estão editores, compiladores, debugger, entre outros. Para a implementação dos módulos da Pangea foi utilizado o ambiente Eclipse, versão Helios, com suporte ao *JDT – Java Development Toolkit*;

- **Frameworks para manipulação de ontologias** – Para a manipulação de ontologias foram utilizadas três bibliotecas: (i) Jena versão 2.6.3; (ii) Sesame Repository API versão 2.0; (iii) Pellet versão 2.2.0;
- **Servidor de serviços OGC** – Os dados geográficos utilizados na Pangea devem ser publicados em um servidor de serviços OGC. O GeoServer versão 2.0.2 é utilizado com esta finalidade;
- **Servidor de aplicação web para disponibilizar as funcionalidades implementadas como serviços web Rest** – Foi empregado o Tomcat versão 6.0 para disponibilizar as funcionalidades implementadas como serviços web Rest;
- **Banco de dados espacial** – Os dados geográficos utilizados no estudo de caso foram armazenados no banco de dados PostgreSQL versão 8.3.6. O PostgreSQL foi acrescido de sua extensão espacial PostGIS versão 1.3.5 para armazenar os dados geográficos;
- **Shell Scripts de suporte** – A automação de algumas atividades foi realizada por meio da construção de *shell scripts* para o interpretador de comandos *bash*. O carregamento dos dados geográficos no banco de dados PostGIS e a publicação como serviços OGC são tarefas que foram automatizadas por meio da implementação de *shell scripts*;
- **Linguagem de programação para o desenvolvimento das funcionalidades** – As funcionalidades implementadas foram desenvolvidas em linguagem Java, utilizando o compilador versão 1.5.0.
- **SIG para visualização dos dados** – Para a visualização dos dados geográficos descobertos e integrados pelo *Semantic Middleware*, foi utilizado o SIG uDIG versão 1.1.1.

Na tabela 20 são sintetizadas as informações acerca das ferramentas utilizadas na implementação.

Tabela 20 - Ferramentas utilizadas na implementação dos componentes da Pangea.

Aspecto considerado	Ferramenta	Versão	Licença
Ambiente de desenvolvimento integrado	Eclipse	Helios	Eclipse Public License v1.0
Framework para manipulação de ontologia	Jena	2.6.3	HP OpenSource License
Framework para manipulação de ontologia (utilizado com foco nos mecanismos de armazenamento de ontologias)	Sesame	2.0	BSD License
Framework para manipulação de ontologia (utilizado com foco nas funcionalidades de dedução lógica e inferência)	Pellet	2.2.0	AGPL v3
Servidor de serviços OGC	GeoServer	2.0.2	GNU General Public License v2
Servidor de aplicação para disponibilizar os serviços web	Tomcat	6.0	Apache Foundation License
Banco de dados	PostgreSQL	8.3.6	Postgre License
Extensão espacial para o banco de dados	PostGIS	1.3.5	GNU General Public License
Scripts para automatizar tarefas	Bash Script	3.2.17	GNU General Public License
Linguagem de programação para o desenvolvimento das funcionalidades	Java	1.5.0	Sun Microsystems License
Sistema de Informação geográfica para a visualização dos serviços OGC descobertos	uDig	1.1.1	GNU General Public License

Nas próximas seções são apresentadas em detalhes as atividades desenvolvidas durante a implementação.

5.2 Disponibilização dos dados geográficos como serviços OGC

Nesta etapa foram implementadas ferramentas para dar suporte à publicação dos dados geográficos como serviços OGC. Os dados geográficos utilizados no estudo de caso (apresentado na seção 5.7) são referentes à 2 domínios distintos: Sensibilidade Ambiental ao derramamento de Óleo e Pontos de interesse. Os dados disponíveis desses domínios são originalmente representados em *shapefiles* e foram portados para o banco de dados PostgreSQL, acrescido de sua extensão

espacial PostGIS. Considerando o universo de dados (82 *shapefiles*) distintos a serem portados para o banco de dados, foi necessário implementar um *script* (APÊNDICE A) para a automação do processo.

Após armazenar os dados no PostGIS, o segundo passo consiste em publicar os dados como serviços OGC no GeoServer, que consiste em:

- i. **Criar um *DataStore*** – O *DataStore* é o componente de ligação entre o banco de dados e o GeoServer. Para a criação do *DataStore* devem ser fornecidas as informações de conexão com o banco de dados. O *DataStore* criado estabelece os meios para o acesso às tabelas armazenadas no banco de dados;
- ii. **Publicar as tabelas do banco de dados como *feature types*** – Para que as tabelas possam ser disponibilizadas como serviços OGC é necessário publicar as tabelas como *feature types*.

Um *shell script* foi desenvolvido para automatizar a criação dos *DataStores* (APÊNDICE B) e outro foi implementado para a publicação das tabelas do banco de dados como *feature types* (APÊNDICE C). Para a criação dos *DataStores*, o *shell script* processa um arquivo XML que descreve as configurações do *DataStore*. Com base nas informações descritas no arquivo XML, o *DataStore* é criado. Procedimento semelhante ocorre na publicação das tabelas do banco de dados como *feature types*: um arquivo XML descritor de *feature types* é processado por um *shell script* (APÊNDICE C) e suas informações são utilizadas para publicar as *feature types* no servidor de dados geográficos. Na figura 40 é ilustrado o arquivo de descrição de *DataStores* e na figura 41 é apresentado o arquivo de descrição das *feature types*.

```
<datastores>
  <dataStore>
    <name>isl</name>
    <connectionParameters>
      <host>localhost</host>
      <port>5432</port>
      <database>isl</database>
      <user>postgres</user>
      <password>postgres</password>
      <dbtype>postgis</dbtype>
    </connectionParameters>
  </dataStore>
  .
  .
  .
</datastores>
```

Figura 40. Arquivo de descrição dos *DataStore*.

```
<dataStore id="isl">
  <featureType>
    <name>barra_de_rio</name>
  </featureType>
  <featureType>
    <name>banco_de_lama</name>
  </featureType>
  <featureType>
    <name>barrancos</name>
  </featureType>
  <featureType>
    <name>praias</name>
  </featureType>
  <featureType>
    <name>estruturas_artificiais</name>
  </featureType>
</dataStore>
```

Figura 41. Arquivo de descrição das FeatureTypes.

Uma vez finalizado o procedimento de publicação das tabelas como *feature types*, é possível verificar se o dado geográfico foi publicado como um serviço OGC. Para tanto é feita uma requisição HTTP à operação *GetCapabilities* do GeoServer. A resposta à requisição é apresentada na figura 42.

```

<wfs:WFS_Capabilities version="1.1.0"
xsi:schemaLocation="http://www.opengis.net/wfs http://localhost:8080/
geoserver/schemas/wfs/1.1.0/wfs.xsd" updateSequence="89">

  <ows:ServiceIdentification>
    <ows:Title>GeoServer Web Feature Service</ows:Title>

    <ows:Abstract>
      This is the reference implementation of WFS 1.0.0 and WFS
      1.1.0, supports all WFS operations including Transaction.
    </ows:Abstract>

    <FeatureTypeList>
      <FeatureType>
        <Name>isl:banco_de_lama_cubatao</Name>
        <Title>banco_de_lama_cubatao</Title>
        <DefaultSRS>urn:x-ogc:def:crs:EPSG:4326</DefaultSRS>
        <ows:WGS84BoundingBox>
          <ows:LowerCorner>350230.0 7349820.5</ows:LowerCorner>
          <ows:UpperCorner>367851.188 7359740.5</ows:UpperCorner>
        </ows:WGS84BoundingBox>
      </FeatureType>
    </FeatureTypeList>

    .
    .
    .

  </ows:ServiceIdentification>

```

Figura 42. Fragmento da resposta resultante da chamada à operação GetCapabilities.

A próxima etapa da implementação consiste na construção das ontologias de domínio que são utilizadas para anotar os dados e também na resolução das consultas.

5.3 Implementação das ontologias de domínio

As ontologias de domínio são utilizadas na Pangea para anotar e descobrir os serviços OGC. Para suportar o estudo de caso desenvolvido neste trabalho, foram implementadas as seguintes ontologias de domínio: (i) Sensibilidade Ambiental ao Derramamento de Óleo; (ii) Pontos de interesse.

Além das ontologias de domínio implementadas, foram reutilizadas as seguintes ontologias: (i) Ontologia de descrição de serviços web OWL-S; (ii) geoOntologies para representar os conceitos e relações geográficas.

A seguir são apresentados em mais detalhes os domínios modelados e são ilustrados fragmentos das ontologias desenvolvidas. As ontologias foram implementadas utilizando o

aplicativo Protégé versão 4.0.

a) Cartas de Sensibilidade Ambiental ao Derramamento de Óleo

As cartas de Sensibilidade a Derrames de Óleo, conhecidas como cartas SAO, são documentos cartográficos que servem como ferramenta crítica no planejamento e nas respostas a acidentes envolvendo o derramamento de óleo. As cartas SAO são utilizadas como um componente essencial e fonte de informação primária para o planejamento de contingência e avaliação dos danos ambientais em casos de derrame de óleo; representam um instrumento fundamental para o balizamento das ações de resposta a vazamentos de óleo, por possibilitarem a identificação dos ambientes mais sensíveis, o que orienta o direcionamento dos recursos e induz a uma maior eficiência das equipes de apoio em campo.

Os tipos mais comuns de mapas de sensibilidade ambiental utilizados no mundo são baseados num sistema de informação espacial que inclui três componentes: um sistema de classificação da sensibilidade dos ambientes costeiros por meio do Índice de Sensibilidade do Litoral (ISL), os recursos biológicos passíveis de serem impactados por derrames e os recursos sócio-econômicos (de valor comercial, recreacional ou de subsistência) envolvidos.

A elaboração das cartas SAO demanda grande quantidade de informações descritivas e espaciais, de modo que as sensibilidades dos vários ambientes sejam analisadas e representadas com a maior segurança possível.

Neste trabalho, foi construída uma ontologia de domínio para suportar a representação dos conceitos e relações referentes às cartas SAO. Nesta ontologia são representados os ambientes costeiros, os índices de sensibilidade associados, recursos biológicos, sócio-econômicos e as informações sobre os procedimentos de limpeza adequados à cada ambiente. O principal aspecto modelado na ontologia desenvolvida consiste em relacionar os ambientes costeiros aos índices de sensibilidade e também aos procedimentos de limpeza. Para atingir esse objetivo foram criadas duas propriedades *hasISL* e *hasCleaningProcedure*. Um fragmento da taxonomia desta ontologia, onde são destacados alguns dos procedimentos de limpeza, é ilustrado na figura 43.



Figura 43. Fragmento da ontologia SAO (destaque para os procedimentos de limpeza).

b) Pontos de interesse

A busca por pontos de interesse é um domínio bastante referenciado ao longo da história da humanidade. Tal importância decorre da necessidade dos seres humanos de se localizarem e movimentar no espaço tendo por objetivo alcançar novas localizações.

Uma rota é a definição de um caminho, mediante alguns critérios, entre dois pontos: a origem e o destino. Um ponto de interesse pode ser definido, se aproveitando da definição de rota, como uma especialização dos pontos de origem e destino. Os pontos de interesse geralmente representam empreendimentos, serviços, locais de entretenimento, entre outros.

Os sistemas de rotas e pontos de interesse fazem parte do cotidiano das pessoas. A ampla adoção dos sistemas de navegação *Turn-By-Turn* baseados na tecnologia de posicionamento GPS (*Global Positioning System*), bem como a disponibilidade de serviços como o Google Maps e

Google Earth tem contribuído para popularizar essas funcionalidades entre as pessoas. Na área de negócios, sistemas de rotas e logística são empregados de modo a possibilitar melhor planejamento e, conseqüentemente, redução de custos. No Turismo, os planos de visitaç o e acesso aos pontos turisticos s o orientados por sistemas desenvolvidos dentro desse dom nio.

No contexto deste trabalho foi implementada uma ontologia de dom nio para representar os principais tipos de pontos de interesse. O levantamento dos tipos de pontos de interesse foi baseado no *NAVTEQ Manual* que apresenta a modelagem de dados realizada pela NAVTEQ para representar os pontos de interesse. Um fragmento da hierarquia de classes da ontologia   apresentado na figura 44.

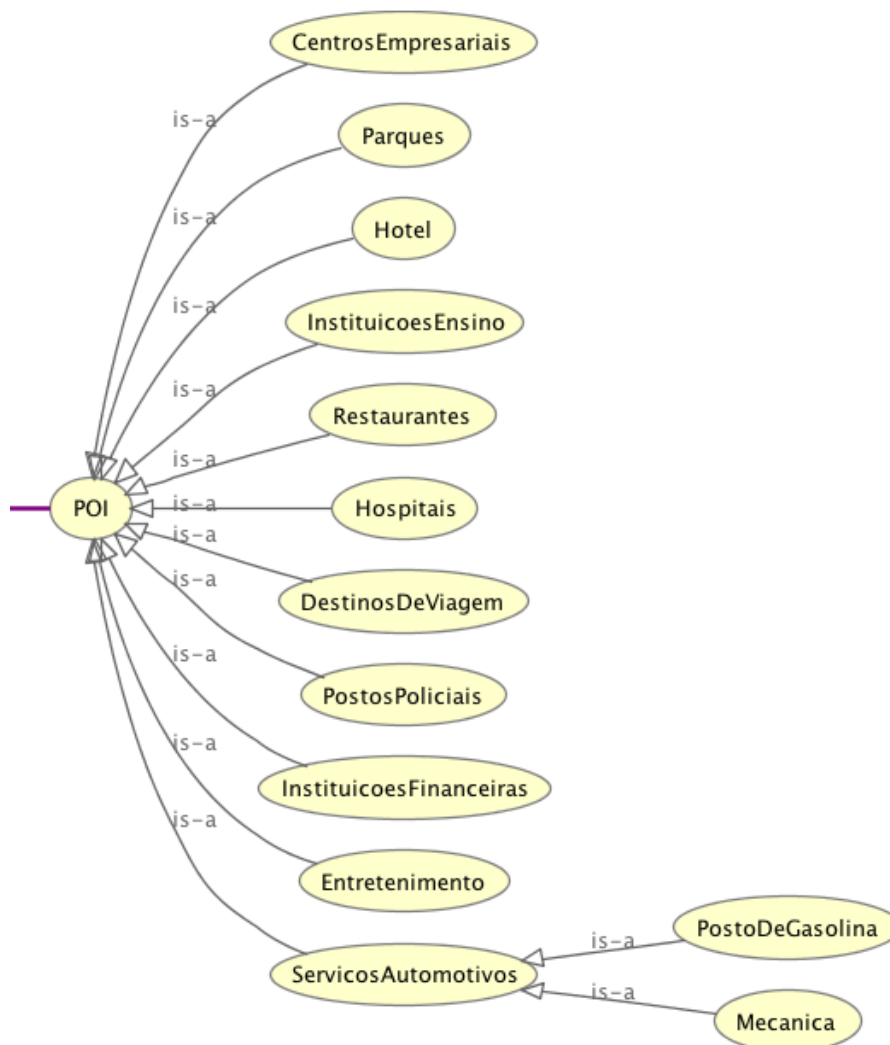


Figura 44. Fragmento da ontologia POI.

