

UERJ

Dissertação de Mestrado em Engenharia de Computação

INTEROPERABILIDADE ENTRE OBJETOS GEOGRÁFICOS HETEROGÊNEOS E DISTRIBUÍDOS: UMA METODOLOGIA DE INTEGRAÇÃO SINTÁTICA E SEMÂNTICA

Autor: Victor Hugo Meirelles de Azevedo

Orientador: Margareth S. P. Meirelles

Co-orientador: Oscar L. M. de Farias

Programa de Pós-Graduação em Engenharia de Computação
Área de Concentração em Geomática.

SETEMBRO - 2007



Faculdade de Engenharia

Faculdade de Engenharia

**INTEROPERABILIDADE ENTRE OBJETOS GEOGRÁFICOS
HETEROGÊNEOS E DISTRIBUÍDOS: UMA METODOLOGIA DE
INTEGRAÇÃO SINTÁTICA E SEMÂNTICA**

Victor Hugo Meirelles de Azevedo

Dissertação submetida ao corpo docente da Faculdade de Engenharia da Universidade do Estado do Rio de Janeiro – UERJ, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Computação.

Orientador: Margareth S. P. Meirelles

Co-orientador: Oscar L. M. de Farias

Programa de Pós-Graduação em Engenharia de Computação - Área de Concentração em Geomática.

Rio de Janeiro
Setembro – 2007

AZEVEDO, VICTOR HUGO MEIRELLES
Interoperabilidade Entre Objetos
Geográficos Heterogêneos e Distribuídos: Uma
metodologia de integração sintática e semântica
[Rio de Janeiro] 2007.

xi, 169 p. 29,7 cm (FEN/UERJ, M.Sc.,
Pós-Graduação em Engenharia de Computação
– Área de Concentração de Geomática, 2007)

Dissertação – Universidade do Estado do
Rio de Janeiro – UERJ

1. Interoperabilidade entre objetos
geográficos heterogêneos
 2. Integração sintática e semântica
 3. Serviços *web* geográficos
 4. Ontologias
 5. Informações espaciais distribuídas
- I. FEN/UERJ II. Título(série)

FOLHA DE JULGAMENTO

Título: Interoperabilidade Entre Objetos Geográficos Heterogêneos e Distribuídos: Uma metodologia de integração sintática e semântica.

Candidato: Victor Hugo Meirelles de Azevedo

Programa de Pós-Graduação em Engenharia de Computação – Área de Concentração em Geomática

Data da defesa: 04 de setembro de 2007

Aprovada por:

Orientador: Margareth Simões Penello Meirelles, DSc, UERJ.

Co-orientador: Oscar Luiz Monteiro de Farias, DSc, UERJ.

Marcelo Tilio Monteiro de Carvalho, DSc, PUC-RJ.

Neide dos Santos, DSc., UERJ.

Às minhas avós Anna Medeiros de Azevedo e Darcília da Silva Meirelles que faleceram
durante a elaboração desta dissertação

AGRADECIMENTOS

Agradeço a todos os amigos, professores, colegas de profissão e instituições que ajudaram a tornar possível a realização deste trabalho. Agradeço em especial:

- Aos meus pais, que além de todo o carinho e amor dispensados, financiaram os meus estudos até que eu pudesse “andar com as próprias pernas”. Seu apoio e conselhos foram, são e sempre serão determinantes na minha caminhada;
- À Ludimila, minha esposa, amiga e companheira, pela dedicação, paciência, apoio, amor, carinho e compreensão nos momentos mais difíceis;
- À Margareth, minha orientadora e amiga, pela dedicação, confiança e pelos conselhos profissionais que me ajudam a crescer;
- Ao pesquisador Rodrigo Peçanha da Embrapa Solos pela ajuda e dedicação na elaboração do material científico desenvolvido nesta dissertação;
- Ao pesquisador e coordenador do projeto de zoneamento agroecológico da cultura do dendê em áreas desmatadas da Amazônia, Antonio Ramalho Filho, pela oportunidade de desenvolver o tema desta dissertação de forma a colaborar com este projeto;
- À UERJ, instituição onde me formei como engenheiro de sistemas e computação, e que agora tenho oportunidade de obter o título de mestre;
- Aos demais professores do curso de pós-graduação em geomática que me deram a oportunidade de aprendizado e crescimento profissional;
- À Embrapa Solos e a FINEP por tornar viável a realização deste trabalho, disponibilizando os recursos materiais e os dados utilizados nesta dissertação;
- Aos demais amigos e familiares pela compreensão e apoio;
- Por fim, agradeço a Deus por ter tido saúde e força para realizar este trabalho.

Resumo da dissertação apresentada à FEN/UERJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.) em Engenharia de Computação.

INTEROPERABILIDADE ENTRE OBJETOS GEOGRÁFICOS HETEROGÊNEOS E DISTRIBUÍDOS: UMA METODOLOGIA DE INTEGRAÇÃO SINTÁTICA E SEMÂNTICA

Victor Hugo Meirelles de Azevedo

Setembro/2007

Orientador: Margareth Simões Penello Meirelles, DSc, UERJ.

Co-orientador: Oscar Luiz Monteiro de Farias, DSc, UERJ.

Programa de Pós-Graduação em Engenharia de Computação – Área de Concentração em Geomática.

A falta de interoperabilidade é um problema encontrado quando se faz necessário utilizar informações disponíveis em diversas instituições geradoras ou detentoras de informações espaciais. Esta dissertação descreve uma proposta metodológica para se obter a interoperabilidade entre objetos geográficos heterogêneos e distribuídos baseada na utilização da especificação do serviço Web Feature Service (WFS) do Open Geospatial Consortium (OGC), como um mecanismo para tornar os formatos de dados uniformes, e a utilização de ontologias do domínio, através da Ontology Web Language (OWL), como esquema sintático e semântico de referência compartilhado para a integração destes objetos. Esta abordagem permite um alto grau de interoperabilidade proporcionando a unificação de estrutura, sintaxe e semântica de dados espaciais heterogêneos em um domínio de interesse.

A metodologia proposta está sendo aplicada na obtenção de informação de solos para o projeto de zoneamento agroecológico da cultura do dendê em áreas desmatadas da Amazônia, coordenado pela Embrapa Solos. Os dados de solos provenientes de fontes geradoras diferentes vêm sendo integrados para utilização na avaliação da aptidão agrícola.

Palavras-chave: Interoperabilidade, objetos geográficos distribuídos, integração sintática e semântica, serviços *web* geográficos, ontologias.

Abstract of dissertation presented to FEN/UERJ as a partial fulfillment of the requirements
for the Master of Science (M.Sc.) degree in Computer Engineering

**INTEROPERABILITY AMONG DISTRIBUTED AND HETEROGENEOUS
GEOGRAPHIC OBJECTS: A METHODOLOGY FOR SYNTACTICAL AND
SEMANTIC INTEGRATION**

Victor Hugo Meirelles de Azevedo

September/2007

Advisor: Margareth Simões Penello Meirelles, DSc, UERJ.

Co-advisor: Oscar Luiz Monteiro de Farias, DSc, UERJ.

Post-graduate Program in Computer Engineering – Field of Geomatics.

The lack of interoperability is an issue found when the use of information, accessible in institutions that generate or detain spatial information, is needed. Therefore, this study describes a methodological proposal in order to achieve interoperability among heterogeneous and distributed geographic objects based on the use of Web Feature Service Specification (WFS) from Open Geospatial Consortium (OGC), as a mechanism to make the data formats standardized, and the use of domain's ontology, through Ontology Web language (OWL), as a syntactic and semantic reference schema shared for the integration of these objects. This approach allows a high level of interoperability which provides the unification of structure, syntax and semantic of heterogeneous spatial data in a domain of interest.

This methodology is being applied in order to obtain soils' information to the agro-ecological zoning project, related to the palm oil cultivation, in deforested Amazon areas, coordinated by the Brazilian National Soil Research Center (Embrapa Solos). Soil data from different sources have been integrated in order to be used for land evaluation.

Keywords: Interoperability, distributed geographic objects, syntactical and semantic integration, geographic web services, ontology, special data infrastructure.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	MOTIVAÇÃO.....	5
1.2	OBJETIVO GERAL.....	5
1.3	OBJETIVOS ESPECÍFICOS	6
1.4	ORGANIZAÇÃO DA DISSERTAÇÃO	7
2	FUNDAMENTOS TEÓRICOS	8
2.1	INTEROPERABILIDADE E PADRONIZAÇÃO	8
2.1.1	<i>O Open Geospatial Consortium (OGC)</i>	9
2.2	SERVIÇOS <i>WEB</i> E AS ESPECIFICAÇÕES DO <i>FRAMEWORK</i> OGC.....	9
2.2.1	<i>Geographic Markup Language (GML)</i>	10
2.2.2	<i>Web Feature Service (WFS)</i>	12
2.2.3	<i>Web Map Service (WMS)</i>	14
2.2.4	<i>OpenGIS Catalog Service (OCS)</i>	14
2.2.5	<i>Web Registry Service (WRS)</i>	15
2.3	METADADOS GEOGRÁFICOS	15
2.4	BASE DE CONHECIMENTO, ONTOLOGIAS E <i>WEB</i> SEMÂNTICA.....	16
2.4.1	<i>Ontology Web Language (OWL)</i>	17
3	METODOLOGIA	21
3.1	DEFINIÇÃO DA ARQUITETURA ORIENTADA A SERVIÇOS DO OGC.....	22
3.2	CRIAÇÃO DOS SERVIÇOS WFS NOS SERVIDORES DE OBJETOS GEOGRÁFICOS	25
3.3	PROCESSO DE ENGENHARIA DO CONHECIMENTO.....	27
3.3.1	<i>Atores</i>	27
3.3.2	<i>Reuso de ontologias</i>	28
3.3.3	<i>Formatos gerados</i>	29
3.4	CONFIGURAÇÃO DO SERVIDOR DE INTEGRAÇÃO	31
3.4.1	<i>Cadastramento dos serviços baseado em metadados</i>	32
3.4.2	<i>Correlação sintática e semântica entre os esquemas conceituais</i>	33
3.5	PUBLICAÇÃO DOS OBJETOS GEOGRÁFICOS INTEGRADOS	39
4	ESTUDO DE CASO.....	43
4.1	FONTES DE DADOS	44
4.1.1	<i>A fonte de dados de solos do SIPAM</i>	44
4.1.2	<i>A fonte de dados de solos do IBGE</i>	46

4.2	APLICAÇÃO DA METODOLOGIA	50
4.2.1	<i>Definição da arquitetura utilizada</i>	50
4.2.2	<i>Configuração dos servidores WFS</i>	51
4.2.3	<i>Configuração do SOG-SIPAM</i>	52
4.2.4	<i>Configuração do SOG-IBGE</i>	53
4.2.5	<i>O Processo de Engenharia do Conhecimento – Criação da Ontologia no domínio de aptidão agrícola dos solos para cultivo de dendê</i>	54
4.2.6	<i>Metodologia de Engenharia do Conhecimento</i>	56
4.2.7	<i>Ontologia de Atributos do Solo para Aptidão Agrícola</i>	57
4.2.8	<i>O Servidor de Integração</i>	58
4.2.9	<i>Publicação das Informações Integradas de Atributos de Solos</i>	64
4.3	RESULTADOS	66
4.4	UTILIZAÇÃO NO PROJETO ZONDENDE	69
4.5	CONSIDERAÇÕES FINAIS	72
5	CONCLUSÕES	74
6	TRABALHOS FUTUROS.....	76
APÊNDICE A - CONFIGURAÇÃO DO SERVIDOR GEOSERVER (29).....		77
A.1	CONFIGURAÇÃO DO SERVIÇO WFS.....	77
A.2	CONFIGURAÇÃO DAS FONTES DE DADOS.....	79
A.3	CRIAÇÃO DE UM NOVO TIPO DE FEIÇÃO GEOGRÁFICA (WFS FEATURETYPE).....	82
A.4	VERIFICANDO O RESULTADO.....	83
APÊNDICE B - DOCUMENTAÇÃO DA ONTOLOGIA DE ATRIBUTOS DO SOLO PARA APTIDÃO AGRÍCOLA		90
B.1	UNIDADE TAXONÔMICA	90
B.2	UNIDADE DE MAPEAMENTO	92
B.3	SATURAÇÃO DE BASES.....	93
B.4	CLASSE DE RELEVO.....	94
B.5	CLASSE DE DRENAGEM.....	95
B.6	CLASSE DE PEDREGOSIDADE	96
B.7	CLASSE DE ROCHOSIDADE.....	97
B.8	CLASSE TEXTURAL.....	98
B.9	ATRIBUTOS DIAGNÓSTICOS.....	99
B.10	CLASSE DE SOLO	100

B.11 REGRAS SWRL	103
APÊNDICE C - DOCUMENTAÇÃO DA BIBLIOTECA DE INTEROPERABILIDADE ENTRE OBJETOS GEOGRÁFICOS BASEADA EM ONTOLOGIAS (OPENOGI)	105
C.1 PACOTE DE TRATAMENTO DE EXCEÇÃO	105
C.2 CLASSE DATASOURCEEXCEPTION	106
C.3 CLASSE OBJECTNOTVALIDEXCEPTION	107
C.4 CLASSE XMLOBJECTNOTFOUNDEXCEPTION	108
C.5 CLASSE XMLSTOREEXCEPTION	109
C.6 PACOTE MODEL	110
C.7 CLASSE DATASOURCE	111
C.8 CLASSE DOMAIN	115
C.9 CLASSE GEOGRAPHICINFORMATIONLAYER.....	119
C.10 CLASSE METADATA	122
C.11 CLASSE SCHEMACORRELATION	126
C.12 CLASSE SCHEMAELEMENT.....	129
C.13 CLASSE WFSDATASOURCE.....	132
C.14 PACOTE SERVICE.....	134
C.15 INTERFACE DATASOURCESERVICE	135
C.16 CLASSE CONFIGURATIONSERVICE	137
C.17 CLASSE EXPORTSERVICE.....	142
C.18 CLASSE ONTOLOGYSERVICE	144
C.19 CLASSE WFSSERVICE	149
C.20 PACOTE UNITTEST	151
C.21 CLASSE SIMILARITYSTRINGTEST.....	151
REFERÊNCIAS	154

ÍNDICE DE FIGURAS

FIGURA 2.1 <i>FRAMEWORK</i> ARQUITETURAL DOS SERVIÇOS <i>WEB</i> DO OGC (8).....	10
FIGURA 2.2 DEFINIÇÃO DO ESQUEMA GML (7)	11
FIGURA 2.3 HIERARQUIA DE CLASSES DO MODELO GML (7).....	12
FIGURA 2.4 OPERAÇÕES DO SERVIÇO WFS (25)	13
FIGURA 2.5 MODELO DE METADADOS ISO 19115 (19).....	16
FIGURA 3.1 DIAGRAMA DE ATIVIDADES DA METODOLOGIA DE INTEGRAÇÃO DE OBJETOS GEOGRÁFICOS.....	22
FIGURA 3.2 DIAGRAMA DA ARQUITETURA COM APENAS UM SIGEO PARA TODOS OS DOMÍNIOS.....	23
FIGURA 3.3 DIAGRAMA DA ARQUITETURA COM UM SIGEO PARA CADA DOMÍNIO	24
FIGURA 3.4 DIAGRAMA DE IMPLANTAÇÃO DO SERVIÇO WFS NOS SOGS.....	25
FIGURA 3.5 DIAGRAMA DE IMPLANTAÇÃO DO FUNCIONAMENTO DO SERVIÇO WFS	26
FIGURA 3.6 DOCUMENTAÇÃO DA ONTOLOGIA NO DOMÍNIO DA HIDROGRAFIA GERADA PELO <i>PROTÉGÉ</i>	30
FIGURA 3.7 CORRELAÇÃO SINTÁTICA E SEMÂNTICA	39
FIGURA 3.8 PROPOSTA DE INTEROPERABILIDADE ENTRE OBJETOS GEOGRÁFICOS HETEROGÊNEOS DISTRIBUÍDOS	42
FIGURA 4.1 MAPA DE SOLOS DO SIPAM APRESENTADA NO ARCMAP (ESRI).....	44
FIGURA 4.2 TABELA DE ATRIBUTOS DO MAPA DE SOLOS DO SIPAM	45
FIGURA 4.3 RELACIONAMENTO ENTRE AS TABELAS ESPACIAIS E AS TABELAS DO MODELO RELACIONAL.....	46
FIGURA 4.4 MAPA ÍNDICE DAS CARTAS DO IBGE (1:1.000.000) (? , FUI EU QUE MONTEI O MAPA A PARTIR DAS INFORMAÇÕES DE UM SIG-WEB DO IBGE)	47
FIGURA 4.5 MAPA DE SOLOS DO IBGE.....	49
FIGURA 4.6 ARQUITETURA DO ESTUDO DE CASO	51
FIGURA 4.7 DIAGRAMA DE COMPONENTES DA BASE DE CONHECIMENTO	55
FIGURA 4.8 ONTOLOGIA DE ATRIBUTOS DO SOLO PARA APTIDÃO AGRÍCOLA NO EDITOR <i>PROTÉGÉ</i> OWL.....	57
FIGURA 4.9 RELACIONAMENTOS DA UNIDADE DE MAPEAMENTO DE SOLO NA ONTOLOGIA.....	58
FIGURA 4.10 PROTÓTIPO DO SERVIDOR DE INTEGRAÇÃO	60