Al m das ontologias desenvolvidas, foram reutilizadas na Pangea dois conjuntos de ontologias:

- **OWL-S Framework** – Consiste no conjunto de conceitos e rela es utilizados para descrever servi os web. O *framework* de descri o   composto por quatro ontologias: (i)

Service; (ii) *Profile*; (iii) *Process*; (iv) *Grounding*.

- **GeoOntologies** – Consiste em três ontologias de domínio que representam aspectos geoespaciais: (i) *geoFeatures*; (ii) *geoRelations*; (iii) *geoCoordinateSystems*.

As ontologias foram armazenadas no *SemanticServer* para serem utilizadas na anotação e descobrimento dos serviços OGC.

5.4 Implementação do Semantic Server

O *Semantic Server* é o módulo responsável por armazenar e prover mecanismos de recuperação e dedução lógica em ontologias. Sua arquitetura é dividida nas camadas: (i) Repositórios; (ii) API; (iii) Serviços web. A camada de repositórios consiste nos mecanismos para armazenamento das ontologias. Os repositórios podem ser internos ou externos. A camada de API contém as classes e bibliotecas que implementam as funcionalidades para armazenar, recuperar e realizar dedução lógica em ontologias. A camada de serviços web disponibiliza algumas das funcionalidades da camada de API para acesso na Web.

Os seguintes passos foram executados na implementação do *Semantic Server*:

- a) Implementação dos repositórios.
- b) Implementação das operações de recuperação de ontologias.
- c) Implementação das operações de dedução lógica.
- d) Armazenamento das ontologias desenvolvidas nos repositórios.

a) Implementação dos repositórios

A implementação dos repositórios consistiu basicamente em especificar e implementar interfaces padrão para os diversos tipos de repositórios suportados no *Semantic Server*. Os tipos de repositórios implementados foram baseados nos requisitos dos módulos da Pangea e também nos tipos de repositórios suportados pelo *framework* Sesame e Jena. Neste contexto, os repositórios suportados pelo Sesame e considerados na implementação do *Semantic Server* são:

- **Main memory RDF** – Repositório que utiliza a memória como mecanismo de armazenamento e empregado em repositórios pequenos, que não necessitam de suporte à inferência e dedução lógica e, além disso, precisam fornecer acesso rápido aos dados;
- **Native RDF** – Repositório que utiliza o disco rígido como mecanismo de armazenamento. Adota um formato binário de armazenamento, otimizado para permitir ocupar pouco espaço

em disco e recuperação de dados rápida. Nesse tipo de repositório é possível configurar o suporte à inferência e dedução lógica, considerando as primitivas contidas nas especificações RDF e RDFS;

- **HTTP Remote** – Consiste em uma interface de acesso a repositórios externos, acessíveis por meio do protocolo HTTP. Por meio dessa interface, qualquer repositório suportado pelo Sesame pode ser referenciado e acessado utilizando o protocolo HTTP;
- **Relational Database** – Consiste em um repositório que armazena seus dados em um banco de dados relacional. Este tipo de repositório é mais lento em relação ao *Native RDF*, no entanto, há vantagens decorrentes da adoção de um sistema gerenciador de banco de dados, como o suporte ao acesso concorrente.

Os repositórios suportados pelo Jena e considerados na implementação do *Semantic Server* são:

- **SDB** – Consiste em um repositório que utiliza um banco de dados relacional para armazenar as ontologias.
- **TDB** – Consiste em um repositório escrito em Java e que utiliza o sistema de arquivos do sistema para armazenar as ontologias.

Considerando os diversos repositórios mencionados, foi definida uma interface padrão para os repositórios e implementada uma classe abstrata denominada *AbstractRepository*. A descrição detalhada da classe *AbstractRepository* é apresentada na tabela 21.

b) Implementação das operações de recuperação de ontologias

Uma vez que os diversos tipos de repositórios tenham sido implementados e padronizados por meio de interfaces, foram implementadas as operações de recuperação de informação semântica. A implementação das operações para a recuperação de informação semântica foi feita utilizando as bibliotecas Sesame e Jena.

Tabela 21 - Descrição dos métodos da classe *AbstractRepository*.

Classe <i>AbstractRepository</i>			
Método	Parâmetros	Retorno	Descrição
initialize	-	-	Inicializa o repositório semântico.
shutdown	-	-	Fecha todas as conexões com o repositório e retira da lista de repositórios disponíveis.

Classe AbstractRepository			
Método	Parâmetros	Retorno	Descrição
publish	SemanticDescription (Object), baseURI (String), context (String)	-	Publica ontologias no repositório. As triplas já existentes não são duplicadas.
doTupleQuery	Query (String)	Result (JSONArray)	Executa uma consulta no repositório e retorna um JSONArray contendo as variáveis consultadas e os respectivos valores.
doBooleanQuery	Query (String)	Result (JSONArray)	Executa uma consulta no repositório e retorna true ou false no caso do padrão expresso na consulta estar presente ou não no repositório.
doGraphQuery	Query (String)	Result (JSONArray)	Executa uma consulta no repositório e retorna um conjunto de triplas RDF.
getStatementsBySubject	Subject (URI)	Result (JSONArray)	Executa uma consulta no repositório recuperando todas as triplas referentes a um sujeito fornecido como parâmetro.
getStatementsByContext	Context (String[])	Result (JSONArray)	Executa uma consulta no repositório recuperando todas as triplas referentes aos contextos fornecidos como parâmetro.
closeConnection	-	-	Fecha conexão com repositório.
setNServiceProfiles	-	-	Seta o número de OWL-S Service Profiles disponíveis no repositório.
getNServiceProfiles	-	NserviceProfiles (Integer)	Retorna o número de OWL-S Service Profiles disponíveis no repositório.
testConnection	-	Boolean	Testa se uma conexão está ativa.
queryResult2JSON	QueryResult (TupleResultSet GraphResultSet ResultSet)	JSONArray	Transforma os variados tipos de resposta provenientes dos repositórios para um formato padrão representado em um JSONArray.

Com base na classe abstrata *AbstractRepository* foram implementados os repositórios elencados na tabela 22.

Tabela 22 - Repositórios implementados no *Semantic Server*.

Tipo de repositório	Descrição
MemoryRepository	Tipo de repositório que armazena as triplas na memória. É possível configurar o repositório de modo que utilize um arquivo em disco para persistir o conteúdo que está na memória. É baseado no Main Memory RDF do Sesame.
NativeRepository	Tipo de repositório que armazena as triplas em disco. É possível associar índices para organizar o conteúdo dos arquivos de modo a permitir otimizar a recuperação das triplas. Sua implementação é baseada no Native RDF do Sesame.
HTTPRepository	Interface HTTP para acesso a repositórios remotos. O repositório HTTP pode ser implementado como um Memory, Native, TDB, SDB. A interface HTTP permite a acessar as funcionalidades do repositório remotamente. Sua implementação é baseada no HTTP Remote do Sesame e é estendida para suportar os tipos SDB e TDB.
RdbmsRepository	Armazena as triplas em um banco de dados relacional. É baseado no tipo Relational Database do Sesame.
TDBRepository	Armazena as triplas em disco, permitindo o acesso através de classes do framework Jena. É baseado no repositório TDB do Jena.
SDBRepository	Armazena as triplas em um banco de dados relacional. É baseado no repositório SDB do Jena.
GoalRepository	Consiste no repositório de objetivos. Os objetivos são descrições semânticas gerais das necessidades de informação de um usuário/agente. O GoalRepository permite agrupar conceitualmente os objetivos em um repositório específico.
ServiceProfileRepository	Consiste no repositório de OWL-S Service Profiles. Os Service Profile são descrições semânticas associadas aos serviços OGC. É a partir dos Service Profiles que os serviços OGC são descobertos.

c) Implementação das operações de dedução lógica

Uma interface padrão para o componente de dedução lógica foi especificada com base nas funcionalidades do *framework* Pellet. Além disso, foram implementadas funcionalidades para facilitar o acesso à funcionalidades de dedução lógica comumente utilizados pelo *Semantic Middleware*, como por exemplo, checar se um conceito é mais geral ou específico do que outro. Tais funcionalidades foram implementada na classe *Reasoner*. Os métodos da classe *Reasoner* são apresentados na tabela 23.

Tabela 23 - Descrição dos métodos da classe *Reasoner*.

Classe Reasoner			
Método	Parâmetros	Retorno	Descrição
isEquivalentConcepts	ConceptA (URI), ConceptB (URI)	Boolean	Verifica se dois conceitos são logicamente equivalentes. A verificação é feita pela análise dos URIs e pela propriedade owl:equivalentClass.
isSubConcept	ConceptA (URI), ConceptB (URI)	Boolean	Verifica se o conceito A é mais específico do que o conceito B.
isDirectSubConceptOf	ConceptA (URI), ConceptB (URI)	Boolean	Verifica se o conceito A é uma subclasse imediata do conceito B.
isDirectSuperConcept	ConceptA (URI), ConceptB (URI)	Boolean	Verifica se o conceito A é uma superclasse imediata do conceito B.
classifyTaxonomy	IdTaxonomy	InferredModel	Realiza uma nova classificação da taxonomia, baseada nos axiomas presentes na ontologia. É implementado com base no classificador do Pellet.

d) Armazenamento das ontologias desenvolvidas nos repositórios

Após a implementação das funcionalidades apresentadas nos passos anteriores, o próximo passo consiste em armazenar as ontologias desenvolvidas nos repositórios. Para permitir automatizar essa tarefa foi implementado um descritor de repositórios em XML. Este descritor contém as informações básicas dos repositórios e indica onde se encontram as descrições semânticas que devem ser armazenadas. As informações principais acerca dos repositórios são as seguintes:

- **Tipo de repositório** – Indica qual o tipo de repositório a ser criado. Os tipos suportados são os listados na tabela 22.
- **Contexto** – Indica a criação de um contexto no repositório sendo criado. Um contexto serve para agrupar logicamente descrições semânticas.
- **Descrição semântica** – Contém as informações básicas sobre a descrição semântica a ser carregada no repositório, considerando o contexto específico.
- **Tipo de descrição semântica** – Contém informações se a descrição semântica é remota ou local.

Um fragmento do descritor de repositório implementado é apresentado na figura 45.

```

<repositories>
  <repository type="rdbms" id="sesame" database="sesame"
    host="localhost" user="postgres" password="postgres">
    <context id="wsDescription" uri="http://www.caiena.net/
      contexts/wsdescription">
      <knowledge id="ServiceProfile" type="remote" url="http://
        www.ai.sri.com/~daml/services/owl-s/1.1/Profile.owl"/>
      <knowledge id="Service" type="remote" url="http://
        www.ai.sri.com/~daml/services/owl-s/1.1/Service.owl"/>
      .
      .
    </context>
    <context id="domainDescription" uri="http://www.caiena.net/
      contexts/domaindescription">
      <knowledge id="ISLDomain" type="local" url="/Users/
        marques/ontologies/anp/anp.owl"/>
      <knowledge id="ISLDomain" type="local" url="/Users/
        marques/ontologies/POI/POI.owl"/>
      .
      .
    </context>
  </repository>
</repositories>

```

Figura 45. Fragmento do arquivo XML descritor de repositórios.

O descritor de repositórios é processado no momento em que o *Semantic Server* é inicializado. A função de processar o conteúdo do descritor de repositórios é de responsabilidade da classe *Bootstrap*. Esta classe contém o método *configureSemanticServer* que processa o conteúdo do descritor e cria os repositórios especificados. A classe *Bootstrap* é apresentada em detalhes na tabela 24.

Tabela 24 - Descrição dos métodos da classe *Bootstrap*.

Classe Bootstrap			
Método	Parâmetros	Retorno	Descrição
configureSemanticServer	Void	SemanticServer	Processa o conteúdo do descritor de repositórios. Cria os repositórios descritos e carrega as informações semânticas solicitadas. Também cria um HashMap contendo todos os repositórios especificados. Desse modo, permite a busca por informação semântica em vários repositórios.