FIGURA 4.11 CADASTRAMENTO DE DOMÍNIOS	61
FIGURA 4.12 CADASTRAMENTO DAS FONTES DE DADOS GEOGRÁFICOS	62
FIGURA 4.13 TABELA DE CORRELAÇÃO SINTÁTICA E SEMÂNTICA	63
FIGURA 4.14 FUNCIONALIDADE DE GERAÇÃO DE MAPA	65
FIGURA 4.15 FUNCIONALIDADE DE GERAÇÃO DE MAPA	66
FIGURA 4.16 ÁREA DE ESTUDO	67
FIGURA 4.17 CAMADA DE INFORMAÇÃO DE SOLOS FORMADA POR FEIÇÕES GEOGRÁFICAS INTEGRADAS.....	67
FIGURA 4.18 ATRIBUTOS DE UMA UNIDADE DE MAPEAMENTO DA CAMADA DE INFORMAÇÃO INTEGRADA.....	68
FIGURA 4.19 SELEÇÃO DA UNIDADE DE MAPEAMENTO DO SIPAM NO MAPA INTEGRADO.....	69
FIGURA 4.20 EXPORTAÇÃO DE DADOS PARA A ONTOLOGIA OWL	70
FIGURA 4.21 INDIVÍDUOS (UNIDADES TAXONÔMICAS) DA ONTOLOGIA.....	71
FIGURA 4.22 CONJUNTO DE REGRAS E AS INFERÊNCIAS REALIZADAS PELO <i>REASONER</i>	71
FIGURA 4.23 REGRA SWRL QUE DEFINE O PERCENTUAL DA UNIDADE TAXONÔMICA.....	72
FIGURA A.1 CONFIGURAÇÃO DAS INFORMAÇÕES SOBRE O SERVIDOR WFS	77
FIGURA A.2 METADADOS DO SERVIDOR.....	78
FIGURA A.3 CONFIGURAÇÃO DA DESCRIÇÃO DO SERVIÇO WFS.....	78
FIGURA A.4 CRIAÇÃO DA FONTE DE DADOS NO SERVIDOR WFS <i>GEOSERVER</i> ..	79
FIGURA A.5 CONFIGURAÇÃO DA FONTE DE DADOS SOLOS_SIPAM NO <i>GEOSERVER</i>	80
FIGURA A.6 CRIAÇÃO DA FONTE DE DADOS SOLOS_IBGE_SB18 NO SERVIDOR SOG-IBGE	81
FIGURA A.7 CONFIGURAÇÃO DA FONTE DE DADOS SOLOS_IBGE_SB18 NO SERVIDOR SOG-IBGE.	81
FIGURA A.8 CRIAÇÃO DE UM TIPO DE FEIÇÃO GEOGRÁFICA NO <i>GEOSERVER</i> ..	82
FIGURA A.9 CONFIGURAÇÃO DO TIPO DE FEIÇÃO NO <i>GEOSERVER</i> – ESTILO E SISTEMA DE REFERÊNCIA.....	83
FIGURA A.10 CONFIGURAÇÃO DO TIPO DE FEIÇÃO NO <i>GEOSERVER</i> – DESCRIÇÃO E ESQUEMA DE DADOS.....	84

FIGURA A.11 RETORNO DA OPERAÇÃO <i>GETCAPABILITIES</i> NO SERVIDOR SOG-SIPAM.....	85
FIGURA A.12 RETORNO DA OPERAÇÃO <i>DESCRIBEFEATURETYPE</i> SOLICITADA AO SERVIDOR SOG-SIPAM.....	86
FIGURA A.13 RETORNO DA OPERAÇÃO <i>DESCRIBEFEATURETYPE</i> NO SERVIDOR SOG-IBGE	87
FIGURA A.14 RETORNO DA OPERAÇÃO <i>GETFEATURE</i> NO SERVIDOR SOG-SIPAM	88
FIGURA A.15 FEIÇÃO GEOGRÁFICA REPRESENTADA EM GML (RETORNO DA FUNÇÃO <i>GETFEATURE</i>)	89
FIGURA B.1 UNIDADE TAXÔNOMICA	92
FIGURA B.2 UNIDADE DE MAPEAMENTO	93
FIGURA B.3 SATURAÇÃO DE BASES	94
FIGURA B.4 CLASSE DE RELEVO.....	95
FIGURA B.5 CLASSE DE DRENAGEM.....	96
FIGURA B.6 CLASSE DE PEDREGOSIDADE	97
FIGURA B.7 CLASSE DE ROCHOSIDADE.....	98
FIGURA B.8 CLASSE TEXTURAL.....	99
FIGURA B.9 ATRIBUTOS DIAGNÓSTICOS.....	100
FIGURA B.10 CLASSE DE SOLO	103
FIGURA B.11 REGRA DE PERCENTUAL DA UNIDADE TAXONÔMICA NA UNIDADE DE MAPEAMENTO	104
FIGURA B.12 CONJUNTO DE REGRAS E VALORES INFERIDOS.....	104

ÍNDICE DE TABELAS

TABELA 2.1 OPERADORES LÓGICOS DE RESTRIÇÃO EM OWL	19
TABELA 3.1 CORRELAÇÃO SINTÁTICO-SEMÂNTICA ENTRE OS ESQUEMAS CONCEITUAIS DO SERVIDOR SOG2 E O ESQUEMA DE REFERÊNCIA	37
TABELA 4.1 ESQUEMA CONCEITUAL DA FONTE DE DADOS DO IBGE	48
TABELA 4.2 EXEMPLOS DE HETEROGENEIDADES ENCONTRADAS NAS BASES DE DADOS DE SOLOS	50
TABELA 4.3 RELAÇÃO DE CAMPOS DA VISÃO UTILIZADA PARA PUBLICAR OS DADOS DE SOLOS DO SIPAM	52
TABELA 4.4 OPERAÇÕES EXECUTADAS NO SOG-IBGE	53
TABELA 4.5 COMPARAÇÃO ENTRE ALGORITMOS DE MEDIDA DE SIMILARIDADE	60
TABELA 4.6 CORRESPONDÊNCIA ENTRE OS ITENS DE METADADOS UTILIZADOS E OS ITENS DA ISO 19115.	62

1 Introdução

A integração de objetos geográficos, armazenados em fontes de dados distintas e com estrutura sintática e semântica heterogênea, tem sido alvo dos pesquisadores que trabalham com sistemas computacionais em ambiente distribuído de Geoprocessamento nos últimos anos. Este fato ocorre devido à crescente necessidade de troca das informações processadas pelas instituições geradoras de dados geográficos. Para alcançar este objetivo, muitos sistemas atualmente mantêm requisitos de interoperabilidade, ou seja, procuram se comunicar uns com outros trocando informações através do uso de padrões abertos.

Muitas iniciativas têm sido realizadas para alcançar esta interoperabilidade. Segundo Fonseca e Egenhofer (1), as primeiras tentativas para tornar os Sistemas de Informações Geográficas (SIG) interoperáveis foram através da tradução direta de formatos de dados entre fabricantes de software. Atualmente, profissionais e instituições da área de Geoprocessamento formam o Open Geospatial Consortium (OGC) (2), tendo como objetivo definir um conjunto de especificações padrão para interoperabilidade para SIGs. A iniciativa GeoBR (3) propõe um esquema de dados único, com elementos pré-definidos, incluindo modelo de dados, metadados, projeções, geometrias e atributos, de tal maneira que os dados possam ser acessados por uma única interface de programação.

As cooperativas de dados geográficos, como descrito em (4), são uma solução em ascensão para tratar do gerenciamento de informações geográficas em grande volume, de forma a permitir o seu uso cooperativo por agências governamentais ou privadas. Neste sentido, a integração entre os dados espaciais heterogêneos gerenciados por instituições que necessitam realizar trocas de informação entre si, é um objeto de estudo de grande importância para a evolução científica e tecnológica na área da geoinformática.

Em projetos envolvendo um ambiente heterogêneo, a obtenção dos dados representa o maior custo, cerca de 60% a 80% do valor total da implementação dos SIGs (3). Sendo assim, o alto custo associado a fase de coleta e produção de dados geográficos faz da interoperabilidade entre informações espaciais já produzidas por diversas instituições uma necessidade. Neste cenário, a automatização da integração em um ambiente distribuído torna-se uma ferramenta útil cujo principal objetivo é incentivar a cooperação entre as instituições produtoras e detentoras de informações espaciais, principalmente no Brasil por ser um país de dimensão continentais.

Alcançar a completa interoperabilidade de objetos geográficos de tal maneira que os mesmos sejam interpretados segundo uma mesma ótica não é tarefa simples (3). Segundo

Casanova et. al.(5), a integração completa entre fontes de dados geográficos que apresentam heterogeneidade, tanto de estrutura e forma dos dados, quanto de significado e interpretação deve resolver a incompatibilidade entre estas fontes em três níveis: formato e estrutura; sintático; e semântico.

Para que um interpretador, seja um indivíduo ou um programa computacional, possa decodificar os dados de maneira uniforme, estes precisam estar em um mesmo formato de dados. Se objetos geográficos, como unidades administrativas (estados da federação, por exemplo), estiverem em um formato não aberto, eles são interpretados por um software de geoprocessamento específico, que saiba interpretar este formato, para serem apresentados em um formato visual de mapa. No entanto, se estes mesmos objetos estiverem em um outro formato, ou, até mesmo, em um banco de dados espacial, eles não poderão ser interpretados por este mesmo software sem que antes seja feita a conversão de formatos.

Não é difícil encontrar informações geográficas já processadas e armazenadas por instituições geradoras, porém, na maioria das vezes, cada instituição fornece estes dados em um formato particular. Ao tentar utilizar dados geográficos de mais de uma instituição, o usuário destes dados, muito provavelmente, encontrará uma situação de incompatibilidade de formatos. Desta maneira, para que os objetos geográficos armazenados em formatos diferentes possam interagir entre si, o aspecto da integração de formatos e estrutura dos dados geográficos heterogêneos deve ser levado em consideração.

A equivalência sintática entre os elementos representados em um mesmo domínio de interesse é um outro aspecto importante na tentativa de integrar informações em um ambiente heterogêneo. Quando um mesmo elemento de domínio é armazenado com uma denominação distinta em locais diferentes, a tarefa de interpretar estes dados se torna difícil de ser realizada. Em uma situação hipotética que, no domínio da hidrografia, um rio seja chamado de “rio” pela Agência Nacional de Águas (ANA), de “hidrovia” pela Diretoria de Hidrografia e Navegação da Marinha (DHN), e de “corpo d’água” pelo Instituto Brasileiro de Geografia e Estatística (IBGE), como fazer para que um software interpretador, ou um usuário leigo, entenda que se trata do mesmo objeto geográfico? O que fazer para que todas estas representações possuam as mesmas características e possam ser interpretadas da mesma maneira? Para isso, é preciso que haja uma uniformização de conceitos e nomenclaturas quando se deseja integrar os dados destas e de outras instituições.

O mais sutil dos níveis de compatibilidade das informações geográficas, talvez seja a equivalência semântica entre os dados. Algumas vezes este conhecimento está representado de maneira organizada, com um modelo de dados bem definido em que os objetos geográficos

são representados pela sua estrutura, pelo relacionamento que existe entre eles e pelas regras de consistência. Em outras, os objetos estão representados de maneira mais simples, em forma de tabelas que não possuem uma devida estruturação. Portanto, quando duas ou mais instituições que desejam trocar informações representam o seu conhecimento acerca do domínio de maneira diferente, ações devem ser tomadas no sentido de garantir a interoperabilidade. Desta forma, um meio unificado de representar o conhecimento sobre um domínio específico se faz necessário para viabilizar a troca de informações.

Muitos esforços no sentido de fornecer mecanismos computacionais de padronização de formatos e uniformização sintática e semântica de objetos geográficos têm sido estudados e fornecidos. A linguagem *Extensive Markup Language* (XML) (6) vem se destacando cada vez mais como uma ferramenta importante para troca de dados entre instituições. Atualmente é inquestionável o uso de XML como padrão para troca de dados (3).

Sendo assim, o OGC fornece um conjunto de especificações para padronizar o processo de interoperabilidade entre diferentes formatos de dados baseadas na tecnologia XML. A *Geographic Markup Language* (GML) (7) pode ser considerada a principal delas, por ser utilizada em muitas outras especificações. Ela foi concebida com o objetivo de representar as informações geográficas, incluindo tanto as informações espaciais quanto as não espaciais dos objetos geográficos. Segundo Davis Jr. et. al. (8), o objetivo da GML é oferecer um conjunto de regras de forma que o usuário possa definir sua própria linguagem para descrever os objetos geográficos que se deseja manipular. Para tanto, a GML é baseada em esquemas XML (*XML Schema*) (10) que definem os elementos usados em um documento que descreve os dados em forma de *tags*. Os esquemas da GML sozinhos não são adequados para criar uma instância de documento, portanto, estes devem ser estendidos pela criação de esquemas para domínios específicos.

Um outro conjunto de especificações do OGC fornece um *framework* arquitetural que define o escopo, objetivos e comportamento de uma série de serviços *web*, denominado *OpenGIS Services Framework* (8). Este *framework* contém as especificações para um conjunto de serviços *web* com o objetivo de fornecer um mecanismo capaz de garantir a interoperabilidade entre as instituições, utilizando a rede global *internet* como meio de troca de dados. Entretanto, os serviços *web* especificados pelo OGC não seguem as recomendações tecnológicas do *World Wide Web Consortium* (W3C) para definição de serviços *web* (11). Sendo assim, a relação com os serviços *web* fornecidos seguindo as especificações W3C pode ficar comprometida sob o ponto de vista da eficácia de integração.

As especificações do OGC não incluem os aspectos semânticos da interoperabilidade na sua abordagem, nem seguem as recomendações do consórcio W3C para a utilização de *web* semântica (12), como afirma Davis Jr. et. al. (8). Portanto, passa a ser necessário realizar uma abordagem voltada para a representação do conhecimento acerca do domínio que se deseja integrar, atendendo desta maneira a um dos principais requisitos para que a interoperabilidade seja alcançada. Neste sentido, ultimamente, as ontologias vêm sendo amplamente utilizadas como técnica de representação do conhecimento.

Ontologias é um conceito utilizado em inteligência artificial para representação do conhecimento sobre um determinado domínio, que utiliza um vocabulário comum para os membros e as relações entre eles no domínio de interesse. Mesmo sendo uma área de conhecimento antiga, avaliada desde o estudo feito por Aristóteles sobre as categorias e a metafísica (1), nos últimos tempos, o desenvolvimento de ontologias vem migrando dos laboratórios de inteligência artificial para os computadores pessoais dos especialistas de domínio (13). A utilização de ontologia como estratégia de representação do conhecimento sobre um determinado domínio apresenta-se eficaz, já que podem ser especificados de maneira explícita e formal os seus termos bem como o relacionamento entre eles.

O consórcio W3C desenvolve um *framework* para formalizar a idéia da *web* semântica (12). Este *framework* é baseado no *Resource Description Framework* (RDF), que integra uma variedade de aplicações usando *XML* (12). O objetivo principal da *web* semântica é criar uma rede de dados e não uma rede de documentos como é predominante hoje, além de padronizar a forma como esses dados fazem referência aos objetos do mundo real. Neste novo cenário surge a *Web Ontology Language* (OWL) (14), uma linguagem de marcação que estende o vocabulário do RDF, com o objetivo de definir, publicar e compartilhar ontologias na *web*.

Outro aspecto que deve ser levado em consideração ao se realizar a integração de objetos geográficos heterogêneos de várias instituições é o uso de metadados. Os metadados descrevem o conteúdo, condição, histórico, localização e o formato dos dados (3). O objetivo do seu uso é proporcionar um mecanismo de identificação dos dados existentes em cada instituição, bem como a qualidade das informações e a maneira de acesso, restrições e utilização destes dados.

Com o objetivo de fornecer uma solução de interoperabilidade entre dados geográficos heterogêneos considerando os aspectos necessários para a integração, Klien, Lutz e Kuhn (15) propõem uma arquitetura baseada em serviços *web* geográficos e utilizam o sistema BUSTER (<http://www.informatik.uni-bremen.de/agki/www/buster/new>) para integração das informações sobre tempestades em florestas. O sistema BUSTER é um sistema baseado em

ontologias utilizado para busca e integração de informações em ambiente distribuído heterogêneo (15).

1.1 Motivação

O processo desordenado de ocupação das terras amazônicas ao longo das últimas décadas gerou uma demanda por iniciativas de planejamento do uso da região. Aliado a esta necessidade, surge o programa federal de biocombustível para produção de biodiesel a partir de culturas oleaginosas e palmáceas. Neste cenário surge o projeto de zoneamento agroecológico para culturas oleaginosas na região amazônica, com ênfase na região de fronteira, denominado Zon-dende (16), financiado pela FINEP (Financiadora de Estudos e Projetos). O projeto tem como objetivo conhecer o potencial da produção sustentada de culturas oleaginosas, com ênfase para o dendê, nas áreas desmatadas da região amazônica, principalmente na faixa de fronteira com países vizinhos, visando à produção de biodiesel.

O zoneamento agro-ecológico é realizado com informações obtidas a partir de levantamentos pedológicos e climáticos. A região amazônica possui um conjunto de levantamentos de solos armazenados em bancos de dados de instituições como o Instituto Brasileiro de Geografia e Estatística (IBGE), o Sistema de Proteção da Amazônia (SIPAM), a Empresa Brasileira de Pesquisa Agropecuária (EMBRAPA) e outras instituições a nível regional que poderão ser utilizados para a avaliação da aptidão agrícola das terras. Entretanto, estes dados possuem estrutura, sintaxe e semântica diferente e precisam ser integrados para posteriormente serem interpretados a fim de gerar o zoneamento agroecológico para o dendê. As informações de solos levantadas estão armazenadas em um conjunto de bases de dados geográficas em diferentes formatos. A necessidade de troca de informação entre estas bases de dados geográficas é a principal motivação para a elaboração de uma metodologia para realizar a interoperabilidade entre os objetos geográficos heterogêneos contidos nestas fontes de dados.

1.2 Objetivo Geral

Nesta pesquisa, é defendida a hipótese de que os serviços *web* do *framework* OGC aliados ao uso de ontologias no formato OWL formam um arcabouço que permite a interoperabilidade entre fontes de dados espaciais. Sendo assim, o principal objetivo deste trabalho é propor uma metodologia com a finalidade de realizar a interoperabilidade entre

objetos geográficos heterogêneos e distribuídos, de tal maneira que os aspectos sintáticos, semânticos e de formato de dados sejam considerados.

A metodologia proposta nesta dissertação sugere a utilização das especificações da GML e dos serviços *web* do OGC como mecanismo tecnológico para a integração sintática e de estrutura entre objetos geográficos heterogêneos. Além disto, as ontologias e a descrição do domínio na forma de linguagem OWL definem os aspectos semânticos e sintáticos da integração fornecendo o braço tecnológico da representação do conhecimento acerca do domínio. O uso de ontologias em linguagem OWL permite ainda que no processo de integração sejam realizadas inferências sobre o domínio, possibilitando a obtenção de informações extras a partir dos dados integrados.

Assim, pretende-se automatizar a interoperabilidade entre objetos geográficos disponíveis nas diversas instituições, com o objetivo de reduzir os custos desta integração e tornar mais ágil o processo de tomada de decisão e análise. As heterogeneidades sintática, semântica e de formato e estrutura dos dados espaciais devem ser consideradas para a realização da integração, portanto, a utilização das especificações da GML e dos serviços *web* do OGC, ultimamente em crescente difusão, fornecem o mecanismo tecnológico para a integração sintática e de estrutura entre objetos geográficos heterogêneos. As ontologias e a sua descrição do domínio na forma de linguagem OWL, definem os aspectos semânticos e sintáticos da integração fornecendo o braço tecnológico da representação do conhecimento acerca do domínio. Aliado a estes fatores, a disponibilidade das informações sobre os dados de cada domínio em forma de metadados deve ser levada em consideração para que a metodologia forneça um mecanismo de consulta eficiente.

Desta forma, pretende-se contribuir com o desenvolvimento tecnológico de pesquisas relacionadas à integração de dados espaciais, possibilitando a cooperação entre as instituições responsáveis por gerar informação geográfica.

1.3 Objetivos específicos

Visando alcançar o objetivo principal, alguns objetivos mais específicos são definidos de maneira a discriminar as etapas necessárias para que se consiga êxito na pesquisa. São eles:

1. Utilizar as especificações de serviços *web* e formato de dados GML do OGC em conjunto com o uso de ontologias de domínio e metadados, como um mecanismo

- tecnológico de automatização do processo de integrações sintática, semântica e de formato e estrutura entre objetos geográficos heterogêneos distribuídos;
2. Criar um ambiente de simulação utilizando duas fontes de dados distintas, com diferenças sintáticas, estruturais e semânticas do domínio de solos representando o ambiente real das instituições geradoras de objetos geográficos de solos;
 3. Modelar o conhecimento do especialista sobre o domínio de solos e compartilhar este conhecimento formalmente utilizando a linguagem das ontologias. O resultado desta modelagem é uma ontologia OWL sobre o domínio de solos que poderá ser utilizada em outros projetos;
 4. Desenvolver um protótipo para realizar a integração das bases de dados heterogêneas e fornecer o resultado em um formato geográfico conhecido. Junto com este protótipo deve-se fornecer uma biblioteca de código aberto que utilize os conceitos propostos na metodologia;
 5. Aplicar a metodologia em um estudo de caso utilizando o domínio de solos no âmbito do projeto Zon-dende.

1.4 Organização da dissertação

Nos capítulos que se seguem, serão levantados os fundamentos teóricos relacionados ao tema da dissertação, bem como a descrição da metodologia, o estudo de caso e os resultados alcançados.

O capítulo 2 descreve a fundamentação teórica que fornece o subsídio necessário para o desenvolvimento da metodologia. Primeiramente são apresentadas as iniciativas relacionadas à interoperabilidade e à padronização, principalmente em SIG; passando pelas especificações do *framework* de serviços *web* do consórcio OGC, o uso de metadados e terminando com o conceito de *web* semântica e ontologias.

No capítulo 3, é apresentada a descrição da metodologia proposta na pesquisa. Primeiro, é apresentada a definição da arquitetura orientada a serviços que deve ser utilizada, posteriormente é inserido o conceito de servidores de objetos geográficos e a utilização dos serviços do consórcio OGC que devem ser disponibilizados nestes servidores. Em seguida, a metodologia apresenta o processo de engenharia do conhecimento, com a criação de uma ontologia de domínio e o conceito de servidor de integração. Os serviços OGC que devem ser fornecidos pelo servidor de integração e o modo como os objetos geográficos devem ser

integrados, baseado na ontologia proposta, é apresentado na seqüência. Por fim, é sugerida uma maneira de disponibilizar os objetos geográficos já integrados.

No capítulo 4, é apresentado o estudo de caso proposto aplicando a metodologia descrita no capítulo anterior.

Os capítulos seguintes apresentam as observações conclusivas, as propostas de trabalhos futuros e as referências bibliográficas.

2 Fundamentos teóricos

2.1 Interoperabilidade e Padronização

Nos dias atuais é gerada uma grande quantidade de informação em formatos diferentes, que muitas vezes precisam interagir entre si para gerar nova informação. Esta troca de dados e informações heterogêneas em formato, sintaxe e semântica é chamada de interoperabilidade. Segundo o banco de dados léxico da universidade de Princeton, “interoperabilidade é a habilidade de trocar e usar informação” (17). Neste sentido, a crescente demanda por troca de informação gera a necessidade da elaboração de produtos e serviços cada vez mais capazes de realizar intercâmbio de dados.

Nas últimas décadas mecanismos de padronização vêm sendo criados para facilitar a interoperabilidade entre materiais, produtos, processos e serviços. Organizações como a *International Organization for Standardization* (ISO), o *World Wide Web Consortium* (W3C) e outras, foram criadas com o objetivo de definir padrões técnicos que devem ser seguidos para que possamos trocar informações entre os produtos e serviços. Os padrões são documentos que contêm especificações técnicas aprovadas por um processo de aceitação estabelecido pelos organismos de padronização. Estes padrões devem ser seguidos como normas para que os recursos sejam capazes de interagir entre si.

No campo das geotecnologias, de uma maneira geral, cada software tem um modelo conceitual próprio. Esta variedade faz com que as instituições produtoras de informações geográficas gerem os seus dados espaciais seguindo a forma e a estrutura definida pelo SIG por ela escolhido. Hoje temos inúmeras formas diferentes de representar e armazenar os dados geográficos, variando de arquivos em formato definido pelo fabricante do SIG até bancos de dados espaciais. Sendo assim, a tarefa da interoperabilidade nesta área de conhecimento é uma necessidade cada vez mais urgente, principalmente se for levado em consideração o alto custo de produção deste tipo de informação.

2.1.1 O Open Geospatial Consortium (OGC)

Com a crescente demanda por interoperabilidade entre informações espaciais foi criado, em 1994, o consórcio OGC com o objetivo de desenvolver padrões internacionais para interoperabilidade entre dados espaciais (2). O consórcio internacional OGC conta com a participação de mais de 330 companhias, agências governamentais e universidades que através de um processo consensual se propõem a desenvolver especificações públicas que permitam a interoperabilidade entre as diferentes geotecnologias.

2.2 Serviços *WEB* e as especificações do *framework* OGC

Um serviço *web* é um conceito utilizado para tornar as funcionalidades de um sistema de informação acessíveis através de tecnologias *web* padrão (18). O objetivo é facilitar a integração entre aplicações fornecendo uma interface de troca de informações utilizando tecnologia padronizada e assim reduzindo a heterogeneidade. A partir da especificação dos serviços *web* por parte do consórcio W3C, um novo paradigma foi criado com a introdução das Arquiteturas Orientadas a Serviços (SOA). O consórcio W3C define um serviço *web* como “uma aplicação identificada por um URI (*Uniform Resource Identifier*) cujas interfaces são capazes de ser definidas, descritas e descobertas na forma de um artefato XML. Um serviço *web* suporta interações diretas com outros agentes usando mensagens baseadas em XML, trocadas via protocolos baseados em Internet”. O consórcio OGC especifica um *framework* arquitetural de serviços *web* (figura 2.1) utilizado para fornecer informações geográficas (8).

Entretanto, o padrão utilizado nas especificações OGC não segue as recomendações contidas nas especificações do W3C para definição de serviços *web* (11). A arquitetura OpenGIS Services Framework não usa necessariamente os padrões usuais para serviços *web* do consórcio W3C, tais como Web Services Definition Language (WSDL) para descrições de serviço, e Universal Description, Discovery and Integration Protocol (UDDI) para implementar serviços de diretório. No lugar da UDDI, por exemplo, os serviços OGC propõem o uso de serviços de catálogo para realizar as operações de publicação, localização e seleção de serviços. Os serviços *web* OGC não utilizam descritores de serviços, já os serviços *web* do consórcio W3C utilizam WSDL para descrever a interface dos seus serviços. Os serviços OGC utilizam um tipo específico de XML, o Geographic Markup Language (GML)

para transmitir os objetos geográficos. As diferenças entre serviços OGC e W3C precisam ser eliminadas rapidamente, tornando os dois padrões compatíveis.

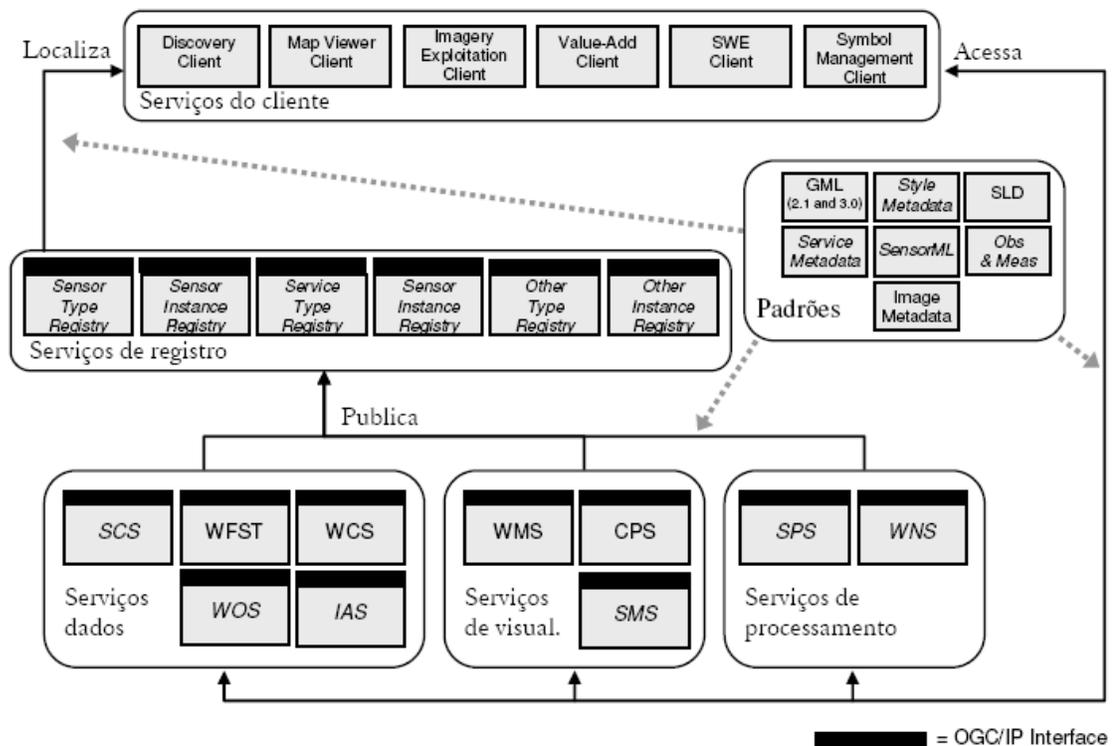


Figura 2.1 Framework Arquitetural dos Serviços web do OGC (8)

Algumas das especificações do *framework* destacam-se pela importância e pela utilização nas mais recentes versões dos softwares de geoprocessamento. A principal delas é a linguagem GML que define propriedades espaciais e não espaciais das feições geográficas (*features*) e é utilizada para o transporte e armazenamento de informações geográficas. Ultimamente, as especificações *Web Map Services* (WMS) e *Web Feature Services* (WFS) vêm sendo adotadas por SIGs e Servidores de Mapas. Outras especificações como *Web Coverage Service* (WCS), *OpenGIS Catalog Service* (OCS), *Web Registry Service* (WRS) e *Coordinate Transformation Service* (TS) ainda não despertam tanto interesse, porém, têm um papel importante na tentativa de realizar a interoperabilidade entre dados espaciais. A seguir algumas especificações estão apresentadas com mais detalhes.

2.2.1 Geographic Markup Language (GML)

A linguagem GML foi especificada para definir uma maneira de codificar dados geográficos, com base no padrão *Extensible Markup Language* (XML) do consórcio W3C. O

objetivo da GML é fornecer um conjunto de regras que definam as propriedades espaciais e não espaciais de feições geográficas de forma extensível, ou seja, de tal maneira que um usuário possa definir sua própria linguagem para descrever os dados do seu domínio de interesse.