5.5 Implementação do Semantic Middleware

O *Semantic Middleware* é o módulo responsável pelo processamento das consultas dos usuários/agentes. O processamento das consultas consiste em descobrir os serviços OGC que satisfazem uma consulta especificada por um usuário/agente. A implementação do *Semantic Middleware* contempla os seguintes aspectos previstos na arquitetura proposta (ver seção 4.3):

- **Spatial Problem Solver** – Pacote que contém os serviços web responsáveis pela resolução das consultas.
- **Goal Discovery** – Serviço web responsável por descobrir objetivos gerais que representam as demandas dos usuários/agentes, especificadas na consulta;
- **Matchmaker** – Serviço web responsável por obter o grau de correspondência entre um objetivo e as descrições dos serviços OGC.

Os serviços web e pacotes mencionados foram implementados com base no desenvolvimento de um conjunto de classes em Java. As classes Java que suportam as funcionalidades do *Semantic Middleware* são ilustradas na figura 46.

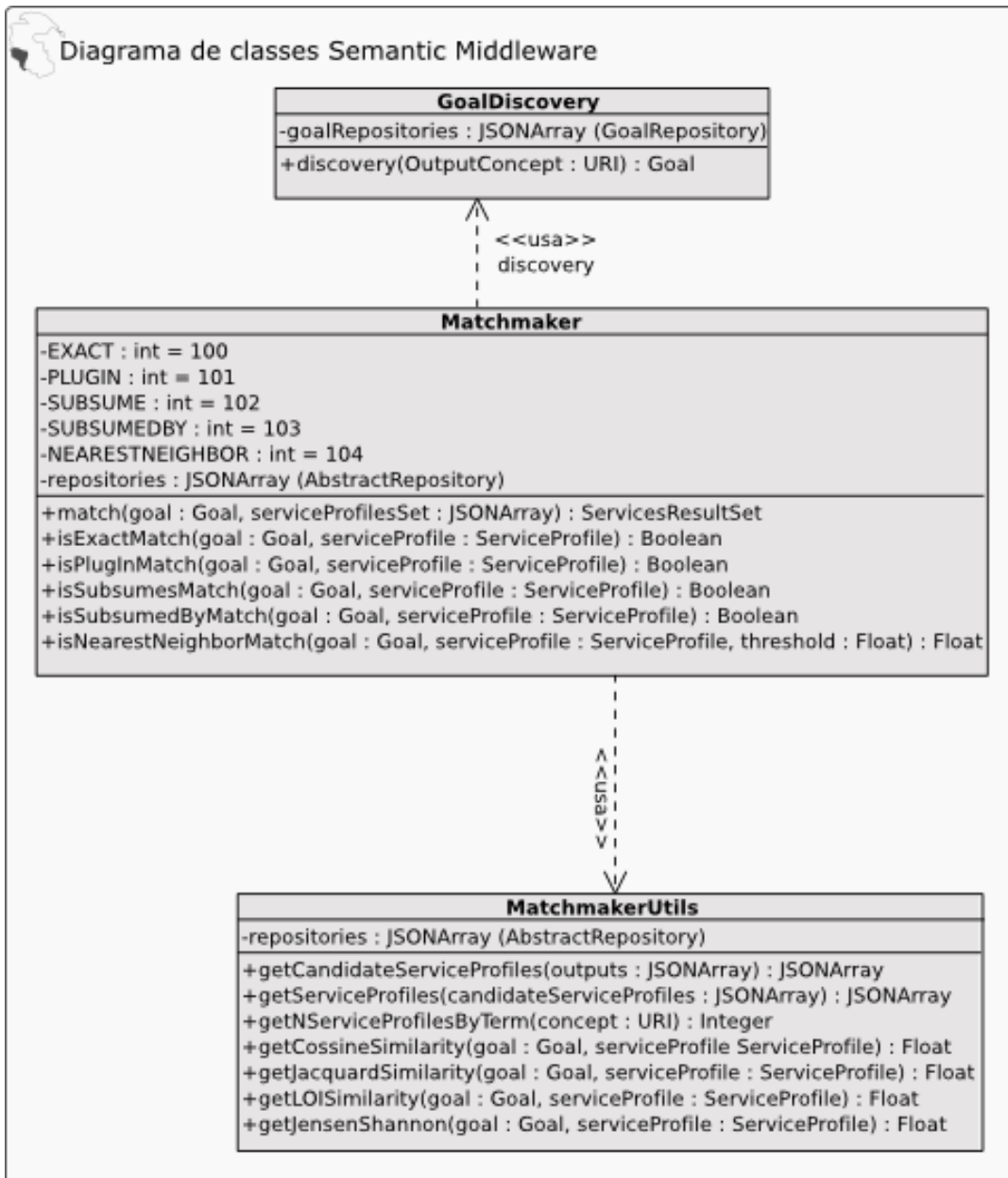


Figura 46. Classes implementadas no *Semantic Middleware*.

A seguir são apresentados em detalhes os métodos das classes ilustradas na figura 46.

Tabela 25 - Descrição dos métodos da classe *GoalDiscovery*.

Classe <i>GoalDiscovery</i>			
Método	Parâmetros	Retorno	Descrição
discovery	OutputConcept (URI)	Goal	Retorna os objetivos que tem como saída o conceito especificado em OutputConcept.

Tabela 26 - Descrição dos métodos da classe *Matchmaker*.

Classe <i>Matchmaker</i>			
Método	Parâmetros	Retorno	Descrição
match	goal (Goal), serviceProfilesSet (JSONArray)	ServicesResultSet (JSONArray)	Retorna os perfis dos serviços OGC que satisfazem o objetivo passado como parâmetro.
isExactMatch	goal (Goal), serviceProfile (ServiceProfile)	Boolean	Verifica se o objetivo e o service profile em análise tem correspondência exata.
isPluginMatch	goal (Goal), serviceProfile (ServiceProfile)	Boolean	Verifica se o objetivo e o service profile em análise tem correspondência plug-in.
isSubsumesMatch	Goal (Goal), serviceProfile (ServiceProfile)	Boolean	Verifica se o objetivo e o service profile em análise tem correspondência subsumes.
isSubsumedByMatch	Goal (Goal), serviceProfile (ServiceProfile)	Boolean	Verifica se o objetivo e o service profile em análise tem correspondência subsumedBy.
isNearestNeighbor	Goal (Goal), serviceProfile (ServiceProfile), threshold (Float)	Float	Obtém a similaridade entre o objetivo e o service profile em análise.

Tabela 27 - Descrição dos métodos da classe *MatchmakerUtils*.

Classe MatchmakerUtils			
Método	Parâmetros	Retorno	Descrição
getCandidateServiceProfiles	Outputs (JSONArray)	ServiceProfilesSet (JSONArray)	Retorna os service profiles candidatos ao Match. Para selecionar os service profiles, considera as saídas esperadas, fornecidas como parâmetro para o método.
getServiceProfiles	CandidateServiceProfiles (JSONArray)	ServiceProfilesObj (JSONArray)	Retorna um JSONArray contendo os objetos ServiceProfiles.
getNServiceProfilesByTerm	Concept (URI)	NServiceProfiles (Integer)	Retorna o número de service profiles que fazem referência a um determinado conceito.
getCossineSimilarity	goal (Goal), serviceProfile (ServiceProfile)	SimilarityDegree (Float)	Retorna o grau de similaridade entre um objetivo e um serviceProfile com base na distância de cosenos.
getJacquardSimilarity	goal(Goal), serviceProfile (ServiceProfile)	SimilarityDegree (Float)	Retorna o grau de similaridade entre um objetivo e um serviceProfile com base na medida JacquardSimilarity.
getLOISimilarity	goal(Goal), serviceProfile (ServiceProfile)	SimilarityDegree (Float)	Retorna o grau de similaridade entre um objetivo e um serviceProfile com base na medida Loss of Information.
getJensenShannon	goal(Goal), serviceProfile (ServiceProfile)	SimilarityDegree (Float)	Retorna o grau de similaridade entre um objetivo e um serviceProfile com base na medida JensenShannon.

Na próxima seção são apresentados os mecanismos utilizados para disponibilizar as funcionalidades das bibliotecas implementadas como serviços web.

5.6 Disponibilização como serviços web

A proposta arquitetural da Pangea é baseada em serviços web agrupados em pacotes e

módulos. Nesta etapa da implementação, algumas das funcionalidades das bibliotecas de classes (descritas nas seções 5.4 e 5.5) foram disponibilizadas como serviços web do tipo Rest. Os serviços *Rest – Representational State Transfer* são baseados em requisições HTTP à operações identificadas por um URI. Essas operações são acessadas pelos métodos tradicionais presentes no HTTP (i.e. POST, GET, DELETE e PUT). Serviços web segundo as convenções Rest não necessitam adotar uma camada sobre o HTTP para serem implementados (como é o caso dos serviços baseados em SOAP por exemplo).

Os serviços web foram implementados como classes Java do tipo *Servlet* e foram disponibilizadas por meio do servidor de aplicação Tomcat. Os *Servlets* são tipos especiais de classes Java que são executados no lado servidor. São utilizados para processar requisições dinamicamente de aplicações clientes, criando desse modo um mecanismo para agregar funcionalidades ao servidor.

Um ponto importante da implementação se refere ao formato de troca de mensagens entre o cliente e os serviços web. O formato das mensagens é o *Java Script Object Notation – JSON*. O formato JSON consiste em uma notação baseada em texto para representar objetos. É adotada na Pangea como formato de transferência de dados devido à sua simplicidade e compatibilidade com a maior parte dos clientes web disponíveis.

Nas tabelas 28 e 29 são apresentados os mapeamentos entre os aspectos projetados no Capítulo 4, as interfaces disponibilizadas como serviços web Rest e, por fim, as classes e operações implementadas nas bibliotecas e utilizadas pelos serviços web. A interface das classes e operações das bibliotecas de classes foi negligenciada para facilitar a leitura. Os detalhes das interfaces já foram apresentados anteriormente.

Tabela 28 - Mapeamento entre projeto e implementação considerando o módulo *Semantic Server*.

Módulo Semantic Server		
Interface do serviço web projetado	Serviço web implementado	Classes e operações utilizadas das bibliotecas de classes
Publisher.publish (Context (String), Semantic_Description (URL), Type (String))	Publisher.publish (JSONArray parms)	AbstractRepository.publish ()
Oracle.getKnowledgeByContext (Context (String))	Oracle.getKnowledgeByContext (JSONArray parms)	AbstractRepository.getStatementsByContext ()

Módulo Semantic Server		
Interface do serviço web projetado	Serviço web implementado	Classes e operações utilizadas das bibliotecas de classes
Oracle.getKnowledge (Pattern (SPARQL/Keyword))	Oracle.getKnowledge (JSONArray parms)	AbstractRepository.getStatementsByContext () AbstractRepository.doTripleQuery ()
Cortex.reasoner (SemanticDescription (URL/SPARQL/CONTEXT), Types (String[]), Flag_Persist)	Cortex.reasoner (JSONArray parms)	Reasoner.classifyTaxonomy () AbstractRepository.publish ()

Tabela 29 - Mapeamento entre projeto e implementação considerando o módulo *Semantic Middleware*.

Módulo Semantic Middleware		
Interface do serviço web projetado	Serviço web implementado	Classes e operações utilizadas das bibliotecas de classes
SpatialProblemSolver (SearchGraph (URL), SolutionGraph (JSONArray))	SpatialProblemSolver (JSONArray parms)	GoalDiscovery.discovery () Matchmaker.match()
GoalDiscovery.discovery (NodeGraph (URL))	GoalDiscovery.discovery (JSONArray parms)	GoalDiscovery.discovery ()
Matchmaker.match (Goals (JSONArray))	Matchmaker.match (JSONArray parms)	Matchmaker.match ()

5.7 Estudo de caso

O objetivo principal do estudo de caso é verificar como os componentes implementados se comportam quando considerados os procedimentos de descobrimento e integração de dados geográficos.

O contexto ao qual se refere o estudo de caso considera os acidentes de derramamento de óleo. Na ocorrência de um derramamento é necessário chegar ao local do acidente e desenvolver as atividades para reduzir os impactos. Nesse sentido, os planos de contingência aos acidentes de derramamento de óleo devem ser elaborados. Na construção do plano de contingência em resposta a um acidente de derramamento de óleo é necessário integrar diversas informações geográficas. Há inúmeras variáveis que influenciam no plano de contingência, entre elas o bioma costeiro, índice de sensibilidade, procedimentos de limpeza recomendados, equipe envolvida, proximidade a pontos logísticos fundamentais para a ação, entre outras variáveis.

Com base no exposto e considerando um estudo de caso, considera-se como exemplo um profissional engajado na construção de um plano de contingência para acidentes de derramamento, que leve em consideração a região geográfica dos municípios de São Sebastião e Ilha Bela. Essa região sofreu aumento de risco de acidentes uma vez que o Porto de São Sebastião tem se tornado ponto estratégico para inúmeros projetos relacionados ao setor de petróleo e gás, entre eles o projeto Mexilhão. A área de interesse do estudo de caso é ilustrada na figura 47.



Figura 47. Área geográfica de interesse do estudo de caso.

No estudo de caso, os dados geográficos utilizados são provenientes de dois domínios: (i) Sensibilidade Ambiental ao Derramamento de Óleo – SAO; (ii) Pontos de Interesse – POI. Os dados geográficos do domínio SAO foram coletados por pesquisadores do grupo de Sensibilidade Ambiental a Derrames de Petróleo da UNESP de Rio Claro. Os dados geográficos do domínio POI foram cedidos pela empresa NAVTEQ com base em licença de uso não comercial.