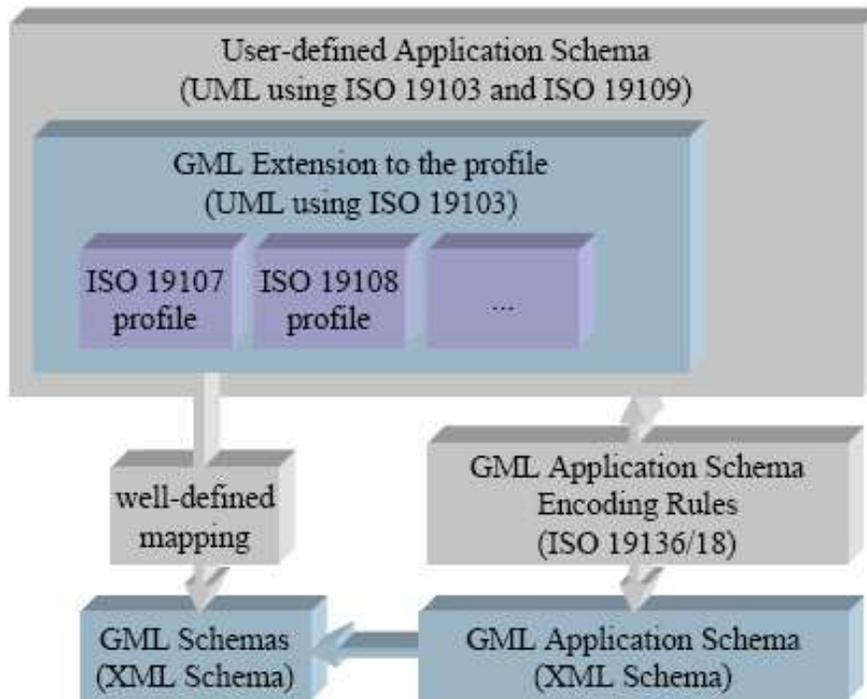


Figura 2.2 Definição do esquema GML (7)

A especificação GML é baseada em esquema XML (*XMLSchema*) (10) e é adotada tanto pelo consórcio OGC quanto pela ISO. A figura 2.2 apresenta o processo de criação de uma linguagem definida pelo usuário para representar os objetos geográficos do seu domínio de interesse, baseada nas definições da especificação GML.

O esquema GML define os elementos que serão utilizados no documento GML através de identificações (*tags*) XML que descrevem o seu conteúdo. O modelo GML (figura 2.3) contém especificações para topologia (representação geométrica), feições geográficas (informações espaciais e não espaciais - propriedades dos objetos), superfícies, sistema de referência de coordenadas, informações temporais (características temporais dos elementos geográficos), metadados, estilos e outras propriedades que definem o documento.

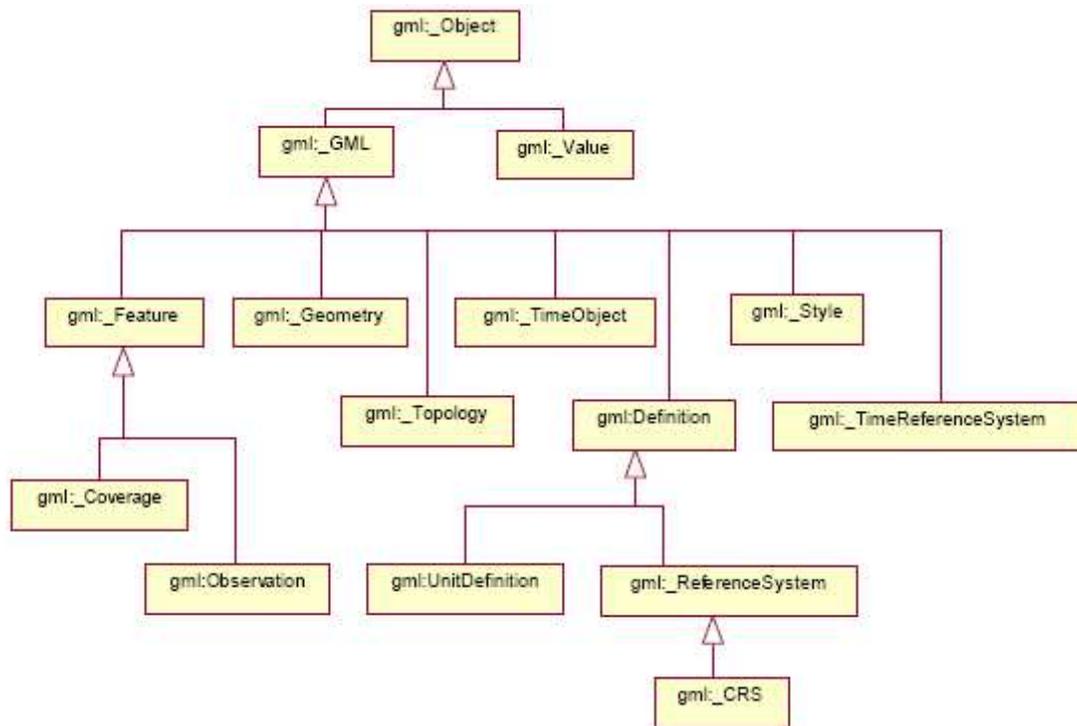


Figura 2.3 Hierarquia de classes do modelo GML (7)

O modelo GML possui geometrias como ponto, linhas, polígonos e suas respectivas coleções (*MultiPoint*, *MultiLine* e *MultiPolygon*) definidos por coordenadas de uma, duas ou três dimensões. As localizações espaciais são associadas a sistemas de referência espacial representadas por coordenadas cartesianas, já que o modelo não prevê o uso de projeções cartográficas.

2.2.2 Web Feature Service (WFS)

A especificação *Web Feature Service* (WFS) do consórcio OGC define um serviço *web* para recuperar e realizar transações com objetos geográficos (*features*). Com este serviço, é possível recuperar informações de metadados do servidor que provê o serviço, informações do esquema XML que define as feições geográficas e os próprios objetos espaciais requeridos. O serviço básico possui somente as funções de consulta, enquanto o transacional, também conhecido como WFS-T, permite as operações de inserção, exclusão e atualização de objetos geográficos. A figura 2.4 apresenta as operações definidas na especificação do serviço WFS segundo uma seqüência típica de execução.

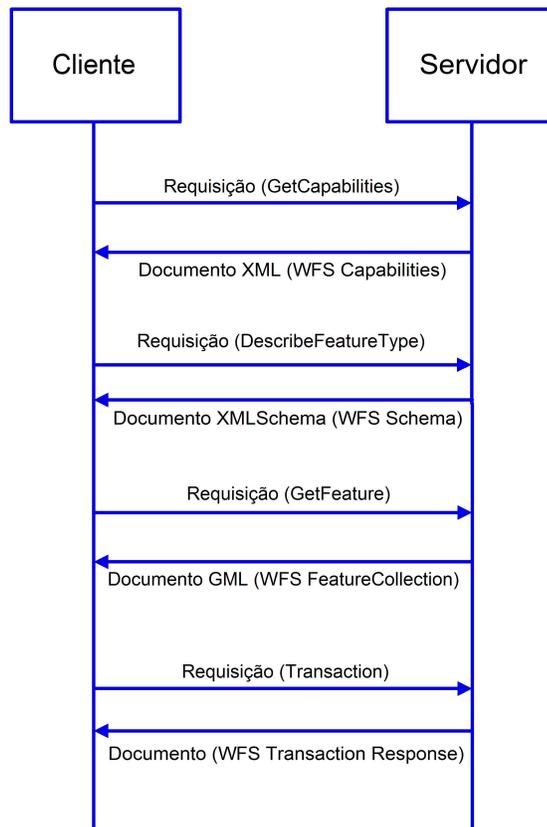


Figura 2.4 Operações do serviço WFS (25)

A seguir são apresentadas as operações definidas para o serviço WFS:

- *getCapabilities* – descreve os metadados que definem as características do servidor apresentando as operações que podem ser requeridas e os tipos de objetos geográficos (*FeatureTypes*) disponíveis dentre outras informações. Esta operação é comum a vários serviços do *framework* OGC;
- *describeFeatureType* – descreve a estrutura do tipo de objeto geográfico requisitado na forma de documento *XMLSchema*;
- *getFeature* – retorna a coleção de objetos espaciais (*gml:FeatureCollection*) disponível como um documento GML, seguindo o esquema conceitual apresentado pela função *describeFeatureType*. A resposta pode ser alterada por critérios de consulta;
- *transaction* – modifica remotamente os objetos geográficos disponíveis, alterando o seu conteúdo através de operações de inserção, remoção e alteração de objetos espaciais.

As requisições e respostas das operações WFS seguem os protocolos utilizados na internet seguindo os padrões definidos pelo consórcio W3C e utilizam linguagem XML.

2.2.3 Web Map Service (WMS)

O consórcio OGC fornece uma especificação para representar graficamente informações geográficas em forma de mapas. Com o serviço *web* WMS o usuário pode obter informações sobre o servidor, quais tipos de mapas ele fornece, é possível visualizar um conjunto de feições geográficas na forma de imagem (*JPG*, *GIF*, *PNG*) ou em formato vetorial (*SVG*) e obter informações sobre um objeto espacial específico. O serviço WMS permite que objetos espaciais armazenados em fontes heterogêneas possam ser visualizados juntos com a combinação de mapas.

O serviço possui três operações:

- *getCapabilities* – fornece os metadados do servidor, descreve os parâmetros e tipos de mapas que podem ser visualizados;
- *getMap* – retorna o mapa requisitado em forma de imagem seguindo os parâmetros passados na requisição;
- *getFeatureInfo* – recupera as informações de um objeto geográfico particular do mapa.

2.2.4 OpenGIS Catalog Service (OCS)

Serviços de catálogo publicam coleções de informações descritivas (metadados) para os dados, serviços e objetos relacionados. Os metadados descrevem as características do recurso que se deseja avaliar quando existe necessidade de realizar consultas. Os serviços de catálogo são utilizados quando se deseja descobrir informações sobre recursos registrados.

A especificação OGC para o serviço *web* de catálogo define uma interface que permite que as aplicações consultem e encontrem recursos espaciais como operações, serviços e dados espaciais em servidores heterogêneos distribuídos. Os metadados funcionam como propriedades dos recursos que são consultadas e avaliadas, e se for do interesse, recuperados.

As consultas são realizadas utilizando a linguagem *Common Catalog Query Language* (8), e os metadados utilizados seguem as recomendações da norma ISO19115 (*Geographic Information - Metadata*) (19).

2.2.5 Web Registry Service (WRS)

Para utilizar os produtos compatíveis com as especificações dos serviços *web* do consórcio OGC, é preciso primeiro localizar os servidores que fornecem estes serviços. Neste sentido, o OGC especifica um serviço de localização de servidores de serviços OGC chamado *Web Registry Service* (WRS). Os responsáveis por fornecer o serviço devem registrar os seus servidores em um ou mais servidores WRS para que os serviços possam ser localizados e utilizados. Esta especificação, entretanto, ainda se encontra em fase de discussão (8).

2.3 Metadados Geográficos

Metadados são informações descritivas sobre os dados. Eles descrevem o conteúdo, origem, qualidade, condição, histórico, localização e outras características dos dados disponíveis. O seu objetivo é fornecer um mecanismo que permita a identificação dos dados existentes, e a partir daí permitir a escolha de quais dados deverão ser acessados e utilizados.

A padronização de metadados vem sendo desenvolvida com o objetivo de tratar a interoperabilidade no nível de gerenciamento dos dados visando facilitar a busca de informações em um ambiente heterogêneo. Quando se trata de dados espaciais, algumas organizações como a ISO, o OGC e o *Federal Geographic Data Committee* (FGDC) vêm definindo terminologias comuns para os conceitos relacionados aos metadados geográficos.

O FGDC é um comitê entre agências para promover a coordenação do uso, intercâmbio e disseminação de informações espaciais nos EUA (5). O padrão FGDC define um conjunto de termos para documentar as informações espaciais. Os seguintes tópicos são abordados por este padrão: identificação, qualidade dos dados, organização espacial, referência espacial, entidades e atributos, distribuição e referência dos metadados (5).

A ISO possui uma especificação para metadados geográficos, denominada ISO 19115:2003 *Geographic Information – Metadata*, que define um grande conjunto de elementos de metadados que podem ser utilizados em informações geográficas. A especificação ISO 19115 consolida as tentativas de padronização realizadas anteriormente, inclusive a da FGDC em um único padrão (19). O modelo de metadados da ISO 19115 (figura 2.5) possui por volta de 400 elementos sendo 20 elementos principais. O serviço de catálogo do OGC atualmente utiliza a especificação da ISO como mencionado no item 2.2.4.

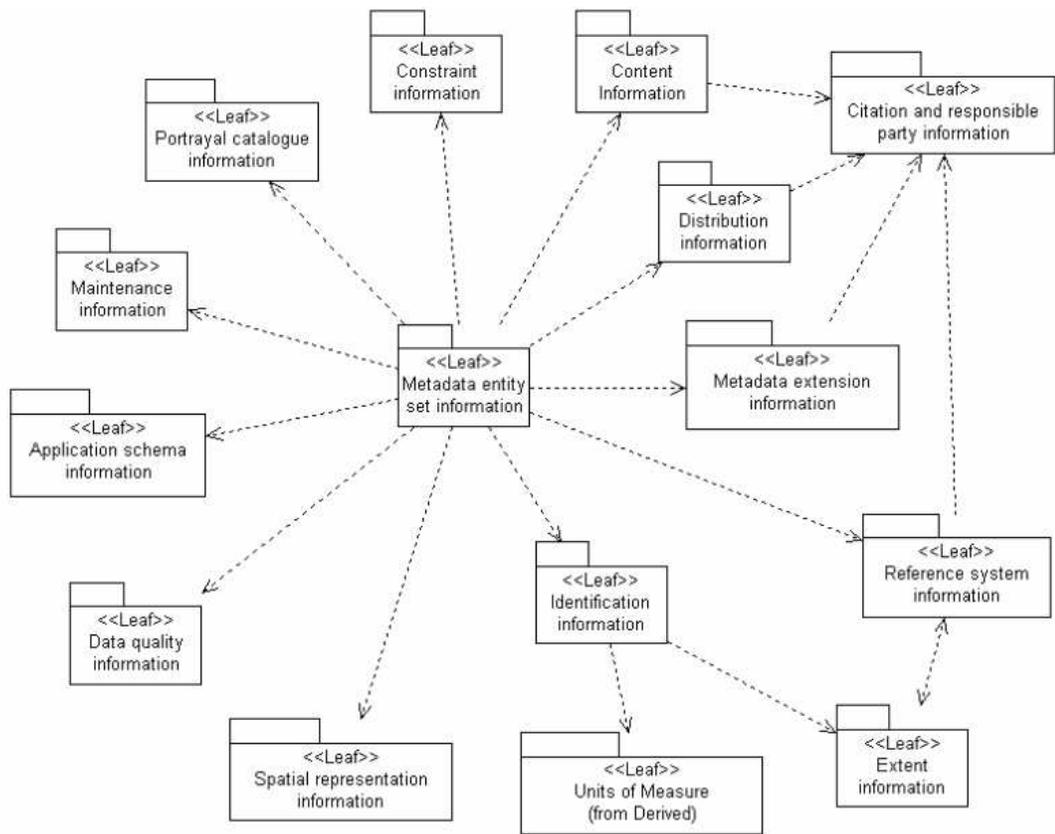


Figura 2.5 Modelo de Metadados ISO 19115 (19)

Casanova et. al. (5) faz uma análise sobre o uso excessivo de informações que descrevem o processo de produção dos dados geográficos, em detrimento da descrição da sintaxe e semântica destas informações, pelos padrões atuais de metadados. O número excessivo de elementos de metadados por sua vez é muitas vezes apontado pelos usuários como um fator que gera dúvida no momento da escolha dos elementos.

2.4 Base de Conhecimento, Ontologias e *Web Semântica*

Com o crescimento da área de inteligência artificial e mais particularmente dos sistemas especialistas, a tendência é que a representação do conhecimento em um formato estruturado esteja cada vez mais presente, fazendo com que as máquinas sejam capazes de realizar inferências dedutivas e indutivas a partir desta representação. Os sistemas especialistas são sistemas computacionais que utilizam uma base de conhecimento para resolver problemas que tradicionalmente necessitariam de um especialista humano em um domínio de interesse.

Portanto, a base de conhecimento é o componente do sistema especialista que armazena fatos sobre os termos do domínio de interesse e as relações entre os termos e os fatos.

Neste cenário, surgem as ontologias, no âmbito dos sistemas especialistas, herdando o termo homônimo da filosofia. Gruber (20) define uma ontologia como uma especificação explícita de uma conceituação: conceitos, objetos e outras entidades que se assume existir em uma área de conhecimento e o relacionamento entre eles. A conceituação referida é uma visão simplificada e abstrata do universo de discurso cujo conhecimento se deseja representar de maneira formal para atender a um propósito específico.

Segundo Noy e McGuinness (13), ontologias são comumente encontradas na *web*, variando de taxonomias para categorizar *sites web* até categorização de produtos e serviços para venda. Muitas linguagens e ferramentas vêm sendo desenvolvidas para representar ontologias ultimamente. Das iniciativas que envolvem o uso de ontologias na *web*, a mais significativa é a definição do consórcio WWW (W3C) de *web* semântica (12) que pretende embutir conhecimento semântico e contexto nas páginas publicadas na internet. O objetivo da *web* semântica é melhorar a forma de interação entre as páginas *web*, os programas e os usuários. Para isto, a iniciativa propõe a alteração do foco das páginas dos documentos para os dados e seus significados, utilizando uma linguagem que relaciona os dados com objetos do mundo real.

O desenvolvimento mais recente em padrão de linguagem de ontologia, é a *Ontology Web Language* (OWL), que é uma linguagem de marcação que estende o vocabulário do *Resource Description Framework* (RDF), com o objetivo de definir, publicar e compartilhar ontologias na *web* (21). Com ela, é possível definir os conceitos do domínio de interesse utilizando os indivíduos, classes e propriedades do seu universo e descrevê-los segundo um modelo lógico que conta com um conjunto rico de operadores. Entre as ferramentas, atualmente existe uma gama de ambientes de edição de ontologias que facilitam o trabalho de criação da base de conhecimento. Dentre eles pode-se destacar o Protégé (22), desenvolvido pelo laboratório de sistemas de conhecimento da Universidade de Stanford, e o Ontolingua (23).

2.4.1 Ontology Web Language (OWL)

A linguagem OWL é uma tentativa do consórcio W3C de criar uma linguagem padrão para representar ontologias (21). Uma ontologia pode ser modelada utilizando os três componentes básicos da linguagem OWL: Indivíduo, Classe e Propriedade. Os indivíduos

representam os objetos físicos ou lógicos, concretos ou abstratos, em um domínio de interesse. As propriedades são relações binárias estabelecidas entre os indivíduos do universo de discurso. Os indivíduos geralmente agrupam-se em conjuntos com características em comum. Estes conjuntos de indivíduos são as classes, que na ontologia é definida por um conjunto de regras que definem quais indivíduos podem pertencer a quais classes.

A linguagem OWL possui uma estrutura semelhante à orientação a objetos, onde uma hierarquia de classes deve ser definida através das suas relações de especialização e generalização. Assim como na orientação a objetos, todos os indivíduos da ontologia que pertencem à classe mais específica devem pertencer também à classe mais geral da hierarquia.

Na ontologia, as propriedades estabelecem uma relação entre indivíduos (*object properties*) ou entre um indivíduo e um valor literal (*datatype properties*). Dentro da hierarquia de classes, uma propriedade que pertence à classe mais geral é herdada pela classe mais específica, portanto, ao definir à que classe de indivíduos a propriedade pertence deve-se considerar a classe mais geral possível na hierarquia. Toda propriedade que relaciona dois indivíduos possui uma propriedade inversa que vincula as classes no sentido oposto. As propriedades em uma ontologia possuem um domínio que determina as classes que utilizam a propriedade, e uma faixa de valores possíveis (*range*) que determina os valores que ela pode assumir dentro da hierarquia de classes. As propriedades vinculam indivíduos do seu domínio a indivíduos do seu *range*.

Um aspecto importante a ser ressaltado na linguagem OWL é o reuso de ontologias. A linguagem permite a importação de ontologias já criadas e compartilhadas em bibliotecas de ontologias. Ao importar uma ontologia, todos os elementos, descrições e axiomas contidos na ontologia importada estarão disponíveis para serem utilizados.

Description Logic (OWL-DL)

A description logic é uma linguagem de representação do conhecimento que visa, por um lado, descrever os elementos do domínio de maneira estruturada e formal e, por outro lado, descrever a semântica do domínio baseada na lógica de primeira ordem.

As classes dentro da ontologia são definidas por um conjunto de restrições que determinam quais indivíduos podem pertencer a ela. Estas restrições são definidas pelas condições necessárias e pelas condições necessárias e suficientes atribuídas à classe. Nas condições necessárias, os indivíduos que atendam às restrições podem ser membros da classe, mas não há garantia de que todos os indivíduos que atendam às restrições sejam membros da

classe. As condições necessárias e suficientes, por sua vez, definem que todos os indivíduos que atendam a estas condições são membros da classe.

Para definir as condições de cada classe, existe um conjunto de operadores lógicos de restrição na linguagem OWL-DL (tabela 2.1).

Grupo	Nome	Símbolo	Significado
Quantificador	Existencial	\exists	Existe algum valor de [P] que pertence a [C]
Quantificador	Universal	\forall	Todos os valores de [P] pertencem a [C]
Cardinalidade	Cardinalidade Mínima	\leq	Número mínimo de relações de [P]
Cardinalidade	Cardinalidade Máxima	\geq	Número máximo de relações de [P]
Cardinalidade	Cardinalidade Exata	$=$	Número exato de relações de [P]
Valor	Tem Valor	\ni	[P] possui valor igual a [C]
Valor	Classes Enumeradas	$\{ \}$	[P] possui valor contido em $\{I_1, I_2, I_3, I_4, \dots\}$
Lógicos	Complemento	\neg	[I] que não pertencem a [C]
Lógicos	Interseção	\cap	$[C_1] \cap [C_2]$ - [I] que pertencem a $[C_1]$ e $[C_2]$
Lógicos	União	\cup	$[C_1] \cup [C_2]$ - [I] que pertencem a $[C_1]$ ou $[C_2]$

[P] – Propriedade
 [C] – Classe ou Literal
 [I] – Indivíduo

Tabela 2.1 Operadores lógicos de restrição em OWL

Através das condições definidas para cada classe, podemos validar a ontologia criada utilizando uma máquina de inferência capaz de verificar se os indivíduos cadastrados podem pertencer às suas respectivas classes e se eles podem ou devem pertencer a outras classes.

Semantic Web Rule Language (SWRL)

A SWRL é uma proposta da web semântica para prover uma linguagem de regras que combina a OWL-DL com a *Rule Markup language*. As regras retratam uma implicação entre um antecedente e um conseqüente, de tal maneira que as condições especificadas para o antecedente, se satisfeitas, fazem com que as condições especificadas no conseqüente também sejam satisfeitas. Um exemplo de expressão em SWRL em uma representação hipotética de uma família pode ser visto abaixo:

$Pessoa(?x) \wedge temPai(?x,?y) \wedge temIrmao(?y,?z) \rightarrow temTio(?x,?z)$

Onde:

- Pessoa – Classe que representa uma pessoa no domínio família;
- temPai – Propriedade que relaciona as classes Pessoa e Pai no domínio família;

- temIrmão – Propriedade que relaciona as classes Pessoa e Irmão no domínio família;
- \wedge - símbolo de intercessão entre átomos da expressão;
- x, y, z – variáveis da expressão;
- \rightarrow - símbolo de implicação.

A expressão acima, depois de processada por uma máquina de inferência, faria com que todos os indivíduos da ontologia que atendam as condições do antecedente, ou seja, uma pessoa X que tem um pai Y que tem um irmão Z , tenham novos valores atribuídos a propriedade temTio de acordo com as condições do conseqüente, ou seja, a pessoa X tem tio Z .

3 Metodologia

Para realizar a integração entre objetos geográficos de diversas fontes de dados com formato, sintaxe e semântica distintas, é proposta uma metodologia utilizando as especificações de serviços sugeridas pelo consórcio OGC, além da criação de uma ontologia de domínio e do uso de metadados. Desta forma, a metodologia é apresentada em um conjunto de seis etapas para que o objetivo final da integração seja alcançado. São eles:

1. Definição da Arquitetura Orientada a Serviços Geográficos (GSOA), com a definição dos Servidores de Objetos Geográficos (SOG) e do(s) Servidor(es) de Integração Geográficos (SIGeo);
2. Utilização de servidores *Web Feature Service* (WFS) para prover os objetos geográficos de cada fonte de dados;
3. Criação da ontologia do domínio em *Ontology Web Language* (OWL), com documentação HTML, utilizando um processo de engenharia do conhecimento;
4. Configuração dos serviços de cada servidor de objetos geográficos no servidor de integração;
5. Disponibilização das informações integradas preferencialmente em formato de serviços WFS e WMS.

A seguir, a figura 3.1 apresenta os passos da metodologia na forma de diagrama de atividades da *Unified Modelling Language* (UML):

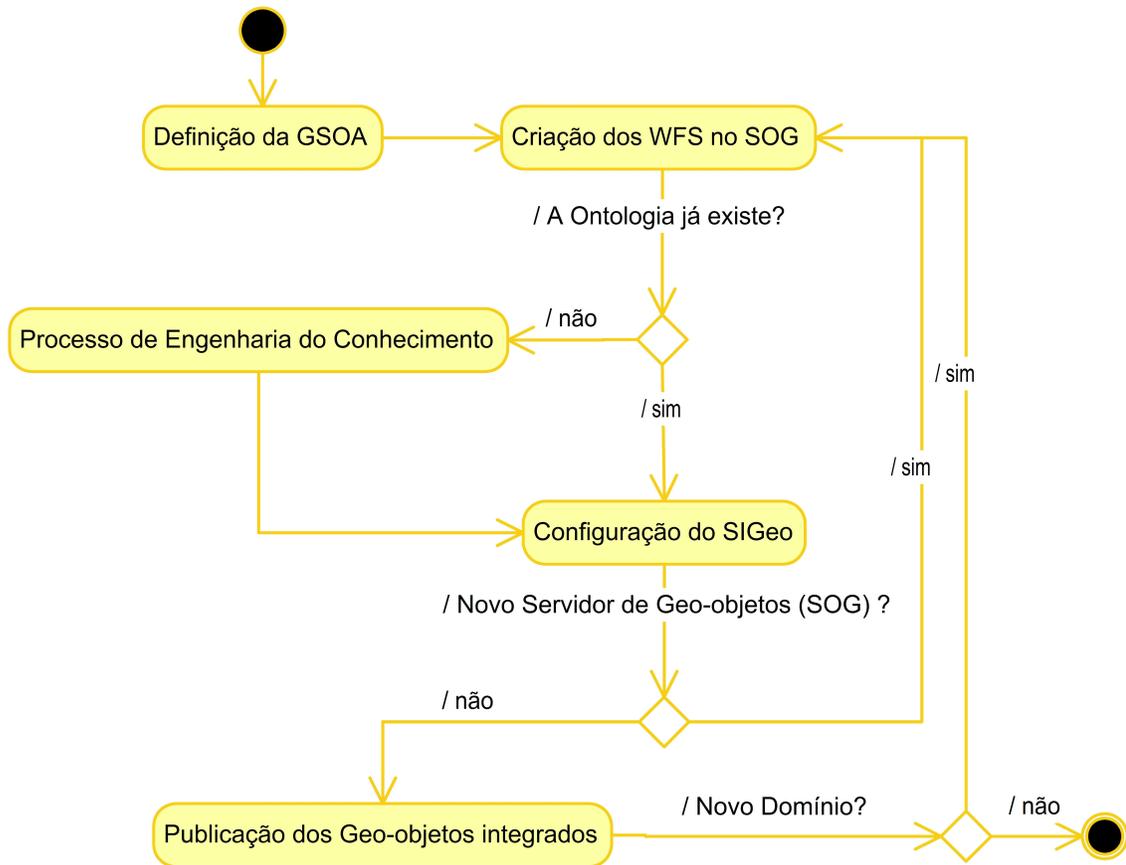


Figura 3.1 Diagrama de Atividades da Metodologia de Integração de Objetos Geográficos

3.1 Definição da arquitetura orientada a serviços do OGC

A primeira etapa da metodologia diz respeito à definição da arquitetura a ser utilizada. Inicialmente, são definidos dois tipos diferentes de servidores que deverão ser implementados: Servidores de Objetos Geográficos (SOG) e Servidores de Integração Geográficos (SIGeo).

Os SOG são responsáveis por fornecer as informações geográficas de cada instituição envolvida na integração, transformando os dados disponíveis para um formato padrão. Estes servidores devem utilizar um software compatível com a especificação do serviço WFS para tornar os objetos geográficos disponíveis em formato GML.

Os SIGeo, por sua vez, são responsáveis por registrar os serviços disponíveis nos SOG e unificar a sintaxe e semântica de cada elemento destes serviços para que todos eles sejam interpretados da mesma maneira. Para isto, os SIGeo devem tornar pública a ontologia do domínio, tanto em OWL quanto em documentação HTML, e utilizá-la para integrar sintática e

semanticamente os esquemas conceituais e os objetos fornecidos por cada serviço. Após a unificação dos dados geográficos, é preciso fornecer as informações espaciais (feições) e não-espaciais (atributos) agregadas para que sejam utilizadas de maneira integrada. Portanto, os SIGeo são responsáveis por prover o mecanismo de unificação das informações através da utilização dos serviços WFS, WMS e OCS que constam nas especificações do *OpenGIS Services Framework*.

Entretanto, a quantidade de SIGeo que deve ser utilizada no dimensionamento, bem como os domínios pelos quais cada SIGeo é responsável por associar definirão a arquitetura que será utilizada. Pode-se optar por uma arquitetura que utilize apenas um SIGeo com a responsabilidade de integrar todos os domínios necessários, ou então, uma segunda arquitetura com vários SIGeo, cada um destes responsáveis por integrar apenas um domínio. Uma terceira arquitetura pode definir determinados grupos de domínios que seriam integrados por um determinado SIGeo.

A seguir, são apresentados dois dos três casos mencionados, como diagramas de implantação da UML:

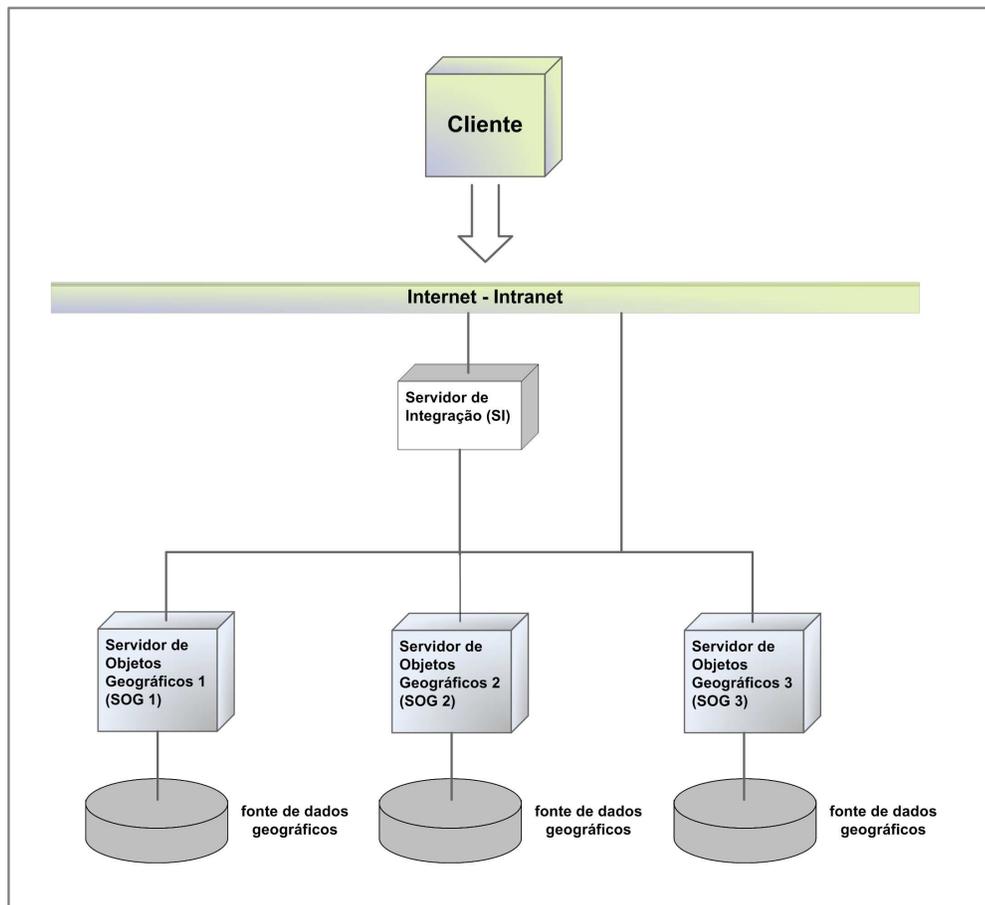


Figura 3.2 Diagrama da Arquitetura com apenas um SIGeo para todos os domínios

A figura 3.2 apresenta uma arquitetura simples com poucos domínios para serem integrados. Neste caso, o volume de informação a ser processada é pequeno e apenas um servidor pode responder pela unificação das informações geográficas de todos os domínios.