Em um cenário hipotético, o profissional encarregado do plano de contingência deve levar em consideração:

- **O índice de sensibilidade dos ambientes costeiros** – Quanto maior o índice de sensibilidade de um ambiente costeiro, maior será o impacto do acidente, requerendo mecanismos de limpeza específicos. Portanto, é fundamental para o plano de contingência saber quais os índices de sensibilidade de um determinado ambiente;
- **Logística de suporte para a equipe de combate ao desastre** – Deve considerar os hotéis e restaurantes próximos ao local do desastre, de modo a servirem de suporte à equipe que atua no combate ao desastre;
- **Logística de manutenção dos veículos envolvidos no combate ao desastre** – É interessante que o plano de contingência leve em consideração oficinas mecânicas que possam sanar os requisitos de manutenção dos veículos envolvidos no combate ao derramamento;
- **Corpo de bombeiros e suporte policial** – Em eventos de combate a derramamentos é

importante contar com a colaboração das autoridades próximas ao local do acidente, como corpo de bombeiros e polícia.

O estudo de caso leva em consideração estas demandas por informação e é baseado nas etapas necessárias para a resolução de uma consulta. A consulta considerada no estudo de caso é a seguinte: “*Ambientes com ISL igual a 8 em Ilha Bela ou São Sebastião que estejam localizados a pelo menos 2 quilômetros de uma mecânica e um posto policial*”. Neste estudo de caso, a resolução da consulta consiste em descobrir os serviços OGC que podem prover os dados para a resolução da consulta solicitada.

Nesse sentido, as etapas do estudo de caso consistem em:

- i. Disponibilizar os dados geográficos como serviços WFS do OGC utilizando o servidor de dados geográficos GeoServer;
- ii. Anotar semanticamente os serviços WFS com os conceitos das ontologias de domínio elaboradas utilizando o editor de ontologias Protege;
- iii. Publicar as ontologias de domínio e as anotações semânticas no *Semantic Server* utilizando os serviços web Rest implementados;
- iv. Resolver a consulta com base nas anotações semânticas publicadas no *Semantic Server*;

As etapas do estudo de caso são apresentadas em detalhes nas próximas seções.

5.7.1 Disponibilizar os dados geográficos como serviços OGC

Os dados utilizados são referentes a 2 domínios distintos:

- **Sensibilidade Ambiental ao derramamento de Óleo (SAO)** – Os dados que representam este domínio consistem no levantamento dos ambientes costeiros do litoral paulista, associados ao Índice de Sensibilidade do Litoral (ISL) correspondente. Além dos ambientes e do índice associado, há também o levantamento dos mecanismos de limpeza que devem ser aplicados em cada ambiente, no caso de ocorrer um derrame de óleo. O formato original através do qual os dados foram disponibilizados é o *shapefile*. Os dados deste domínio foram disponibilizados em 62 *shapefiles*;
- **POntos de Interesse (POI)** – Consiste em dados de pontos de interesse de todo o Brasil. Contém 20 tipos diferentes de pontos de interesse, agrupados em temas como: hospitais, serviços automotivos, restaurantes, serviços bancários, entre outros. Os dados foram

originalmente disponibilizados como *shapefiles*, em 20 arquivos distintos.

Utilizando os procedimentos e ferramentas apresentados na seção 5.2, os dados e modelos geográficos utilizados no estudo de caso foram publicados como serviços WFS OGC.

5.7.2 Anotação semântica dos serviços OGC

Os serviços OGC utilizados no estudo de caso devem ser anotados com os conceitos dos domínios de Sensibilidade Ambiental ao Derramamento de Óleo e de Pontos de Interesse. Uma vez que o módulo de anotação semântica previsto na Pangea (ver seção 4.4.1) não foi implementado, a anotação dos serviços OGC que serão utilizados no estudo de caso deve ser realizada manualmente. A ferramenta Protégé 4.0 foi utilizada com essa finalidade.

A anotação semântica dos serviços OGC consiste em criar os artefatos OWL-S de modo a descrever as propriedades e capacidades dos serviços OGC em um formato que possa ser processado pelo *Semantic Middleware* durante o descobrimento e integração dos serviços OGC.

Os artefatos OWL-S são organizados em quatro perspectivas: (i) *Process Model*; (ii) *Service Profile*; (iii) *Service Grounding*; (iv) *Service*. A perspectiva *Process Model* descreve como o serviço web executa suas tarefas associadas, enquanto a perspectiva *Service Profile* é utilizada com a finalidade de publicar as funcionalidades do serviço web de modo que possam ser descobertas. A perspectiva *Service Grounding* é utilizada para especificar como o serviço web deve ser invocado. Por fim, a perspectiva *Service* é utilizada para ligar as outras perspectivas em uma unidade que pode ser publicada e executada. Para o estudo de caso conduzido, os componentes *Process Model*, *Service Profile* e *Service* mencionados devem ser criados.

Para o estudo de caso foram considerados os seguintes serviços OGC:

- **Serviço WFS Ambientes** (ver figura 48) – Serviço OGC do domínio SAO que disponibiliza os dados geográficos de ambientes costeiros no litoral paulista. Os dados são provenientes da tabela “Ambientes” armazenada no PostGIS;
- **Serviço WFS PostoPolicial** (ver figura 48) – Serviço OGC do domínio POI que disponibiliza os dados geográficos referentes aos postos policiais no Brasil. Os dados são provenientes da tabela “PostosPoliciais” armazenada no PostGIS. Essa tabela possui inúmeros atributos associados como endereço do posto, telefone, responsável;

- **Serviço WFS Cidades** (ver figura 48) – Serviço OGC do domínio SAO que disponibiliza os dados geográficos referentes às cidades do litoral paulista. Os dados são provenientes da tabela “Cidades” armazenada no PostGIS. Essa tabela possui o atributo “NomeCidade” do tipo texto, cuja finalidade é representar o nome da cidade;
- **Serviço WFS PostoGasolina** (ver figura 48) – Serviço OGC do domínio POI que disponibiliza os dados geográficos referentes aos postos de gasolina no Brasil. Os dados são provenientes da tabela “PostosGasolina” armazenada no PostGIS. Essa tabela possui inúmeros atributos associados, como nome do empreendimento, tipo, endereço, telefone, entre outros.

Na figura 48 são apresentados os serviços OGC considerados no estudo de caso. Na figura são evidenciadas as entradas e as saídas que serão anotadas semanticamente, para permitir o descobrimento dos serviços OGC.

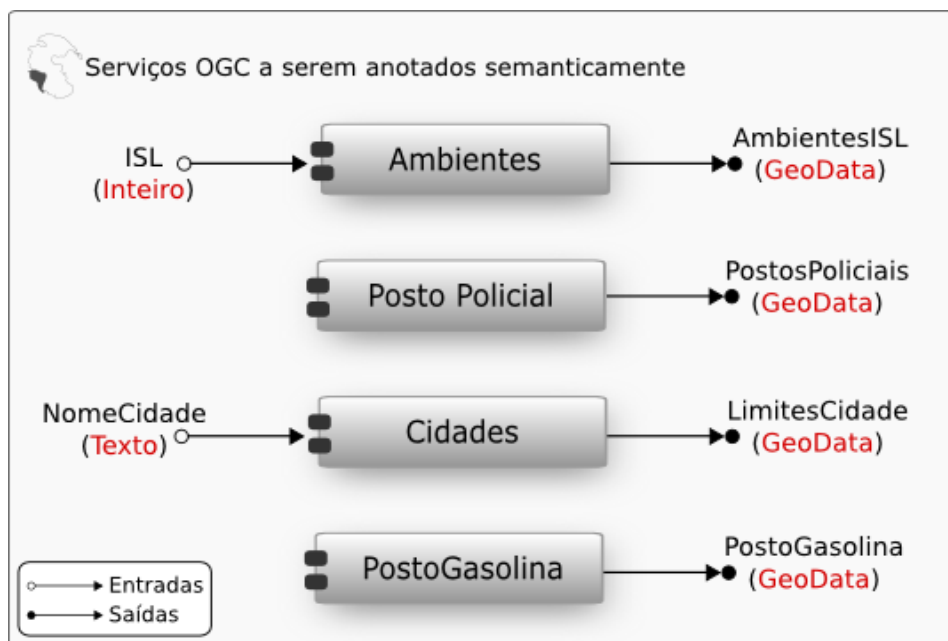


Figura 48. Serviços OGC considerados no estudo de caso.

A anotação semântica dos serviços OGC foi realizada na ferramenta Protege e consistiu das seguintes etapas:

- Importar as ontologias *Process Model*, *Service Profile*, *Service Grounding* e *Service* no Protege;

- ii. Importar as ontologias dos domínios SAO e POI para o Protege;
- iii. Para cada serviço OGC, criar uma instância do conceito *AtomicProcess* (proveniente da ontologia importada *Process Model*);
- iv. Associar às entradas e saídas da instância do conceito *AtomicProcess* (criada na etapa 2) os conceitos de domínio;
- v. Para cada *AtomicProcess* criado na etapa (iii), criar pelo menos uma instância do conceito *Profile* (proveniente da ontologia importada *ServiceProfile*). As instâncias de *Profile* são utilizadas no processo de descobrimento;
- vi. Associar as instâncias de *Profile* às instâncias de *AtomicProcess* correspondentes. A associação é feita pela propriedade *has_process* (proveniente da ontologia *Service Profile*);
- vii. Para cada *AtomicProcess* criado na etapa (iii), criar uma instância do conceito *Service* (proveniente da ontologia *Service*);
- viii. Relacionar a instância criada na etapa (vii) às instâncias criadas nas etapas (iii) e (v). A associação entre a instância do conceito *Service* e do conceito *AtomicProcess* se dá pela propriedade *describedBy*. Já a associação entre a instância do conceito *Service* e do conceito *Profile* se dá pela propriedade *presents*.

Na figura 49 é apresentado o modelo conceitual criado com a execução das etapas mencionadas anteriormente.

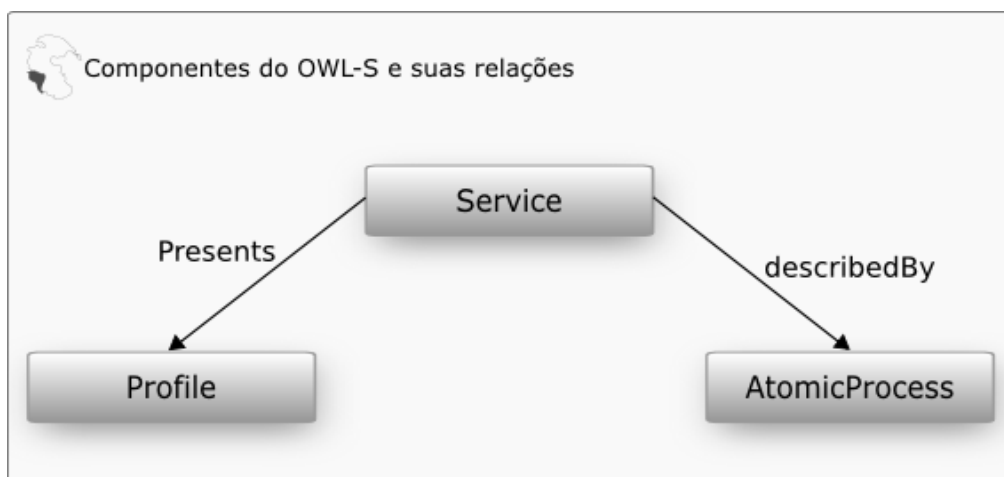


Figura 49. Modelo conceitual dos elementos criados para anotação.

Utilizando os passos anteriores foram anotados semanticamente os serviços OGC. O resultado da anotação semântica é apresentado na figura 50. As saídas e as entradas foram associadas aos conceitos das ontologias de domínio, por meio da criação dos artefatos OWL-S (figura 49).

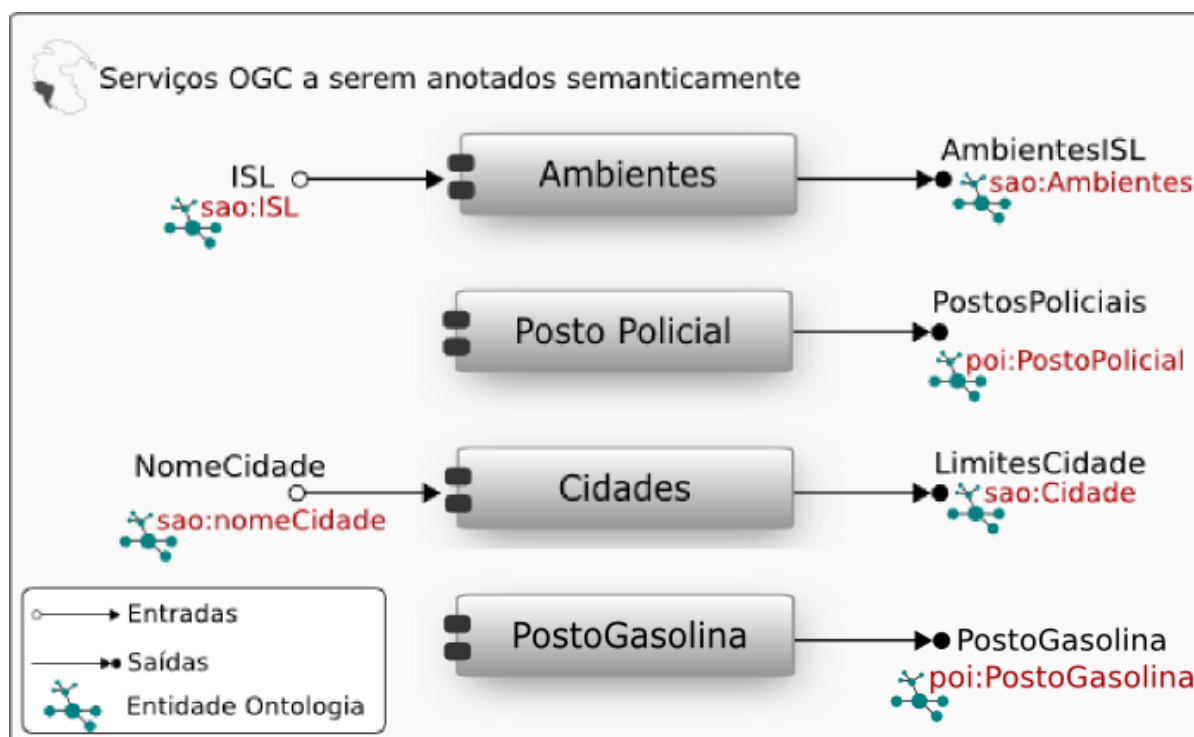


Figura 50. Serviços OGC anotados com os conceitos de domínio.