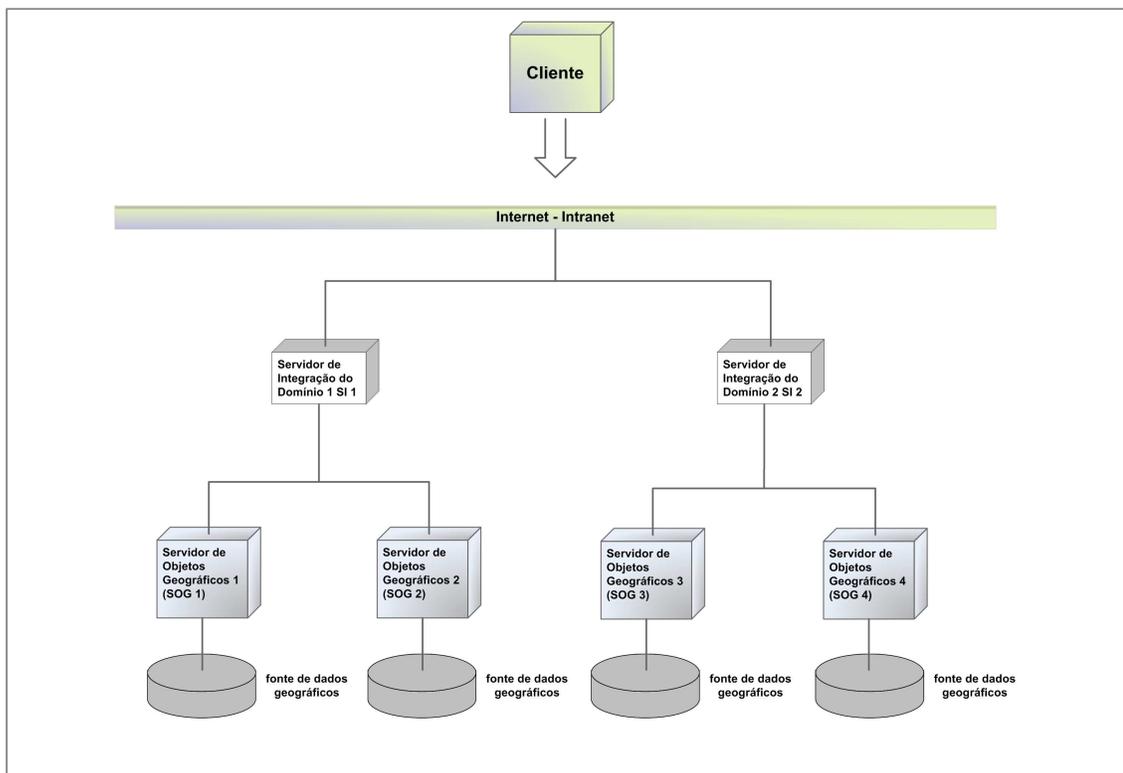


Figura 3.3 Diagrama da Arquitetura com um SIGeo para cada domínio

Quanto maior o número de domínios a serem utilizados, maior deverá ser a capacidade de processamento do servidor de integração que será responsável pela unificação dos dados geográficos dos diferentes domínios. Em projetos contendo diversos domínios, ou com muitos servidores de objetos geográficos para cada domínio, a idéia de designar um servidor de integração para cada domínio, como mostra a figura 3.3, pode ser utilizada para dimensionar melhor a carga no servidor de integração. Em alguns casos, pode ser interessante separar os domínios em grupos e designar um servidor de integração para cada grupo de domínios visando a melhoria do desempenho.

Outras arquiteturas podem ser utilizadas de acordo com as demandas, desde que alguns aspectos sejam levados em consideração:

- Deve existir pelo menos um domínio a ser integrado;

- É preciso que pelo menos um servidor de integração seja especificado;
- Devem ser especificados mais de um servidor de objetos geográficos, já que o objetivo é a integração das informações disponíveis em mais de uma instituição.

3.2 Criação dos serviços WFS nos servidores de objetos geográficos

A segunda etapa da metodologia diz respeito à criação dos serviços *Web Feature Service* (WFS) nos servidores de objetos geográficos. Depois de definida a arquitetura e conhecidos todos os servidores de objetos geográficos necessários para que os dados espaciais institucionais sejam fornecidos, é preciso automatizar o processo de busca e publicação destas informações armazenadas em cada instituição. Para isto, são definidos em cada servidor de objetos geográficos os serviços de WFS necessários para tornar os objetos geográficos disponíveis no formato GML, como apresentado na figura 3.5.

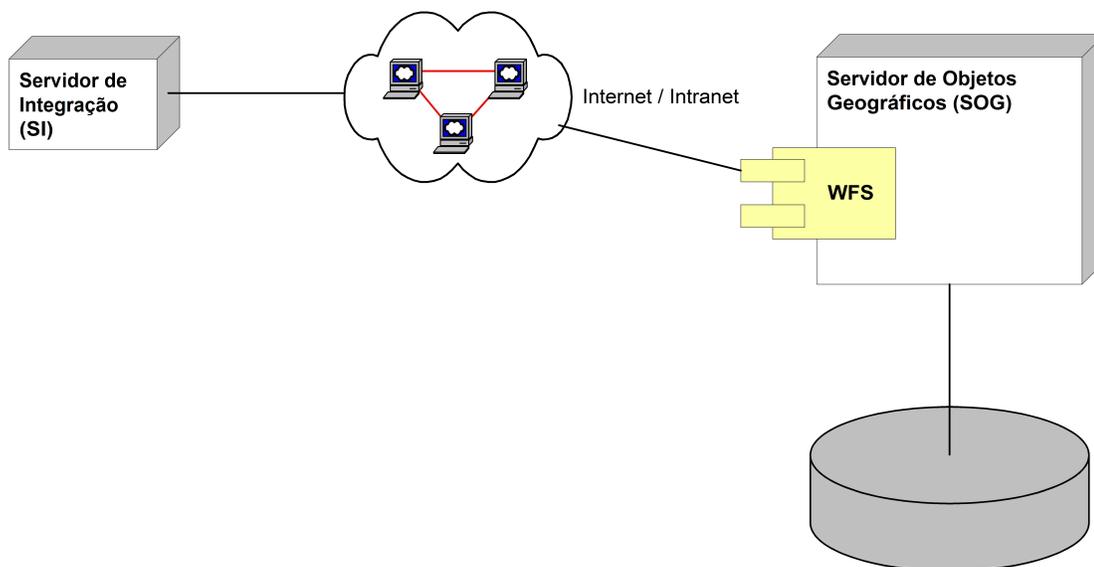


Figura 3.4 Diagrama de Implantação do Serviço WFS nos SOGs

Os principais objetivos desta etapa são: converter as estruturas de dados dos objetos geográficos disponíveis para um formato único (GML); e fornecer um mecanismo de acesso às informações armazenadas utilizando os protocolos da internet. A utilização de diversos Sistemas de Informações Geográficas para manipular as informações espaciais em cada instituição é uma situação bastante encontrada hoje em dia, desta maneira, cada ferramenta define o formato como os dados geográficos estão armazenados. Utilizando um servidor

WFS, fica transparente para o cliente a estrutura interna das informações espaciais armazenadas nos repositórios das instituições (figura 3.5). O serviço WFS fornece os objetos geográficos em GML, sendo assim, as diferentes formas de representação da informação espacial utilizadas pelos vários servidores de objetos geográficos são convertidas para GML e depois disponibilizada ao requisitante do serviço neste formato.

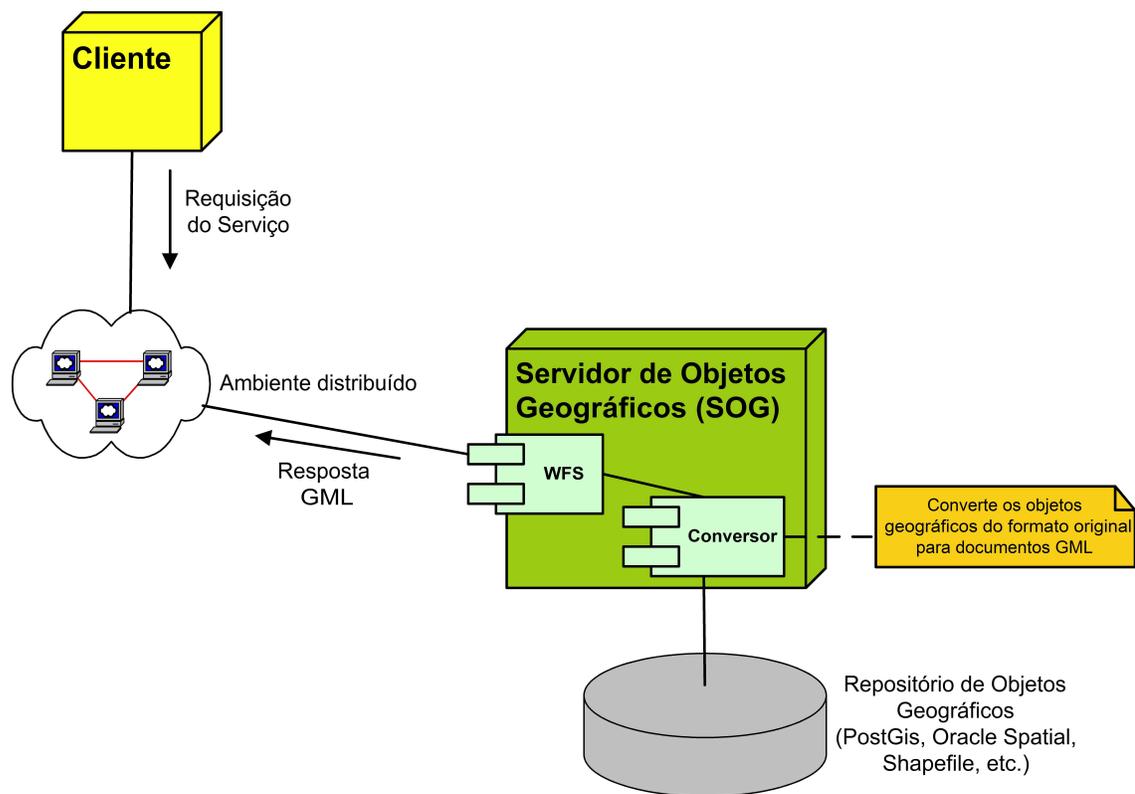


Figura 3.5 Diagrama de Implantação do funcionamento do Serviço WFS

Alguns SIGs e servidores de mapas já são compatíveis com a especificação WFS do consórcio OGC e convertem os objetos geográficos para a sua respectiva representação em GML. A utilização destas ferramentas é recomendável sempre que for possível, para que haja uma economia de esforços na produção do ambiente de integração. O consórcio OGC mantém em seu *web site* a relação das ferramentas compatíveis com as suas especificações (24). Estes produtos passam por um processo de teste de compatibilidade realizado pelo próprio consórcio. Algumas destas ferramentas são soluções de software livre com o código aberto, permitindo, desta maneira, que sejam modificadas para atender a uma necessidade específica.

Entretanto, a decisão de qual ferramenta utilizar dependerá das características de cada projeto e, principalmente, do formato de dados em que os objetos geográficos estão armazenados. Em alguns casos, pode ser necessário implementar o serviço, ou adequar a ferramenta, se o formato de dados utilizado não puder ser convertido. É importante salientar, no entanto, que a solução escolhida deve seguir as recomendações da especificação do serviço de *Web Feature Service* do consórcio OGC (25).

Ao final desta fase da metodologia, os servidores de objetos geográficos devem prover os serviços WFS necessários para fornecer os objetos geográficos disponíveis em formato GML. Desta forma, a unificação de formatos de dados será realizada resolvendo um dos níveis de incompatibilidade entre dados espaciais heterogêneos, além de fornecer um mecanismo de acesso padrão a estes dados.

3.3 Processo de Engenharia do Conhecimento

Nesta fase da metodologia, a unificação da estrutura dos dados espaciais foi realizada com a utilização dos serviços WFS nos servidores de objetos geográficos, porém não existe ainda uma unificação sintática e semântica entre eles. A terceira etapa da metodologia propõe a realização de um processo de engenharia do conhecimento, cujo objetivo é gerar ontologias para representar o conhecimento sobre os termos dos domínios e suas relações de maneira formal. É fornecido, desta maneira, um esquema conceitual de referência que possibilitará a realização da uniformização da sintaxe e da semântica das informações geográficas disponíveis em GML.

3.3.1 Atores

Dois atores são identificados para realizar a tarefa de representar o conhecimento sobre o domínio. Nesta metodologia são denominados: Engenheiro de Conhecimento e Especialista de Domínio. O Especialista de Domínio é o profissional que detém o conhecimento e tem experiência prática em lidar com a área que se deseja modelar. O Engenheiro do Conhecimento é o profissional que tem como missão extrair o conhecimento sobre o domínio do especialista e representá-lo formalmente através da criação de uma ontologia.

O Engenheiro do Conhecimento deve utilizar algumas técnicas já consagradas para ajudar a extrair o conhecimento do especialista. Podem ser:

1. entrevistas com os especialistas: realizada individualmente, tentando extrair o conhecimento e as experiências já vividas pelo especialista ao lidar com o domínio, já que cada um pode trazer contribuições diferentes sem a interferência de outros especialistas;
2. acompanhamento no ambiente do domínio: realizada no ambiente de trabalho do especialista, ajuda a entender melhor o domínio e as tarefas realizadas por ele;
3. revisão bibliográfica: Leitura de documentos consagrados indicados pelos próprios especialistas;
4. workshop: realizado em conjunto com vários especialistas, tenta confrontar o conhecimento de cada um, com o objetivo de chegar a um consenso quanto à especificação dos termos do domínio e de como eles se relacionam;
5. modelagem conjunta: utilizando uma ferramenta de modelagem adequada, visa trocar opiniões com os especialistas sobre os conceitos adotados.

Desta maneira, os engenheiros de conhecimento e os especialistas de domínio devem interagir entre si, com o objetivo de formalizar uma linguagem padrão de representação do conhecimento dos termos do domínio.

3.3.2 Reuso de ontologias

Ao iniciar o processo de engenharia do conhecimento do domínio, é preciso ter a idéia de como este domínio irá se relacionar com outros domínios e o que já foi realizado em termos de representação do conhecimento na área de estudo. Em alguns casos, o domínio já pode estar representado por uma ontologia e o engenheiro do conhecimento só precisaria reutilizá-la herdando as suas características e adaptando-a para a sua realidade.

Reusar ontologias torna-se uma necessidade quando se pretende interagir com outros domínios que já estão formalmente representados pelo uso de uma ontologia particular. Hoje em dia já existem algumas ontologias disponíveis em formato eletrônico que podem ser importadas e utilizadas em ferramentas de edição de ontologias, e a tendência é que haja um crescimento na produção e publicação das ontologias com o passar do tempo, incentivada principalmente pelo avanço nas pesquisas relacionadas à *web* semântica.

Da mesma maneira, ao se realizar o processo de engenharia do conhecimento, é desejável que se tenha preocupação em facilitar o reuso. Provavelmente, a ontologia criada ao

final do processo será útil para outros engenheiros do conhecimento que poderão reutilizá-las em seus processos de engenharia do conhecimento.

Segundo Noy e McGuinness (13), existem bibliotecas de ontologias na *web* e na literatura. Alguns exemplos são as bibliotecas de ontologias Ontolingua (23), DAML (26) e Protégé (22), o que facilitaria tanto a reutilização de ontologias já cadastradas, quanto o cadastramento das ontologias geradas.

3.3.3 Formatos gerados

Cada domínio deve ter a sua ontologia materializada em dois formatos: em forma de documentação em HTML; e, em linguagem OWL, sendo este último de uso obrigatório. A ontologia em documentação HTML, disponível na Internet ou Intranet, facilita o entendimento dos especialistas do domínio e dos usuários da ontologia, de maneira geral, sem que estes precisem consultar um documento em OWL, que, por ser baseado em XML, não é de fácil entendimento. A representação em linguagem OWL tem como principal objetivo fornecer uma estrutura formal do esquema conceitual do domínio que pode ser facilmente interpretada pela aplicação responsável por realizar a integração entre as várias fontes de dados disponíveis nos servidores de objetos geográficos. Os dois formatos devem ficar disponíveis no servidor de integração geográfico para que sejam compartilhados por todos os envolvidos.

Para a elaboração da ontologia recomenda-se que sejam utilizados editores de ontologias com capacidade de exportar a ontologia no formato OWL e gerar a documentação em HTML. Os editores de ontologias possibilitam a modelagem da ontologia graficamente, facilitando assim o entendimento entre as partes envolvidas. A seguir é apresentado, na figura 3.7 e na listagem 3.1, um exemplo resumido da ontologia no domínio da hidrografia baseado nas informações disponíveis na Agência Nacional de Águas (ANA), utilizando o editor de ontologias *Protégé* (22).

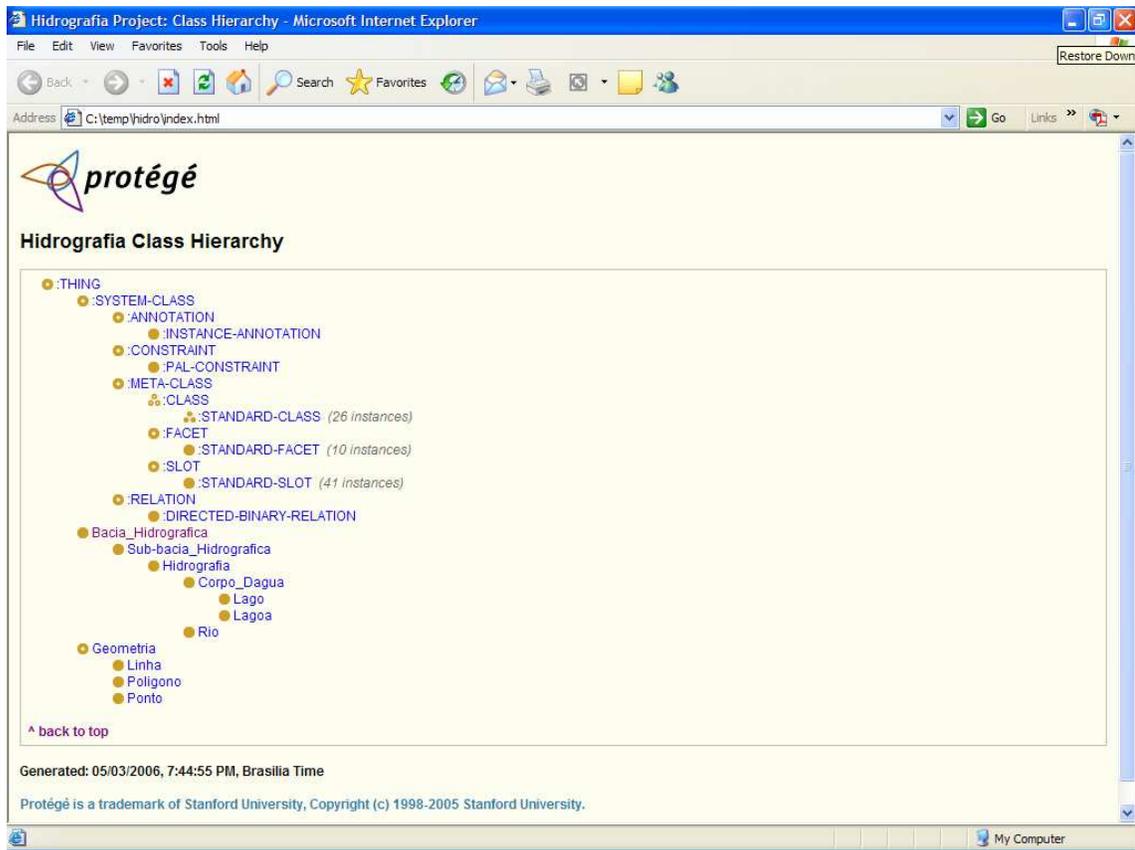


Figura 3.6 Documentação da Ontologia no Domínio da Hidrografia gerada pelo *Protégé*.

```

<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns="http://www.owl-ontologies.com/unnamed.owl#"
  xml:base="http://www.owl-ontologies.com/unnamed.owl">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:about="#Bacia_Hidrografica">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Representa uma bacia hidrográfica.</rdfs:comment>
  </owl:Class>
  <owl:Class rdf:ID="Sub-bacia_Hidrografica">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Representa uma sub-bacia hidrográfica</rdfs:comment>

```

```

<rdfs:subClassOf>
  <owl:Class rdf:ID="Bacia_Hidrografica"/>
</rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Hidrografia">
  <rdfs:subClassOf rdf:resource="#Sub-bacia_Hidrografica"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Representa um rio, lago, lagoa, etc.</rdfs:comment>
</owl:Class>
.....
.....
</rdf:RDF>
<!-- Created with Protege (with OWL Plugin 2.1, Build 284) http://protege.stanford.edu -->

```

Listagem 3.1 Listagem da ontologia no domínio da hidrografia em OWL gerada pelo *Protégé*

Ao gerar e compartilhar a ontologia em uma linguagem formal padrão, os usuários do domínio passam a possuir um referencial sintático e semântico sobre a sua área de conhecimento. Aplicações que relacionadas à área de conhecimento representada pela ontologia criada, podem e devem utilizá-la como um modelo de referência do campo de conhecimento para que o processo de integração seja facilitado.

3.4 Configuração do servidor de integração

Para que a interoperabilidade entre dados espaciais heterogêneos seja completamente realizada, é preciso criar um mecanismo automatizado de unificação dos dados geográficos disponíveis nas várias fontes de dados. A quarta etapa da metodologia define a configuração deste mecanismo automatizado em duas etapas. A primeira é o cadastramento no servidor de integração, das informações que definem os serviços WFS disponíveis em cada servidor de objetos geográficos, utilizando uma estrutura de metadados. Esta estrutura deve fornecer informações sobre os dados disponíveis, qualidade dos dados, ano de coleta da informação, instituição e etc. A segunda etapa é definida pela criação de uma correlação entre o esquema conceitual de cada fonte de dados disponível a partir do serviço WFS e o esquema de

referência definido na forma de uma ontologia do domínio em OWL. A seguir, as ações são descritas com maiores detalhes.

3.4.1 Cadastramento dos serviços baseado em metadados

Para que o servidor de integração localize os objetos geográficos que devem ser integrados, é preciso que as suas fontes de dados estejam cadastradas em algum lugar da rede. As fontes de dados, que neste caso são os serviços WFS criados nos servidores de objetos geográficos, precisam ser cadastradas em um servidor que registre os dados relevantes e os tornem disponíveis através de serviços de registro. Sendo assim, propõe-se a utilização do servidor de integração como elemento registrador de todos os serviços WFS disponíveis em cada servidor de objetos geográficos especificado na arquitetura.

É recomendável que estes registros de serviços tornem-se públicos pelo uso de um serviço específico. O consórcio OGC criou a especificação *Web Registry Server* (WRS) com este propósito, ou seja, fornecer um serviço que seja capaz de prover uma estrutura de localização de serviços e servidores na rede. Entretanto, ela ainda não foi publicada, o que torna a sua utilização pouco confiável atualmente.

O local na rede onde os serviços se encontram não é a única informação que deve ser cadastrada. Outras informações que definem as características dos dados fornecidos pelos serviços devem ser fornecidas, tais como o seu conteúdo, ano de levantamento dos dados, localização geográfica, instituição responsável e etc. Propõe-se a utilização de um catálogo de metadados para fornecer informações sobre os dados disponíveis. A definição de qual informação deve ser cadastrada pode variar de domínio para domínio, já que cada área de conhecimento e cada comunidade de usuários têm características muito particulares.

Recomenda-se que a escolha dos metadados a serem utilizados seja feita com base no padrão ISO19115 – *Geographic Information Metadata*, que serve de base para o esquema de metadados básico adotado pelo consórcio OGC (8).

A utilização da especificação *OpenGIS Catalog Services* (OCS) do consórcio OGC, como meio de publicar o catálogo com as coleções de informações sobre os objetos geográficos fornecidos, incluindo um mecanismo de busca baseada nos metadados e também na ontologia, pode fornecer um meio flexível de acesso aos dados em diferentes fontes de dados geográficas heterogêneas.

Sugere-se a criação de uma interface *web* de cadastramento dos serviços WFS disponíveis em cada servidor de objetos geográficos com a inclusão dos metadados definidos.

Esta interface deve ser capaz de armazenar as informações cadastradas para que sejam consultadas posteriormente. O formato de armazenamento das informações de publicação pode variar de acordo com as opções feitas pela utilização ou não utilização dos serviços de publicação de serviços (WRS) e de catálogos de metadados (OCS).

3.4.2 Correlação sintática e semântica entre os esquemas conceituais

Neste ponto, o servidor de integração detém as informações sobre a localização dos serviços que disponibilizam os objetos geográficos na rede, entretanto, é preciso tratar a heterogeneidade sintática e semântica entre as classes e propriedades destes elementos e as classes e propriedades definidas na ontologia de referência do domínio.

Em uma situação fictícia, a seguir são descritas duas fontes de dados com o esquema conceitual dos objetos geográficos no domínio da hidrografia para exemplificar a heterogeneidade semântica entre elas, assim como as suas possíveis diferenças em relação à ontologia de referência. Os esquemas conceituais do exemplo estão representados da seguinte maneira:

Servidor de Objetos Geográficos 1 – SOG1

Classe: Bacia

Atributos: codigo – código da bacia hidrográfica;

geometria - geometria do objeto geográfico (ponto, linha, polígono...);

nome – nome da bacia hidrográfica;

area – área da bacia hidrográfica;

Classe: Sub-bacia

Atributos: codigo – código da Sub-bacia hidrográfica;

geometria - geometria do objeto geográfico (ponto, linha, polígono...);

codigo_bacia - código da bacia hidrográfica a qual a sub-bacia pertence;

nome – nome da sub-bacia hidrográfica;

area – área da sub-bacia hidrográfica;

Classe: Rio

Atributos: codigo – código do rio;

geometria - geometria do objeto geográfico (ponto, linha, polígono...);

codigo_subbacia – código da sub-bacia a qual o rio pertence

nome – nome do rio;

extensao – extensão do rio;
volume – volume médio anual de água;
largura – largura média do rio;
profundidade – profundidade média do rio;

Servidor de Objetos Geográficos 2 – SOG2

Classe: HIDROGRAFIA

Atributos: OID – código da hidrografia;

GEOMETRY – geometria do objeto geográfico (ponto, linha, polígono...);

BACIA – nome da bacia hidrográfica;

SUBBACIA – nome da sub-bacia hidrográfica;

NOME – nome da hidrografia;

TIPO – tipo de hidrografia (rio, lago, lagoa e etc)

LARGURA – largura média da hidrografia;

PROFUND – profundidade média do rio;

Nesta representação, podem ser notadas as diferenças entre os dois esquemas conceituais dos objetos geográficos armazenados em duas fontes de dados, SOG1 e SOG2. Percebe-se que no SOG1, a estrutura é mais bem elaborada, com o uso de códigos que identificam cada objeto e os objetos separados em classes bem definidas. Este esquema conceitual se assemelha ao esquema relacional utilizado em modelos de banco de dados. Os objetos deste primeiro esquema devem supostamente fazer parte de um banco de dados geográfico, sendo cada classe do esquema armazenada em forma de tabela geográfica, com cada campo “codigo” participando da chave primária da tabela geográfica e os campos “codigo_bacia” e “codigo_subbacia” sendo definidos como chaves estrangeiras. No esquema conceitual do SOG2, a estrutura está agrupada em uma única classe. As bacias e sub-bacias hidrográficas são representadas apenas pela descrição do nome e os nomes das propriedades têm limitação de oito caracteres apenas. Este esquema conceitual assemelha-se aos utilizados em um arquivo de SIG (por exemplo, *shapefile*), onde os objetos geográficos são armazenados juntamente com os atributos em forma de tabelas.

Pode-se perceber que existe uma grande heterogeneidade sintática e semântica entre os dois esquemas conceituais, onde classes e propriedades com o mesmo significado são representadas por nomes diferentes, e as estruturas de armazenamento são completamente diferentes entre as duas fontes de dados.

O esquema conceitual de referência que serve de base para definir a estrutura sintático-semântica padrão que será utilizada pelos objetos geográficos daquele domínio, representado pela ontologia, pode ser exatamente igual a um dos dois esquemas conceituais ou ser completamente diferente. É esperado que a ontologia de referência seja mais detalhada, ou, ter pelo menos o mesmo nível de detalhes que o esquema conceitual mais detalhado dentre os esquemas disponíveis. A seguir, é apresentada uma ontologia de referência de exemplo que poderia ser utilizada para definir o domínio da hidrografia:

Classe: Bacia_Hidrografica

Atributos: codigo – código identificador do objeto;

geometria - geometria do objeto geográfico (ponto, linha, polígono...);

nome – nome da bacia hidrográfica;

area – área da bacia hidrográfica;

Classe: Sub-bacia_Hidrografica

Sub-classe de: Bacia_Hidrografica

Classe: Hidrografia

Sub-classe de: Sub-Bacia_Hidrografica

Atributos: hidrografia_associada – relaciona o objeto de hidrografia ao qual ele está associado (por exemplo: afluente do rio madeira);

Classe: Rio

Sub-classe de: Hidrografia

Atributos: extensao – extensão do rio;

volume – volume médio anual de água;

largura – largura média do rio;

profundidade – profundidade média do rio;

navegabilidade – indica se o rio é navegável;

Classe: Corpo_dagua

Sub-classe de: Hidrografia

Atributos: area – área do corpo d' água;

Classe: Lago

Sub-classe de: Corpo_dagua

Classe: Lagoa

Sub-classe de: Corpo_dagua

Pode-se perceber que o esquema conceitual definido pela ontologia de referência é mais detalhado que os esquemas conceituais das fontes de dados. Além disso, possuem uma estrutura distinta dos demais, muito semelhante ao paradigma da Análise Orientada a Objetos (OO).

Sendo assim, pode-se encontrar um ambiente tão heterogêneo que todas as fontes de dados possuem esquemas conceituais diferentes entre si, e o esquema de referência determinado pelo processo de engenharia do conhecimento (item 3.3) é, da mesma maneira, diferente dos demais esquemas encontrados nos serviços WFS fornecidos pelos servidores de objetos geográficos.

A proposta de criação da ontologia do domínio tem como objetivo fundamental fornecer um referencial sintático e semântico para as demais fontes de dados. Isto não significa que as fontes de dados devem se adaptar ao esquema conceitual de referência criado, modificando os seus esquemas conceituais já consolidados. Os esquemas conceituais de cada fonte atendem à uma comunidade de usuários específica com uma cultura particular e que possui uma visão diferente da realidade geográfica, já que a natureza é complexa e leva a percepções distintas (5).

Então, como realizar esta integração entre os esquemas conceituais das fontes de dados e o esquema conceitual de referência? Propõe-se criar um mecanismo de correlação entre os esquemas conceituais, ou seja, os serviços WFS que forem publicados no servidor de integração devem informar a correlação entre o esquema conceitual que rege este serviço e o esquema conceitual de referência representado pela ontologia do domínio. Isto deve ser feito para cada elemento do esquema de referência da fonte de dados que esteja sendo publicado no servidor de integração. Utilizando como exemplo a situação apresentada para o domínio da hidrografia, ao publicar o serviço WFS da fonte de dados SOG2 no servidor de integração, o responsável pela publicação deverá informar a correlação entre os elementos nele presentes e os elementos que constam no esquema de referência. A seguir é apresentada a correlação entre os dois esquemas conceituais na tabela 3.1:

Esquema Conceitual do SOG2	Esquema Conceitual de Referência
OID	Hidrografia.codigo
GEOMETRY	Hidrografia.geometria
BACIA	Bacia_Hidrografica.nome
SUBBACIA	Sub-Bacia_Hidrografica.nome
NOME	Hidrografia.nome
TIPO	
LARGURA	Rio.largura
PROFUND	Rio.profundidade

Tabela 3.1 Correlação sintático-semântica entre os esquemas conceituais do servidor SOG2 e o esquema de referência

Nota-se que o elemento “TIPO” não possui nenhuma correlação com a ontologia de referência, portanto ele é ignorado. Por estar lidando com estruturas heterogêneas, alguns elementos encontrados no esquema conceitual das fontes de dados não farão sentido na ontologia e serão ignorados.

Como as fontes de dados fornecem os seus objetos geográficos utilizando servidores WFS, que por sua vez tornam-nos disponíveis em formato GML, o esquema conceitual de cada fonte de dados é representado por um esquema XML (*XMLSchema*), denominado esquema de aplicação GML (*GML application schema*). Este esquema XML é escrito de acordo com as regras definidas pela especificação GML para representar o vocabulário dos objetos geográficos para uma aplicação em particular (7), no caso, o serviço WFS oferecido. Por definir a sintaxe dos objetos geográficos disponíveis na fonte de dados, este *XMLSchema* deve ser utilizado para fazer a correlação com a ontologia de referência.