Após a anotação semântica dos serviços OGC com os artefatos OWL-S e os conceitos dos domínios SAO e POI há duas extensões ao padrão OWL-S necessárias: (i) Deve ser criado o

conceito *Goal* para representar os objetivos dos usuários/agentes (*Goals* - ver seção 4.3), necessários ao descobrimento de serviços web realizado pelo *Semantic Middleware*. (ii) Deve ser criada uma propriedade *OGCServiceURL* que é responsável por apontar para as informações referentes ao serviço OGC alvo da anotação.

Os objetivos são descrições abstratas de necessidades de informação. A noção de *Goal* advém do modelo conceitual do WSMO e de seu mecanismo de descobrimento de serviços web, baseado em classificação heurística (ver seção 3.7.2.1).

O conceito *Goal* não é considerado no *framework* OWL-S. Desse modo, deve ser criado um conceito *Goal* e associado ao modelo conceitual do OWL-S. Nesse sentido, esse conceito deve ser um subconceito de *Profile* (proveniente da ontologia *Service Profile*).

A propriedade *OGCServiceURL* deve ser associada ao conceito *Service*. As informações dessa propriedade devem ser utilizadas para invocar os serviços OGC relacionados aos *Service Profiles* recuperados.

A próxima etapa do estudo de caso consiste em publicar as anotações semânticas no *Semantic Server*. Na próxima seção são apresentados os passos necessários para a publicação das anotações semânticas.

5.7.3 Publicação das ontologias e anotações semânticas no Semantic Server

Para que as anotações semânticas sejam utilizadas no procedimento de descobrimento e integração de serviços OGC é necessário publicar as ontologias e as anotações semânticas no *Semantic Server*. Para tanto, foi criado um repositório com base em um banco de dados relacional PostgreSQL. O repositório criado foi dividido em três contextos:

- i. ***wsDescription*** – Contexto criado com o objetivo de agrupar as ontologias do *framework* OWL-S. São publicadas nesse contexto as ontologias *Service Profile*, *Service*, *Process Model* e *Grounding*;
- ii. ***domainDescription*** – Contém as ontologias dos domínios SAO e POI;
- iii. ***geoDescription*** – Contém as ontologias do domínio geográfico. São três ontologias consideradas: *getFeatures*, *geoRelations* e *geoCoordinateSystems*.

A criação do repositório foi automatizada por meio da especificação do descritor de repositórios. Na tabela 30 são apresentados os detalhes especificados no descritor de repositórios.

Tabela 30 - Descrição dos repositórios a serem criados.

Contexto	Ontologia	URL	Tipo
wsDescription	ServiceProfile	http://www.schemaweb.info/webservices/rest/GetRDFByID.aspx?id=138	Remoto
wsDescription	Service	http://www.schemaweb.info/webservices/rest/GetRDFByID.aspx?id=137	Remoto
wsDescription	Process	http://www.schemaweb.info/webservices/rest/GetRDFByID.aspx?id=136	Remoto
wsDescription	Grounding	http://www.schemaweb.info/webservices/rest/GetRDFByID.aspx?id=135	Remoto
domainDescription	ISL	file:///ontologies/isl/isl.owl	Local
domainDescription	POI	file:///ontologies/poi/poi.owl	Local
geoDomainDescription	geoFeatures	http://www.mindswap.org/2003/owl/geo/geoFeatures.owl	Remoto
geoDomainDescription	GeoRelations	http://www.mindswap.org/2003/owl/geo/geoRelations.owl	Remoto
geoDomainDescription	geoCoordinateSystems	http://www.mindswap.org/2003/owl/geo/geoCoordinateSystems.owl	Remoto

5.7.4 Representação da consulta

Um dos requisitos previstos para a Pangea é o processamento de consultas em linguagem natural para posterior resolução. No entanto, o componente responsável por processar as consultas em linguagem natural não foi alvo da implementação. Assim, neste estudo de caso as consultas são consideradas já pré-processadas e representadas de forma inequívoca como um grafo.

Para representar a consulta como um grafo foi adotado o padrão *eXtensible Graph Markup and Modeling Language* – *XGMML*. Este padrão consiste em um vocabulário XML especificado para a representação de grafos. O padrão XGMML foi estendido para suportar informações extras necessárias ao processamento da consulta. Desse modo, a consulta considerada no estudo de caso foi representada em um arquivo XML conforme o vocabulário XGMML. A representação da consulta “Ambientes com ISL igual a 8 em Ilha Bela ou São Sebastião que estejam localizados a pelo menos 2 quilômetros de uma mecânica e um posto policial” no arquivo XML é ilustrada na figura 51.

Após a consulta estar devidamente representada, a etapa seguinte consiste na resolução da

consulta que é detalhada na próxima seção.

```
<graph directed="1" id="2">
  <node id="1" label="Hotéis"
        uri="http://www.caiena.net/semanticweb/2010/6/10/ POI.owl#Hotel"
        token="GeoData"/>
  <node id="2" label="Em" uri="http://www.caiena.net/anp.owl#Em" token="Em"/>
  <node id="3" label="Ilha Bela" uri="http://www.caiena.net/anp.owl#IlhaBela"
        token="GeoData"/>
  <node id="4" label="Ou" uri="http://www.caiena.net/anp.owl#Ou" token="Ou"/>
  <node id="5" label="São Sebastião"
        uri="http://www.caiena.net/anp.owl#SaoSebastiao" token="GeoData"/>
  <node id="6" label="A" dist="2" unit="km"
        uri="http://www.caiena.net/anp.owl#A" token="A"/>
  .
  .
  .
  <edge source="1" target="2" label=""/>
  <edge source="2" target="4" label=""/>
  <edge source="3" target="4" label=""/>
  .
  .
  .
</graph>
```

Figura 51. Representação da consulta considerada no estudo de caso.

5.7.5 Resolução da consulta

A consulta a ser processada é a seguinte: “*Ambientes com ISL igual a 8 em Ilha Bela ou São Sebastião que estejam localizados a pelo menos 2 quilômetros de uma mecânica e um posto policial*”.

A resolução da consulta consiste em descobrir e associar os serviços OGC que fornecem os dados geográficos representados no nó do grafo de consulta. O descobrimento dos serviços OGC é realizado com base nas anotações semânticas relacionadas aos serviços OGC (ver seção 5.7.2).

As etapas do processamento da consulta são as seguintes:

- i. Os nós do grafo de consulta que representam dados geográficos são selecionados;

- ii. Para cada nó selecionado, o serviço web *GoalDiscovery* é invocado (var passo 1 na figura 52) com a finalidade de descobrir Objetivos (i.e. Goals) que correspondam à necessidade de informação expressa no nó. Os Objetivos são armazenados nos repositórios do *Semantic Server*. Desse modo, o serviço web *GoalDiscovery* interage com o serviço web *Oracle* (ver passo 2 da figura 52) para obter os objetivos. No caso de não ser encontrado nenhum objetivo que corresponda ao nó em análise, a necessidade de informação expressa no nó é abstraída e uma nova consulta ao *Oracle* é efetuada. Se mesmo após a generalização nenhum objetivo for encontrado, a necessidade de informação expressa no nó é transformada em um objetivo e publicada nos repositórios do *Semantic Server*;
- iii. O objetivo selecionado/descoberto na etapa anterior é utilizado para selecionar os *Service Profiles* que são candidatos para solucionar a necessidade de informação expressa no objetivo. Os *Service Profiles* descrevem os serviços OGC em função de suas entradas e saídas (ver seção 5.7.2) e estão armazenados nos repositórios do *Semantic Server*. Para recuperar os *Service Profiles*, o serviço web *Matchmaker* invoca o serviço web *Oracle* (ver passo 5 da figura 52);
- iv. Uma vez recuperados os *Service Profiles*, o serviço web *Matchmaker* verifica as relações semânticas entre as entradas e saídas expressas nos *Service Profiles* e no Objetivo, de modo a estabelecer um grau de correspondência.

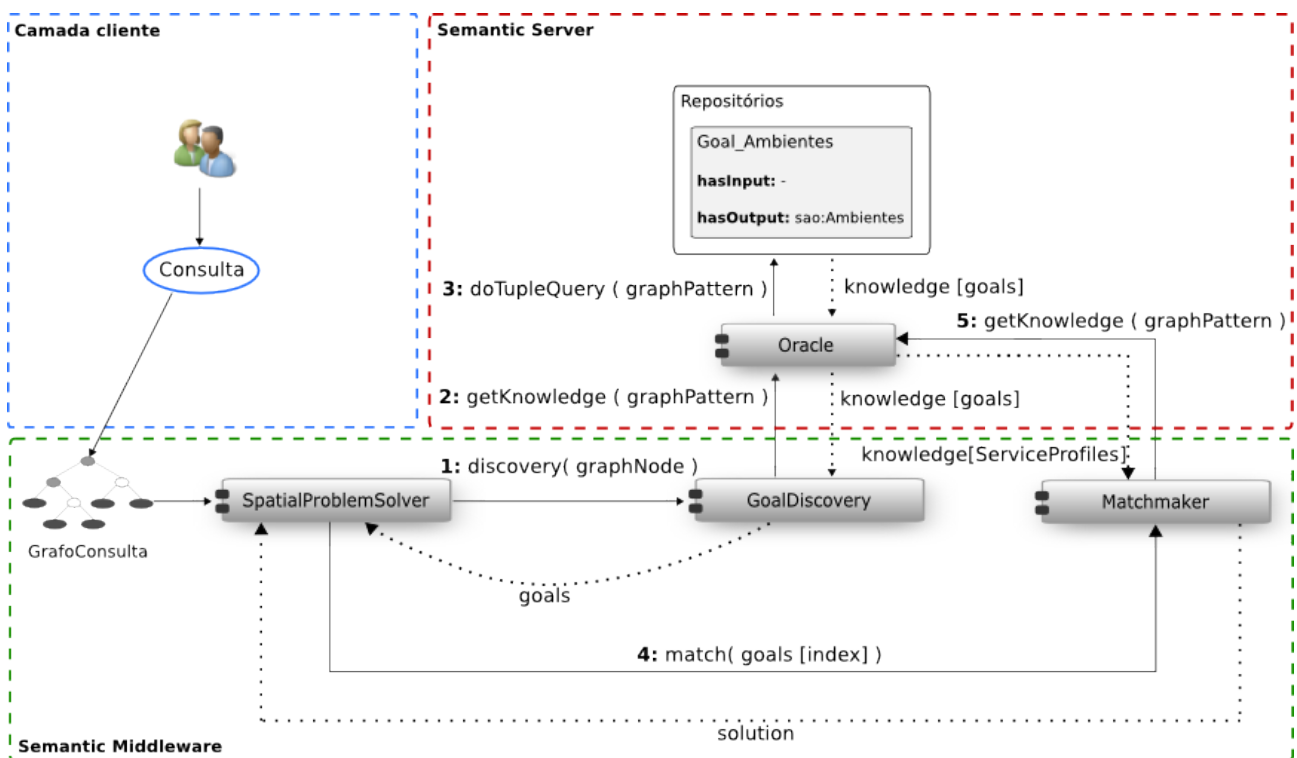


Figura 52. Esquema geral do processamento dos nós do grafo de consulta.

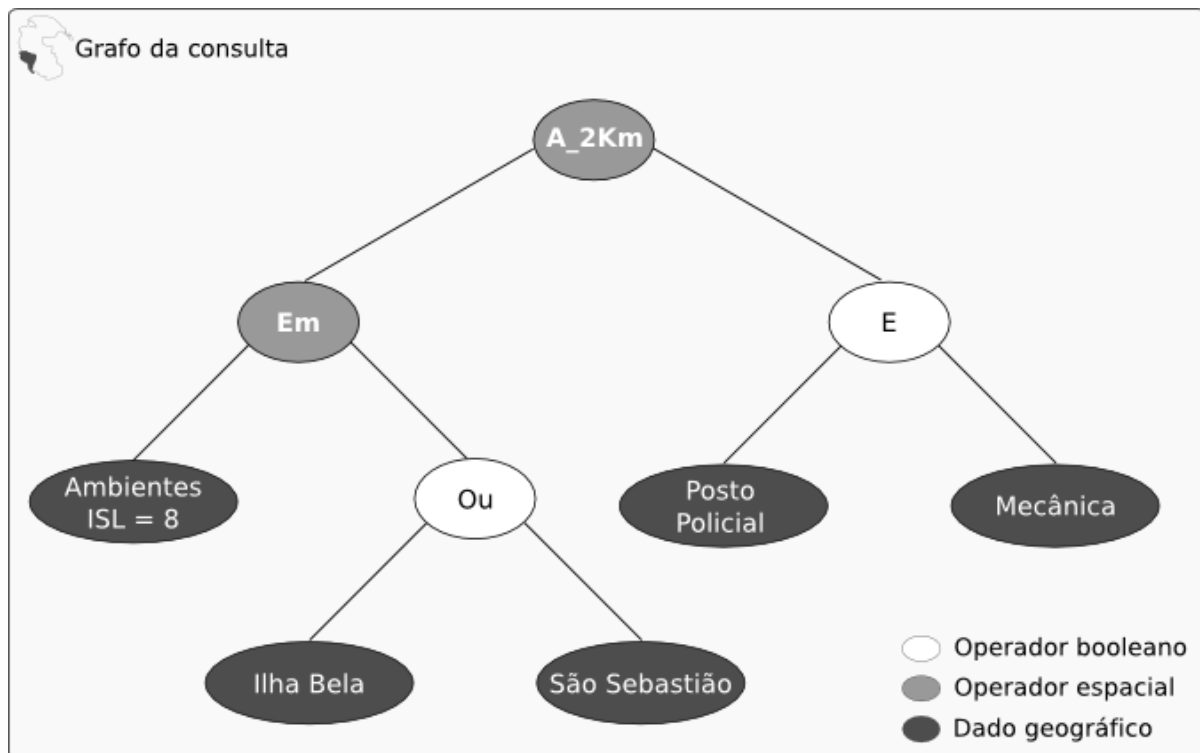


Figura 53. Grafo que representa a consulta do usuário.

Na execução do grafo, o primeiro nó selecionado é o que contém o rótulo “Ambientes ISL = 8” (figura 53). Esse nó corresponde ao conceito *sao:AmbientesISL8*. Portanto, a primeira etapa da resolução do nó consiste em invocar a operação *discovery* do serviço web *GoalDiscovery* (passo 1 na figura 52). Para descobrir os objetivos que correspondem ao nó, é realizada a consulta expressa na figura 54.

```

PREFIX sao: <http://www.caiena.net/sao#>
SELECT ?g WHERE
{
  ?g a <http://www.daml.org/services/owl-s/1.1/Profile.owl#Goal> .
  ?g <http://www.daml.org/services/owl-s/1.1/Process.owl#hasOutput> sao:AmbientesISL8
}
  
```

Figura 54. Consulta para descobrir os objetivos que correspondem ao nó.

A consulta expressa na figura 54 não encontra um objetivo correspondente, sendo necessário, portanto recuperar um objetivo abstrato. O primeiro passo é abstrair o conceito requerido pelo nó do grafo de consulta. Para tanto, é executada a consulta apresentada na figura 55. Essa consulta retorna o conceito mais geral *sao:Ambientes*. Após, é executada a consulta apresentada na figura 56, para recuperar algum objetivo que case com o conceito *sao:Ambientes*.

```

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX sao: <http://www.caiena.net/sao#>
SELECT ?c WHERE
{
  sao:AmbientesISL8 rdfs:subClassOf ?c
}

```

Figura 55. Consulta para descobrir conceitos mais gerais.

```

PREFIX sao: <http://www.caiena.net/sao#>
SELECT ?g WHERE
{
  ?g a <http://www.daml.org/services/owl-s/1.1/Profile.owl#Goal> .
  ?g <http://www.daml.org/services/owl-s/1.1/Process.owl#hasOutput> sao:Ambientes
}

```

Figura 56. Consulta para descobrimento de objetivos.

No caso da consulta expressa na figura 56, é retornado o objetivo *Goal_AmbientesCosteiros*. Esse objetivo não possui entradas anotadas, porém, contém a saída anotada com o conceito *sao:Ambientes*. Com base no objetivo selecionado, são recuperados os OWL-S *Service Profiles* que são candidatos ao conjunto solução (passo 5 na figura 52). No caso do objetivo considerado, é retornado o *Service Profile* que tem sua entrada anotada com o conceito *sao:ISL* (ver figura 50) e sua saída anotada com o conceito *sao:Ambientes* (ver figura 50). O *Service Profile* candidato e o objetivo são então analisados em função de suas entradas e saídas para verificar o grau de correspondência (as regras de correspondência são apresentadas na seção 4.3.2.1). Na figura 57 é ilustrado o objetivo abstrato e o *Service Profile* recuperados, e a correspondência semântica entre as saídas.

A resolução dos nós rotulados com “*Ilha Bela*” e “*São Sebastião*” é similar à resolução conduzida para o nó rotulado com “*Ambientes ISL = 8*” (figura 53). Ambos são representados por um objetivo abstrato cuja saída é anotada pelo conceito *sao:Cidade*. Esse objetivo corresponde exatamente ao *Service Profile* que possui como saída o conceito *sao:Cidade* (ver figura 50). A resolução do nó rotulado com “*Posto Policial*” é mais simples, na medida em que não é necessário abstrair o conceito para encontrar um objetivo correspondente. O objetivo encontrado corresponde de maneira exata ao *Service Profile* que contém a saída anotada com o conceito *poi:PostoPolicial* (ver figura 50).

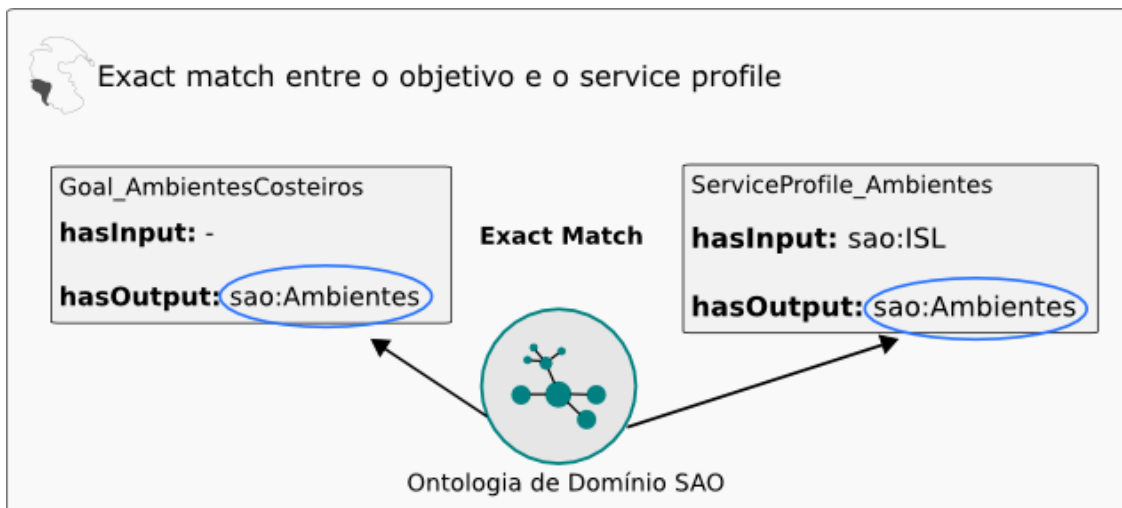


Figura 57. Nível de casamento exato entre o objetivo e a descrição do serviço OGC.

A resolução do nó rotulado com “*Mecânica*” é ligeiramente diferente. Durante o descobrimento de objetivos, é retornado por meio do procedimento de abstração um objetivo que tem sua saída anotada com o conceito *poi:ServicosAutomotivos*. Esse objetivo é utilizado para selecionar os *Service Profiles* que são candidatos ao conjunto solução. Com base no objetivo, é selecionado o *Service Profile* que possui sua saída anotada com o conceito *poi:PostoGasolina* (ver figura 50). Então, é verificado o grau de correspondência entre o objetivo e o *service profile* selecionado, resultando no grau *PlugIn Match* de correspondência, já que o conceito *poi:PostoGasolina* é mais específico do que o conceito *poi:ServicosAutomotivos*. Portanto, ainda que não haja dados geográficos sobre oficinas mecânicas, o *Semantic Middleware* retorna dados sobre postos de gasolina (que eventualmente podem auxiliar em um conserto mecânico mais simples) com base na análise das relações semânticas entre os conceitos.

Os *Service Profiles* descobertos possuem referências para as instâncias do conceito *Service* (ver figura 49). As instâncias de *Service* que se relacionam com os *Service Profiles* contém a referência ao serviço OGC por meio da propriedade *OGCServiceURL*. Desse modo, é possível invocar os serviços OGC descobertos com base nas informações fornecidas em *OGCServiceURL*.

Os serviços OGC referenciados pelos conceitos *Service* foram adicionados no uDIG (ver tabela 20), de modo a permitir visualizar os dados encontrados. É importante ressaltar que as restrições espaciais expressas no grafo de consulta não foram tratadas nesse estudo de caso. Foi apenas considerado o procedimento de descobrimento dos serviços OGC por meio das relações semânticas entre os conceitos utilizados para anotar semanticamente os serviços OGC. A visualização dos dados no uDIG é apresentada na figura 58. É importante notar que na figura 58 foi adicionado manualmente a *feature type* referente ao Porto de São Sebastião, pois este não faz parte

dos serviços OGC descobertos. Outro ponto importante a ressaltar é que os serviços OGC selecionados disponibilizam dados que, em alguns casos, extrapolam a área de interesse. Nesse caso, as *features* pertinentes à área geográfica de interesse seriam corretamente selecionadas com base no processamento das condições espaciais expressas no grafo de consulta. Além disso, seria papel do serviço web *Refine Solution* (ver seção 4.3.2.2) selecionar os serviços OGC com base em aspectos espaciais. Na legenda da figura 58, as camadas rotuladas com “Ambiente” representam ambientes diversificados, no entanto, são apresentadas com o mesmo rótulo para corresponder ao nó expresso no grafo de consulta “Ambientes”. Outro fator importante é que o filtro “ISL=8” não foi aplicado aos serviços, sendo esse papel relegado ao serviço web “*Refine Solution*”.

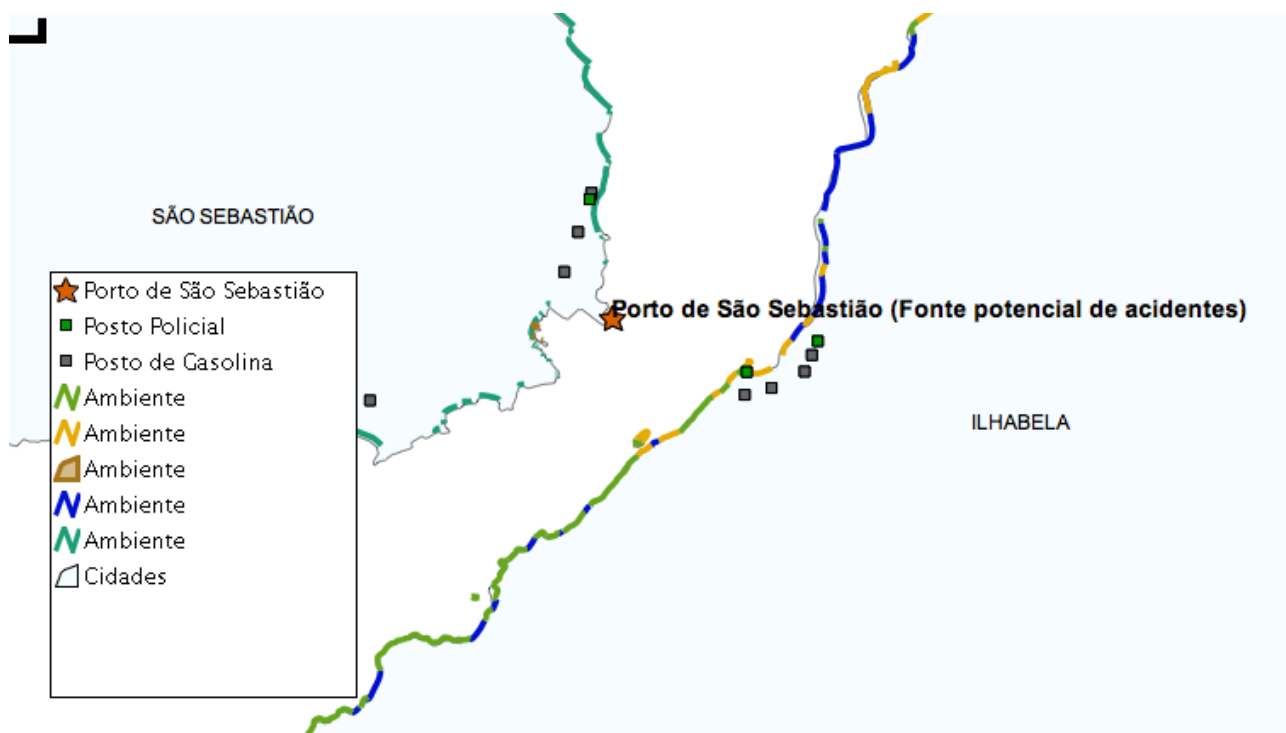


Figura 58. Visualização dos dados geográficos descobertos e integrados.

Capítulo 6

Conclusões e extensões

6.1 Conclusões

O trabalho apresentado nesta dissertação consiste na especificação de uma arquitetura que suporte os requisitos de descobrimento e integração de serviços OGC com base em descrição semântica. A arquitetura especificada foi denominada Pangea. A arquitetura é composta por módulos de (i) anotação semântica, (ii) integração de ontologias, (iii) repositórios semânticos, (iv) descobrimento e integração de dados e modelos geográficos, (v) personalização e (vi) processamento de linguagem. Dos módulos previstos para a arquitetura, foram especificados em detalhes os módulos i, ii, iii e iv. Destes, foram implementados e avaliados o módulo repositórios semânticos e as partes consideradas principais do módulo descobrimento e integração de dados e modelos geográficos. Os módulos especificados na Pangea foram propostos com o objetivo principal de permitir melhorias nos aspectos relacionados à interoperabilidade sintática e semântica de dados e modelos geográficos, principalmente levando em consideração algumas das limitações das Infraestruturas de Dados Espaciais – IDE, expostas em (DAVIS JR., C.; FONSECA, F.; CÂMARA, G, 2009).

Para atacar o objetivo principal, o trabalho foi dividido em etapas. Na primeira, foram revisadas três arquiteturas que adotam descrição semântica nos processos de descobrimento e integração de serviços OGC. Na segunda etapa, com base nos estudos das arquiteturas revisadas, foram identificados os aspectos transversais às arquiteturas. Na terceira etapa, para cada aspecto transversal levantado, foi feita revisão de alguns trabalhos relacionados aos aspectos. Com base no levantamento dos trabalhos relacionados a cada aspecto transversal, na quarta etapa foram especificados os módulos da arquitetura. Finalmente, na quinta etapa, foi implementado o módulo repositórios semânticos e partes do módulo de descobrimento e integração de serviços OGC. Para avaliar os módulos implementados foi conduzido um estudo de caso integrando dados do domínio ambiental e de pontos de interesse. O cenário adotado no estudo de caso se baseia nos planos de contingência necessários para acidentes de derramamento de óleo no litoral.

Com relação às dificuldades encontradas no desenvolvimento do trabalho, a mais marcante

é decorrente da grande quantidade de aspectos a serem estudados e tratados, quando considerado o problema do descobrimento e da integração de serviços OGC. Os aspectos mencionados fazem referência aos aspectos transversais identificados nas arquiteturas estudadas.

As principais contribuições desta dissertação são:

- Levantamento dos aspectos principais a serem considerados em arquiteturas de descobrimento e integração de serviços OGC com base em descrição semântica;
- Especificação e implementação de um repositório semântico que suporta o armazenamento, recuperação e procedimentos de dedução lógica;
- Especificação e implementação parcial de um mecanismo de descobrimento e integração de serviços OGC;
- Especificação de um módulo de anotação semântica de serviços OGC cuja característica marcante é realizar a anotação com base em enriquecimento semântico e também em dados de contorno geográfico recuperados de serviços web;
- Especificação de um módulo de integração de ontologias multicritério, baseado em técnicas levantadas na literatura.

Considerando as arquiteturas que foram estudadas para a realização deste trabalho, as principais contribuições são apresentadas na tabela 31.

Tabela 31 - Comparação entre Pangea e arquiteturas estudadas.

Arquitetura/Aspectos	WeBIOS	SWING	DAGIS	Pangea
Servidor Semântico	Sim	Sim	Não	Sim
Anotação semântica de serviços geográficos	Não	Sim	Sim	Sim
Integração de ontologias	Sim	Não	Não	Sim
Framework de descrição de serviços	Não	Sim	Sim	Sim
Descobrimto de serviços (match híbrido)	Não	Sim	Sim	Sim
Personalização	Não	Não	Não	Sim

6.2 Extensões

Algumas das muitas extensões possíveis para este trabalho são as seguintes:

- **Investigar em mais detalhes cada aspecto transversal identificado** – Neste trabalho foram levantados com base na literatura aspectos que devem ser considerados no projeto de uma arquitetura que satisfaça o requisito de descobrimento e integração de serviços geográficos. Uma extensão bastante importante seria considerar cada aspecto levantado em mais detalhes, com foco em uma especificação mais minuciosa e, além disso, efetivando as implementações e testes necessários;
- **Especificar e implementar uma arquitetura P2P para o *Semantic Server*** – A especificação proposta para o módulo *Semantic Server* é centralizada. Uma extensão bastante importante seria realizar uma especificação P2P para este módulo. A principal motivação para esta abordagem é tornar a arquitetura Pangea descentralizada e, por consequência, possibilitar melhorias na escalabilidade;
- **Especificar e implementar camada de segurança e controle de redundância no *Semantic Server*** – A especificação e implementação realizadas neste trabalho referentes ao *Semantic Server* não levam em conta muitos aspectos relacionados com segurança e controle de redundância dos dados. Apesar de na especificação atual ter sido previsto um controle de permissões de acesso, este não foi implementado;
- **Investigar os problemas relacionados às buscas distribuídas em múltiplos repositórios de ontologias** – Um problema que não foi investigado neste trabalho tem relação com a busca distribuída em múltiplos repositórios. Apesar de na especificação e implementação realizadas neste trabalho esse aspecto ser realizado, a consulta é realizada de forma “linear”, em um repositório cadastrado por vez. No caso de haver uma quantidade grande de repositórios, o tempo de resposta para a resolução de uma consulta pode ser seriamente prejudicado;
- **Investigar e especificar um plugin para o *Semantic Server* que permita conformidade com os padrões de catálogo do OGC** – Consiste em adicionar ao *Semantic Server* um módulo que permita disponibilizar funcionalidades como um catálogo conforme a especificação do OGC para catálogos;
- **Investigar e comparar outras métricas de recuperação de informação para agregar no *Semantic Middleware*** – No *Semantic Middleware* foram implementadas as métricas de

recuperação de informação propostas em (KLUSCH; FRIES; SYCARA, 2006). Uma atividade futura pode consistir em investigar, implementar e comparar o desempenho de outras métricas de recuperação de informação;

- **Investigar uma etapa de “filtragem” no *Semantic Middleware* com base em métricas de recuperação de informação** – Na especificação e implementação realizadas neste trabalho, a seleção dos *Service Profiles* candidatos ao *match* é realizada com base em dedução lógica. Um trabalho futuro poderia consistir em investigar mecanismos baseados em métricas de recuperação de informação para “filtrar” de maneira rápida os *Service Profiles* candidatos ao procedimento de *match*. Esse aspecto pode ser fundamental em casos onde haja muitos *Service Profiles* para serem verificados;
- **Especificar e implementar um mecanismo de processamento espacial das consultas** – O *Semantic Middleware* recupera os serviços geográficos porém não processa os predicados espaciais representados no grafo de consulta. Um trabalho futuro necessário seria implementar o mecanismo de processamento dos predicados espaciais contidos na consulta;
- **Especificar e implementar uma “ponte” entre o OWL-S e os serviços OGC** – Na proposta atual do OWL-S é prevista apenas interação com serviços web baseados em SOAP. Na implementação atual da Pangea, a ligação entre a descrição semântica OWL-S e os serviços OGC é feita pela modelagem de um atributo. Uma extensão necessária seria criar um *Service Grounding* específico para os serviços OGC;
- **Implementar os serviços web especificados para o módulo de anotação semântica** – Os serviços web referentes ao módulo de anotação semântica não foram implementados. Uma atividade futura pode considerar a implementação dos serviços web propostos;
- **Implementar os serviços web especificados para o módulo de integração de ontologias e analisar o desempenho do método multicritério proposto** – Os serviços de integração de ontologias não foram implementados. Nesse sentido, seria possível focar na implementação dos serviços web propostos e nos testes de desempenho do método multicritério proposto;
- **Especificar em detalhes o módulo *Personalization*** – Apesar de ser previsto na arquitetura Pangea, o módulo *Personalization* não foi especificado em detalhes. Uma extensão possível seria especificar este módulo;
- **Especificar em detalhes o módulo *Language*** - Apesar de ser previsto na arquitetura Pangea, o módulo *Language* não foi especificado em detalhes. Uma extensão possível seria especificar este módulo;

Referências bibliográficas

AHMAD, M. N.; COLOMB, R. M. **Overview of Ontology Servers Research**. *Webology*, Iran, v.4, n. 2, jun. 2007.

ALAM, A.; SUBBIAH, G.; KHAN, L.; THURASINGHAM, B. **DAGIS: A Geospatial Semantic Web Services Discovery and Selection Framework**. In: International Conference on GeoSpatial Semantics, 2., 2007, Mexico City. *Lecture Notes in Computer Science* Berlin: LNCS, 2007. p. 268-277.

AYRES, L.; FURTADO, V. **OWLPref: Uma representação declarativa de preferências para Web Semântica**. In: Congresso da Sociedade Brasileira de Computação, 27., 2007, Belém, Pará.

BAIDA, Z.; GORDIJN, J.; OMELAYENKO, B. **A shared terminology for online service provisioning**. In: International Conference on Electronic Commerce, 6., 2004, Delft. *ACM International Conference Proceeding Series 60* Delft: ACM, 2004. p. 1-10.

BEPPLER, F. D. **Um modelo para recuperação e busca de informação baseado em ontologia e no círculo hermenêutico**. 2008. 122f. Tese de doutorado (Doutorado em Engenharia do Conhecimento) - Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento da Universidade Federal de Santa Catarina, Florianópolis, 2008.

BERNERS-LEE, T. **Relational databases on the Semantic Web**, agosto 2009. Disponível em: <<http://www.w3.org/DesignIssues/RDB-RDF.html>>. Acesso em: 12 jan. 2010.

BERNERS-LEE, T.; HENDLER, J.; LASSILA, O. **The Semantic Web**. *Scientific American*, New York, v.284, n. 5, p. 34-43, maio, 2001.

BREIVOLD, H.; LARSSON, M. **Component-based and service-oriented software engineering: key concepts and principles**. In: Conference on Software Engineering and Advanced Applications, 33., 2007, Lübeck. *Institute of Electrical and Electronics Engineers* Lübeck: IEEE, 2007. p. 13-20.

BRGM – BUREAU DE RECHERCHES GÉOLOGIQUES ET MINIÈRES – FRENCH GEOLOGICAL SURVEY. **Deliverable 1.1 Use Case Definition and I&T Requirements**, 2007.

CÂMARA, G.; DAVIS JR., C.; MONTEIRO, A. M. V. **Introdução à ciência da geoinformação**, 2001.

CLANCEY, W. J. **Heuristic Classification**. *Artificial Intelligence*, v. 27, n. 3, p. 289-350, 1985.

CONCAR - COMISSÃO NACIONAL DE CARTOGRAFIA. **Perfil de Metadados Geospaciais do Brasil (Perfil MGB)**, 2009 Disponível em: <http://www.concar.ibge.gov.br/arquivo/Perfil_MGB_Final_v1_homologado.pdf>. Acesso em: 12 de jan. 2010.

CRAGLIA, M. et al. **Next-Generation Digital Earth: A position paper from the Vespucci Initiative for the advancement of geographic information science**. International Journal of Spatial Data Infrastructures Research, v. 3, p. 146-167, 2008.

CRUZ, I.; ANTONELLI, P. F.; STROE, C. **Efficient selection of mappings and automatic quality-driven combination of matching methods**. In: International Semantic Web Conference, 8., 2009, Washington DC., USA.

DALTIO, J. **Aondê: Um Serviço Web de Ontologias para Interoperabilidade em Sistemas de Biodiversidade**. 2007. 107f. Dissertação (Mestrado em Ciências da Computação) – Programa de Pós-Graduação em Ciências da Computação da UNICAMP, Universidade de Campinas, Campinas, 2007.

DARWIN CORE. **Quick Reference Guide**, dezembro, 2002 Disponível em: <<http://rs.tdwg.org/dwc/terms/index.htm>>. Acesso em: 12 jan. 2010.

DAVIS JR., C.; FONSECA, F.; CÂMARA, G. **Infraestruturas de Dados Espaciais na Integração entre Ciência e Comunidades para Promover a Sustentabilidade Ambiental**. In: Workshop de Computação aplicada ao Meio Ambiente, 1., 2009, Bento Gonçalves, RS, Brasil. Sociedade Brasileira de Computação 2009.

DBPEDIA. **Structured Information from Wikipedia**, 2010, Disponível em: <<http://dbpedia.org/About>>. Acesso em: 5 de jun. 2010.

DOMINGUE, J.; ROMAN, D.; STOLLBERG, M. **Web Service Modeling Ontology (WSMO)**. In: W3C Workshop on Frameworks for Semantic Web Services, 2005, Innsbruck, Austria.

EGENHOFER, M. **Toward the semantic geospatial web**. In: International Symposium on Advances in geographic information systems, 10., 2002, McLean. Association for Computing Machinery New York: ACM, 2002. p. 1-4.

FARQUHAR, A.; FIKES, R.; RICE, J. **The Ontolingua Server: a tool for collaborative ontology construction**. International Journal of Human-Computer Studies, v. 46, n. 6, p 707-727, Academic Press, jun. 1997.

FENSEL, D.; BUSSLER, C. **The web service modeling framework WSMF**. Electronic Commerce Research Applications, v. 1, n. 2, p. 113-137, 2002.

FENSEL, D.; KELLER, U.; LAUSEN, H.; POLLERES, A.; TOMA, I. **WWW or What is Wrong with Web Service discovery**. In: Workshop on frameworks for semantics in web services, 2005, Innsbruck, Austria.

FONSECA, F.; EGENHOFER, M.; BORGES, K. **Ontologias e Interoperabilidade Semântica entre SIGs**. In: Workshop Brasileiro em Geoinformática, 2., 2000, São Paulo, SP, Brasil.

FONSECA, F.; EGENHOFER, M. **Ontology-Driven Geographic Information Systems**. In: Symposium on Advances in Geographic Information Systems, 7., 1999, Kansas City, MO. Association for Computing Machinery, 1999, p 14-19.

FONSECA, F.; SHETH, A. **The GeoSpatial Semantic Web**. UCGIS Research Priorities, 2002. Disponível em: <http://www.personal.psu.edu/faculty/f/u/fuf1/Fonseca-Sheth.pdf> Acesso em: 5 de jun. 2010.

GARCIA, E. **Term vector calculations – A fast track tutorial**. 2006. Disponível em: <http://www.miislita.com/information-retrieval-tutorial/term-vector-fast-track.pdf> . Acesso em: 5 de jun. 2010.

GEONAMES. **GeoNames geographical database**, 2010, Disponível em: <<http://www.geonames.org>>. Acesso em: 5 de jun. 2010.

GOMES JR., L. C. **Uma arquitetura para consultas a repositórios de biodiversidade na Web**. 2007. 86f. Dissertação (Mestrado em Ciências da Computação) – Programa de Pós-Graduação em Ciências da Computação da UNICAMP, Universidade de Campinas, Campinas, 2007.

GRACIA, J.; LOPEZ, V.; D'AQUIN, M. **Solving semantic ambiguity to improve semantic web based ontology matching**. In: International Semantic Web Conference, 6., 2007, Bexco, Busan, Korea.

GRCAR, M. **Software Module for Semantic Annotation of a Web Service**, Swing Deliverable 4.5, Jožef Stefan Institute, 2008

GRUBER, T. R. **A Translation Approach to Portable Ontology Specifications**. Knowledge Acquisition, Palo Alto, v. 5, n. 2, p. 199-220, abril, 1993.

GRUBER, T. R. **Ontology**. In: Encyclopedia of Database Systems, Ling Liu e M. Tamer Ozsu

(Eds.) Springer-Verlag, 2009.

HEROLD, M. **WSMX Documentation**. Digital Enterprise Research Institute, 2008, Galway, Ireland.

HESS, G. N. **Towards Effective Geographic Ontology Semantic Similarity Assessment**. 2008. 118f. Ph.D. Tese (Doutorado em Computação) – Programa de Pós-Graduação em Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2008.

HILDEBRAND, M.; VAN OSSENBRUGGEN; J. R.; HADMAN, L. **An Analysis of Search-based User Interaction on the Semantic Web**, Technical Report CWI (INS-E0706), 2007. Disponível em <http://www.cwi.nl/ftp/CWIREports/INS/INS-E0706.pdf> . Acesso em: 12 jan. 2010

IPEA – INSTITUTO DE PESQUISA ECONÔMICA APLICADA. **IPEA Mapas**, 2009 Disponível em: <<http://mapas.ipea.gov.br>> . Acesso em: 12 jan. 2010.

JENA. **Java framework for building Semantic Web applications**, 2005 Disponível em: <<http://jena.sourceforge.net>>. Acesso em: 12 de jan. 2010.

KELLER, U.; LARA, R.; POLELRES, A.; TOMA, I.; KIFER, M.; FENSEL, D. **WSMO Web Service Discovery**. WSMO Working Draft, v. 1, 2004.

KLIEN, E. M. **Semantic Annotation of Geographic Information**. 2008. 152f. PhD Tese (Institute for Geoinformatics) - University of Muenster, Muenster, Alemanha.

KLIEN, E. M.; LUTZ, M. **The Role of Spatial Relations in Automating the Semantic Annotation of Geodata**. In: Conference of Spatial Information Theory, 3., 2005, Ellicottville, New York. Lecture Notes in Computer Science, vol. 3693, 2005, p 133-148.

KLUSCH, M. **Semantic Service Coordination**. In: CASCOM – Intelligent Service Coordination in the Semantic Web. , Birkhaeuser Verlag, Springer, 2008. p 59-104.

KLUSCH, M.; FRIES, B.; SYCARA, K. **Automated Semantic Web Service Discovery with OWLS-MX**. In: International Conference on Autonomous Agents and Multi-Agent Systems, 5., 2006, Hakodate, Japão.

KLUSCH, M.; KAUFER, F. **WSMO-MX: A hybrid Semantic Web service matchmaker**. Web Intelligence and Agent Systems, v. 7, n. 1, p. 23-42, 2009.

KIRYAKOV, A.; POPOV, B.; TERZIEV, I.; MANOV, D.; OGNJANOFF, D. **Semantic annotation, indexing, and retrieval**. Web Semantics: Science, Services and Agents on the World Wide Web, v. 2, n. 1, p 49-79, 2004.

LOPEZ, V.; UREN, V.; MOTTA, E.; PASIN, M. **AquaLog: An ontology-driven question answering system for organizational semantic intranets**. Semantics: Science, Services and Agents on the World Wide Web, v. 5, n. 2, p. 72-105, 2007.

MACÁRIO, C. G.; MEDEIROS, C. B. **The Geospatial Semantic Web – are GIS Catalogs prepared for this?** In: Web Information Systems and Technologies, 5., 2009, Lisboa, Portugal. WEBIST 2009: 335-340.

MACÁRIO, C. **Anotação semântica de dados geoespaciais**. 2009. 126f. Tese – Instituto de Computação, Universidade Estadual de Campinas, Campinas, 2009.

MACHADO, J. C. **Um estudo sobre o desenvolvimento orientado a serviços**. 2004. 89f. Dissertação (Mestrado em Informática) – Programa de Pós-Graduação em Informática da PUC-Rio, Pontifícia Universidade Católica, Rio de Janeiro, 2004.

MAUÉ, P.; SCHADE, S.; DUCHESNE, P. OGC Discussion Paper 08-167r1: **Semantic Annotations in OGC Standards**. Technical report, OGC, 2009.

MCGUINNESS, D. L.; SHVAIKO, P.; GIUNCHIGLIA, F.; SILVA, P. P. **Towards Explaining Semantic Matching**. In: International Workshop on Description Logics, 17., 2004, Whistler, BC, Canada. CEUR Workshop Proceedings 147, 2004.

NOY, N. F.; MUSEN, M. A. **The PROMPT Suite: Interactive Tools For Ontology Merging And Mapping**. International Journal of Human-Computer Studies, v. 59, p. 983-1024, 2004.

OBITKO, M. **Introduction to Ontologies and Semantic Web**, 2007. Disponível em: <http://www.obitko.com/tutorials/ontologies-semantic-web/>. Acesso em: 5 de jun. 2010.

OPEN GEOSPATIAL CONSORTIUM INC. **OGC 08-062r4: OGC Reference Model**. Massachusetts, 2008. Disponível em: <http://www.opengeospatial.org/standards>. Acesso em: 5 de jun. 2010.

OPENSTREETMAPS. **Free editable map of the whole world**, 2010. Disponível em: <http://www.openstreetmap.org/>. Acesso em: 5 de jun. 2010.

OWL-S **Semantic Markup for Web Services**, 2004. Disponível em: <http://www.w3.org/Submission/OWL-S/>. Acesso em: 5 de jun. 2010.

PAOLUCCI, M.; KAWAMURA, T.; PAYNE, T.; SYCARA, K. **Semantic Matching of Web Service Capabilities**. In: International Semantic Web Conference, 2002, Lecture Notes in Computer Science 2342, Springer 2002, p. 333-347.

PREIST, C. **A conceptual architecture for semantic web services**. In: Third International Semantic Web Conference, Hiroshima, Japan. Lecture Notes in Computer Science 3298, Springer 2004, p. 395-409 .

RDF WORKING GROUP. **Resource Description Framework: RDF**, fevereiro, 2004 Disponível em: <<http://www.w3.org/2001/sw/wiki/RDF>> . Acesso em: 12 de jan. 2010.

REEVE, L.; HAN, H. **Survey of semantic annotation platforms**. In: Symposium on Applied Computing, Web Technologies and Applications track, 20., 2005, Santa Fé, New Mexico. Association for Computing Machinery : 2005.

ROMAN, D.; KLIEN, E.; SKOGAN, D. **SWING - A Semantic Web Services Framework for the Geospatial Domain**. In: Terra Cognita - Directions to the Geospatial Semantic Web, 2., 2006, Athens, GA, USA. Disponível em: http://ifgi.uni-muenster.de/~klien/publications/SWING-PositionPaper_TerraCognita2006_final.pdf, Acesso em: 12 de jan. 2010

SABOU, M.; D'AQUIN, M.; MOTTA, E. **Using the Semantic Web as background knowledge for ontology mapping**. In: Workshop: International Workshop on Ontology Matching (OM-2006) at International Semantic Web Conference (ISWC'06), 2006, Athens, Georgia, USA.

SBODIO, M. L.; MOULIN, C. **SPARQL as an expression language for OWL-S**. In: EUROPEAN SEMANTIC WEB CONFERENCE, n. 4, 2007, Innsbruck. Proceedings of OWL-S: Experiences and Directions, Berlin: Springer, 2007.

SCHARFFE, F. **Ontology mapping specification language**. In: Knowledge Web Ph.D. Symposium ESWC 2007, 2., 2007, Innsbruck, Austria.

SDI COOKBOOK **The SDI cookbook**. Global Spatial Data Infrastructure, 2004. Disponível em: <http://www.gsdi.org/docs2004/Cookbook/cookbookV2.0.pdf>, Acesso em: 03 de jul. 2010.

STOUTENBURG, S. K. **Acquiring advanced properties in ontology mapping**. In: PhD Workshop on Information and knowledge management, 2., 2008, Napa Valley, California, USA. Association for Computing Machinery, 2008, p. 9-16.

SESAME. **Open source framework for storage, inferencing and querying of RDF data**, 2008 Disponível em: <<http://www.openrdf.org/>>. Acesso em: 12 de jan. de 2010.

SCHADE, S. et al. **Deliverable 3.3: Repository with ontologies. SWING Project Deliverable**, 2008. Disponível em: <<http://ifgi.uni-muenster.de/~schades/>>. Acesso em: 25 jan. 2010.

STARLAB. **Ontology Server for Semantic Web**, 2009 Disponível em: <<http://www.starlab.vub.ac.be/research/dogma/OntologyServer.htm>>. Acesso em: 2 jun. 2010.

STOILLOS, G.; STAMOU, G.; KOLLIAS, S. **A string metric for ontology alignment**. In: 4th International Semantic Web Conference, ISWC 2005, Galway, Ireland, November 6-10, 2005, Lecture Notes in Computer Science 3729, Springer, p. 624-637.

SWING CONCEPT REPOSITORY, 2010 Disponível em: <<http://swing.uni-muenster.de/core/>>. Acesso em: 2 de jun. 2010.

SWOOGLE. **Semantic Web Search**, 2007 Disponível em: <<http://swoogle.umbc.edu/>>. Acesso em: 12 jan. 2010.

VIEGAS, R.; SOARES, V. **Querying a Geographic Database using an Ontology-Based Methodology**. In: Advances in Geoinformatics. Springer, 2007. p. 165-182

W3C RDF PRIMER. **RDF Primer**, fevereiro, 2004. Disponível em: <<http://www.w3.org/TR/2004/REC-rdf-primer-20040210>>. Acesso em: 10 jan. 2010.

W3C SEMANTIC WEB ACTIVITY. **The Semantic Web Activity**, 2010 Disponível em: <<http://www.w3.org/2001/sw/>>. Acesso em: 15 abr. 2010.

APÊNDICE A

Código de implementação do shell script utilizado para realizar a carga dos shapefiles no banco de dados PostgreSQL + PostGIS

É apresentado o código fonte do shell script implementado para auxiliar na transformação dos arquivos de dados geográficos do formato shapefile para o banco de dados PostgreSQL, acrescido de sua extensão espacial PostGIS.

```
#!/bin/bash

//Setando configurações de diretórios
// [TODO] – Alterar o script para que tais configurações sejam setadas como
// parâmetro durante a chamada ao script por linha de comando.
POSTGRE=/Applications/Caiena/Postgres/install;
POSTGIS=/Applications/Caiena/Postgres/source/postgresql-8.3.6/contrib/postgis-
1.3.5
DATABASE="isl";

ANP=/Users/marques/Paulina/SHAPEFILES;
SQLDATA=/Users/marques/Paulina/SQL;

echo

echo "[INFO] - Convertendo arquivos Shapefile....";
echo "[INFO] - Este procedimento pode demorar alguns minutos....";

//Apagando e criando um novo Database espacial
$POSTGRE/bin/dropdb $DATABASE;
$POSTGRE/bin/createdb $DATABASE;
$POSTGRE/bin/createlang plpgsql $DATABASE;

$POSTGRE/bin/psql -d $DATABASE -f $POSTGIS/lwpostgis.sql;
$POSTGRE/bin/psql -d $DATABASE -f $POSTGIS/spatial_ref_sys.sql;

cd $ANP;

//Gerando os arquivos SQL com base nos arquivos shapefiles. Isso é feito
// utilizando o shp2pgsql.
for file in *.shp; do
    echo
    echo

    echo "[INFO] Convertendo $file ...";
    table=`echo $file | cut -f1 -d.`;

    # $POSTGRE/bin/shp2pgsql -s 4326 -W latin1 $file $table | $POSTGRE/bin/psql
    -d
    $DATABASE;

    $POSTGRE/bin/shp2pgsql -s 4326 -p -W latin1 $file $table >
    /Users/marques/Mestrado/SQL/DDL/$table.sql
    $POSTGRE/bin/shp2pgsql -s 4326 -a -W latin1 $file $table >
    /Users/marques/Mestrado/SQL/DML/$table.sql
done
```

APÊNDICE B

Código de implementação do shell script utilizado para criar os DataStores no GeoServer

É apresentado o código fonte do shell script implementado para auxiliar na criação dos DataStores no GeoServer.

```
#!/bin/bash

//Setando o diretório onde se encontra o arquivo de configuração do DataStore
// [TODO] – Permitir que as configurações sejam passadas como como parâmetros
// durante a invocação do script.
CONFDIR=/Users/marques/Mestrado;

//Setando workspace onde será criado o DataStore.
WORKSPACE=http://localhost:8080/geoserver/rest/workspaces/anp/datastores

//Enviando requisição HTTP ao GeoServer para a criação do DataStore
curl -u admin:geoserver -XPOST -T $CONFDIR/dataStoreConf.xml -H 'Content-type:
text/xml' $WORKSPACE
```

APÊNDICE C

Código de implementação do shell script utilizado para criar as FeatureTypes no GeoServer

É apresentado o código fonte do shell script implementado para auxiliar na criação das FeatureTypes no GeoServer.

```
#!/bin/bash

//Setando o diretório onde se encontra o arquivo de configuração das
FeatureTypes
// [TODO] – Permitir que as configurações sejam passadas como como parâmetros
durante a invocação do script.
CONFDIR=/Users/marques/Mestrado;

//Setando workspace onde será criado a FeatureType.
DATASTORE=http://localhost:8080/geoserver/rest/workspaces/anp/datastores/anp/featuretypes

//Enviando requisição HTTP ao GeoServer para a criação da FeatureType
curl -u admin:geoserver -XPOST -T $CONFDIR/featureTypeConf.xml -H 'Content-type:
text/xml' $WORKSPACE
```

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)