O processo de engenharia do conhecimento (item 3.3) gera como resultado uma ontologia do domínio tanto em formato de documentação HTML quanto em formato OWL. Desta forma, os elementos da ontologia podem ser lidos a partir do arquivo OWL criado, e colocados à disposição para serem correlacionados aos elementos do esquema XML disponível no serviço WFS. A documentação da ontologia em HTML deve estar publicada no servidor de integração para que prováveis dúvidas quanto ao significado de cada elemento da ontologia sejam esclarecidas.

A ação de correlacionar os elementos dos esquemas conceituais, deve ser realizada utilizando-se uma interface de cadastramento no servidor de integração. Posteriormente, esta informação será utilizada para a unificação das informações geográficas distribuídas e para a realização de consultas a partir da ontologia de referência.

Natureza da informação

Um aspecto importante da correlação é a natureza das informações de cada elemento do esquema. As informações podem possuir uma natureza quantitativa ou qualitativa, que geralmente é definida pelo tipo do dado atribuído ao elemento do esquema. Uma informação quantitativa apresenta aspectos quantificáveis relacionados à feição geográfica como área, distância, profundidade, volume, um código, etc. As informações qualitativas apresentam descrições que definem um objeto geográfico. Tipos de dados numéricos, data/hora, etc. definem uma natureza quantitativa às informações enquanto tipos alfanuméricos definem uma natureza qualitativa.

A correlação deve levar em consideração este fator, tanto no momento de elaborar a tabela de correlação quanto no momento de integrar as informações. Alguns tipos de dados não são passíveis de correlação com outros e estes aspectos devem ser respeitados no momento da elaboração da tabela. Um tipo numérico não poderá se correlacionar com uma data e vice-versa, assim como uma informação qualitativa não poderá ser convertida em uma informação qualitativa automaticamente.

No caso das informações qualitativas, deve-se considerar a utilização de algoritmos de definição de métricas de similaridade. A utilização de métodos de medida de distância entre cadeias de caracteres como Hamming distance, Levenshtein distance, Needleman-Wunch distance, Smith-Waterman distance, Monge-Elkan distance, Jaro-Winkler (9), etc. permite a integração de informações qualitativas semelhantes.

Entretanto, no caso de informações qualitativas, podem existir casos em que nem mesmo o uso de algoritmos de similaridade seja suficiente para que a integração dos dados seja realizada. Nestes casos, deve-se construir também uma tabela de correlação de dados que irá correlacionar um dado de um servidor de objetos geográficos com um indivíduo da ontologia.

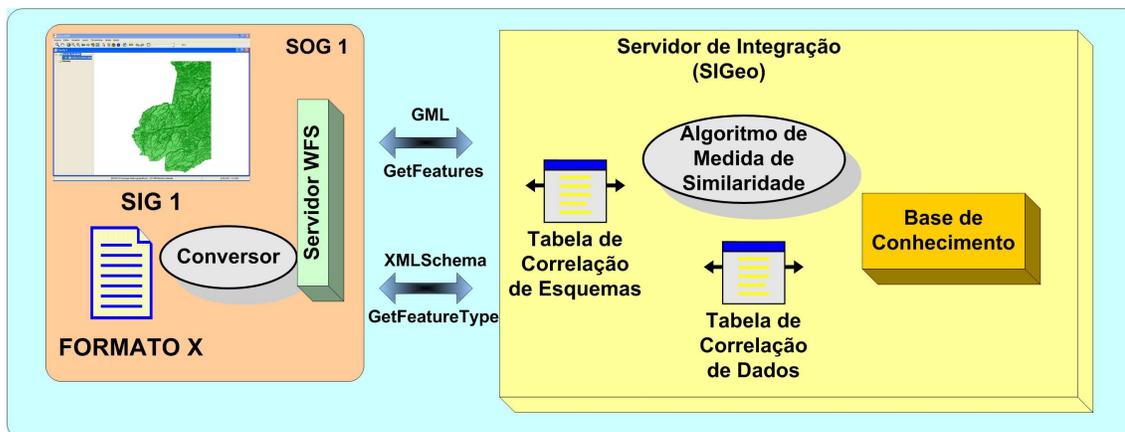


Figura 3.7 Correlação sintática e semântica

Desta maneira, a configuração do servidor de integração é realizada através do cadastramento da localização dos serviços WFS disponíveis em cada fonte de dados na rede juntamente com as informações sobre os serviços em forma de metadados e a correlação sintático-semântica entre os esquemas conceituais. Com esta configuração realizada, o servidor de integração poderá realizar de fato a unificação dos objetos geográficos disponíveis em cada uma das fontes de dados nele cadastradas.

3.5 Publicação dos objetos geográficos integrados

Com os objetos geográficos já disponíveis e o servidor de integração devidamente configurado, o próximo passo da metodologia é realizar de fato a unificação dos objetos que se encontram disponíveis através de serviços WFS. Para isto, devem ser novamente utilizados os serviços do consórcio OGC e um mecanismo automatizado de geração da informação unificada em formato padrão GML.

Os objetos geográficos unificados devem ser disponibilizados em formato GML por meio de serviços WFS, assim como foi feito para publicar as informações geográficas de cada fonte de dados. O serviço WFS permitirá que os objetos geográficos unificados funcionem como uma nova fonte de dados.

Além da publicação em formato GML, uma representação visual dos dados geográficos unificados deve ser fornecida utilizando o *Web Map Service* (WMS). O serviço WMS viabilizará a análise e manipulação do mapa gerado a partir dos objetos geográficos unificados.

Para realizar a publicação destes serviços, primeiramente é necessário que se crie o esquema XML de referência que fornecerá a estrutura sintático-semântica a ser utilizada no

retorno do serviço em GML. Este é o primeiro passo da unificação dos dados, ou seja, um esquema conceitual único para todos os dados disponíveis nas fontes de dados cadastradas. Este esquema, de fato, já existe. Ele é o próprio esquema de referência da correlação sintático-semântica realizada da etapa anterior, ou seja, a ontologia do domínio. Entretanto, pelo fato da ontologia encontrar-se em formato OWL, e não em *XMLSchema* (10), a conversão de um para o outro deve ser realizada. Este esquema conceitual gerado em *XMLSchema* deve ser retornado ao requisitante do serviço WFS quando a função *describeFeatureType* for requisitada.

Os objetos geográficos, que em suas fontes de dados de origem estão em esquemas conceituais particulares, devem ser fornecidos aos requisitantes do serviço seguindo o esquema conceitual de referência. Ao unificar os objetos geográficos, o servidor de integração deve adequar a sua estrutura para o novo esquema conceitual de referência criado em *XMLSchema*. Para isto, o servidor deve utilizar as informações de correlação entre esquemas conceituais previamente armazenadas na fase de configuração (item 3.4.2). O serviço WFS deve fornecer todos os objetos geográficos distribuídos nas fontes de dados configuradas no servidor de integração. Quando o cliente requisitar a função *getFeature*, o serviço WFS deve buscar os objetos geográficos armazenados nas fontes de dados cadastradas e adequá-los ao esquema conceitual de referência utilizando a tabela de correlação entre esquemas conceituais armazenada em XML. Estas informações espaciais devem ser retornadas ao requisitante do serviço como se estivessem em uma única fonte de dados, com um mesmo esquema conceitual em *XMLSchema*.

Os objetos geográficos são fornecidos aos requisitantes do serviço WFS em formato GML e podem ser manipulados utilizando outros tipos de software de geoprocessamento, bastando para isto que os dados sejam convertidos de GML, que é o formato padrão definido pelo consórcio OGC, para o formato de trabalho do SIG que se está utilizando. Esta conversão pode ser feita previamente pelo usuário, ou seja, antes de utilizar o SIG, ou automaticamente por um SIG que interprete WFS ou GML.

Enquanto o serviço WFS do servidor de integração fornece os dados espaciais unificados em formato GML, o serviço WMS deve fornecer um mapa com a representação visual destes objetos geográficos. Segundo a especificação WMS (27), um mapa não é formado propriamente pelos dados geográficos, ele é a representação destes dados em formato digital para ser apresentada em uma tela de computador. As informações espaciais são convertidas para formatos digitais de imagens como JPG, GIF, PNG e etc. e apresentados ao requisitante do serviço WMS.

O servidor de integração deve responder ao requerimento da função *getMap* do serviço WMS com um mapa contendo a representação visual de todos os objetos geográficos das fontes de dados cadastradas. Para o requisitante do serviço, é transparente o fato de existirem várias fontes de dados heterogêneas já que o serviço retorna um mapa como se os dados estivessem armazenados em uma fonte de dados apenas. O mapa retornado na função *getMap* não possui informações detalhadas dos objetos geográficos, porém, o usuário requisitante poderá utilizar a função *getFeatureInfo* do serviço WMS para recuperar as informações sobre um objeto particular que pertence ao mapa.

Tanto o serviço de WFS quanto o de WMS disponíveis no servidor de integração devem seguir integralmente as respectivas especificações elaboradas pelo consórcio OGC. Ambos podem ser utilizados a partir de um *browser web*, ou em SIGs que possuam uma interface para utilização dos serviços OGC.

Cada domínio cadastrado no servidor de integração deve fornecer um serviço WFS específico para aquele domínio, tornando disponíveis todos os objetos geográficos das fontes de dados relacionadas a ele, sob o esquema de referência definido pela ontologia de referência do domínio. O serviço WMS, pelo contrário, deve agrupar todos os domínios, sendo cada um considerado como uma camada ou plano de informação. A figura 3.9 resume esta proposta metodológica.

Através da publicação dos serviços WFS e WMS, as informações espaciais oriundas de diversas fontes de dados heterogêneas tornam-se disponíveis de maneira unificada, resultando em um mecanismo que fornece um mesmo formato, estrutura, sintaxe e semântica para todos os dados espaciais disponíveis nestas fontes de dados geográficos. Sendo assim, a metodologia promove a integração dos objetos geográficos heterogêneos distribuídos em diversas fontes de dados institucionais, como será demonstrado no capítulo 4 onde será apresentado o estudo de caso.

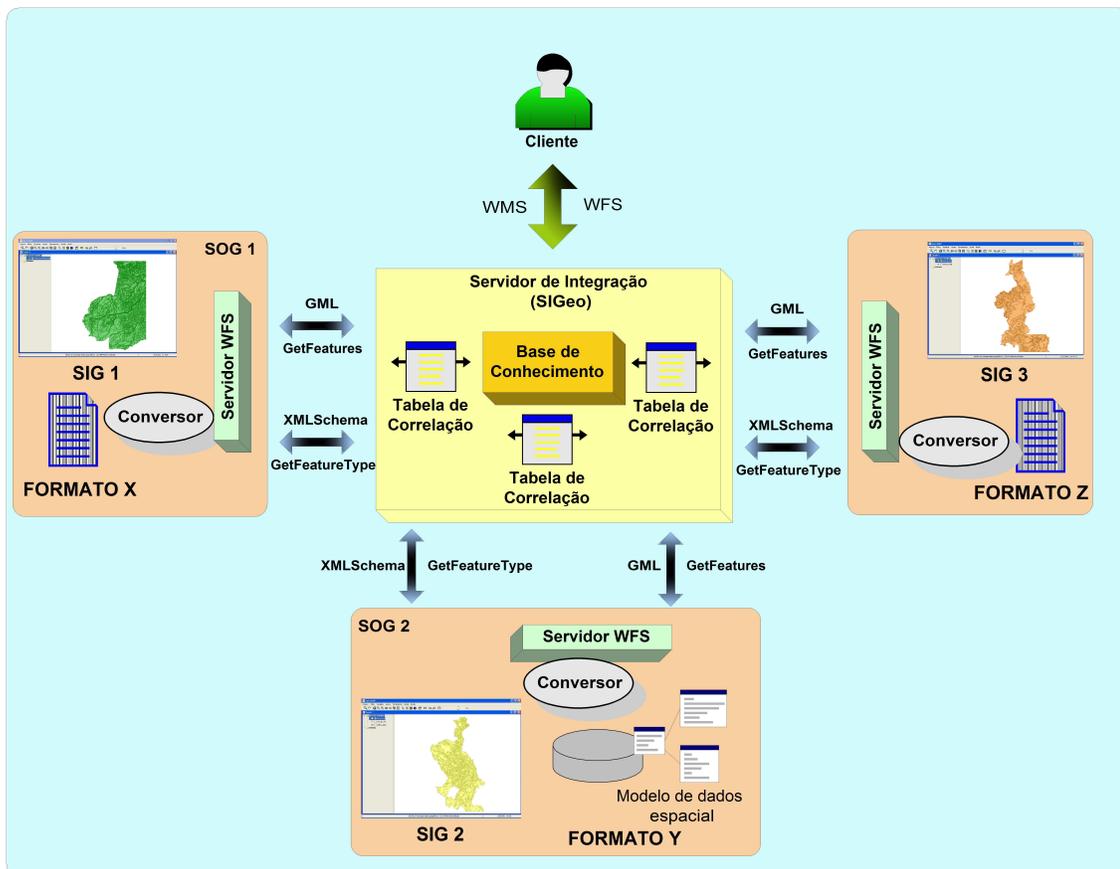


Figura 3.8 Proposta de interoperabilidade entre objetos geográficos heterogêneos distribuídos

4 Estudo de Caso

Com a finalidade de validar as hipóteses levantadas, e, com o objetivo de comprovar a eficiência da metodologia proposta, realizou-se um estudo de caso no domínio de atributos dos solos para análise da aptidão agrícola das terras, no âmbito do projeto de zoneamento agro-ecológico para culturas oleaginosas com ênfase no cultivo do dendê, denominado Zon-dende, realizado pelo Centro Nacional de Pesquisa de Solos - CNPS, da Empresa Brasileira de Pesquisa Agropecuária (EMBRAPA), na região amazônica. O projeto Zon-dende tem como objetivo realizar o zoneamento agro-ecológico visando conhecer o potencial da produção sustentada de culturas oleaginosas nas áreas desmatadas da região amazônica, com ênfase para o cultivo do dendê para produção de biodiesel, como parte do programa nacional de biocombustível do governo federal brasileiro.

O projeto Zon-dende utiliza como base, as informações obtidas pela Embrapa, Projeto Radambrasil (28) e outras fontes de dados públicas e privadas. A base de dados do projeto conta ainda com a base de recursos naturais do Sistema de Proteção da Amazônia (SIPAM), além da criação de novos mapas de solos gerados pela Embrapa a partir dos mapas já existentes em escalas mais detalhadas e em áreas de maior interesse. Conforme a necessidade, dados de instituições locais ou regionais também podem ser incorporados ao projeto.

A avaliação de aptidão dos solos é feita com base na interpretação dos dados das unidades de mapeamento de solos, através da avaliação dos atributos considerados e das exigências da cultura avaliada. Sendo assim, existe a necessidade real de integração das diversas fontes de dados disponíveis para se alcançar o objetivo esperado.

Neste estudo de caso, duas fontes de dados foram escolhidas para realizar a integração sintática, semântica e estrutural das informações geográficas de solos da região amazônica: base de recursos naturais do Sistema de Proteção da Amazônia (SIPAM) e as informações georeferenciadas de recursos naturais do Instituto Brasileiro de Geografia e Estatística (IBGE). Primeiramente, preparou-se um ambiente de simulação com a presença das duas fontes de dados mencionadas e posteriormente, foi aplicada a metodologia proposta em uma tentativa de verificar os resultados obtidos. Nos próximos capítulos são detalhadas as etapas realizadas no estudo de caso.

4.1 Fontes de Dados

4.1.1 A fonte de dados de solos do SIPAM

As informações de solos do SIPAM estão estruturadas em uma arquitetura dual, onde os dados espaciais e os dados não espaciais são armazenados em locais diferentes com tecnologias distintas. Os dados espaciais estão armazenados no formato *shapefile* (<http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>) e podem ser manipulados utilizando qualquer SIG que interprete este formato. Os dados não-espaciais, pelo contrário, estão armazenados seguindo um modelo conceitual bem definido, em uma estrutura de banco de dados relacional utilizando o Sistema Gerenciador de Banco de Dados (SGBD) Relacional da *Oracle* (<http://www.oracle.com/database/index.html>).

O modelo de dados do SIPAM contém somente os dados não-espaciais. Os objetos espaciais da fonte de dados de solos do SIPAM são apresentados a seguir no *ArcGIS* como mostram as figuras 4.1 e 4.2, respectivamente, o mapa de solos e a sua tabela de atributos.

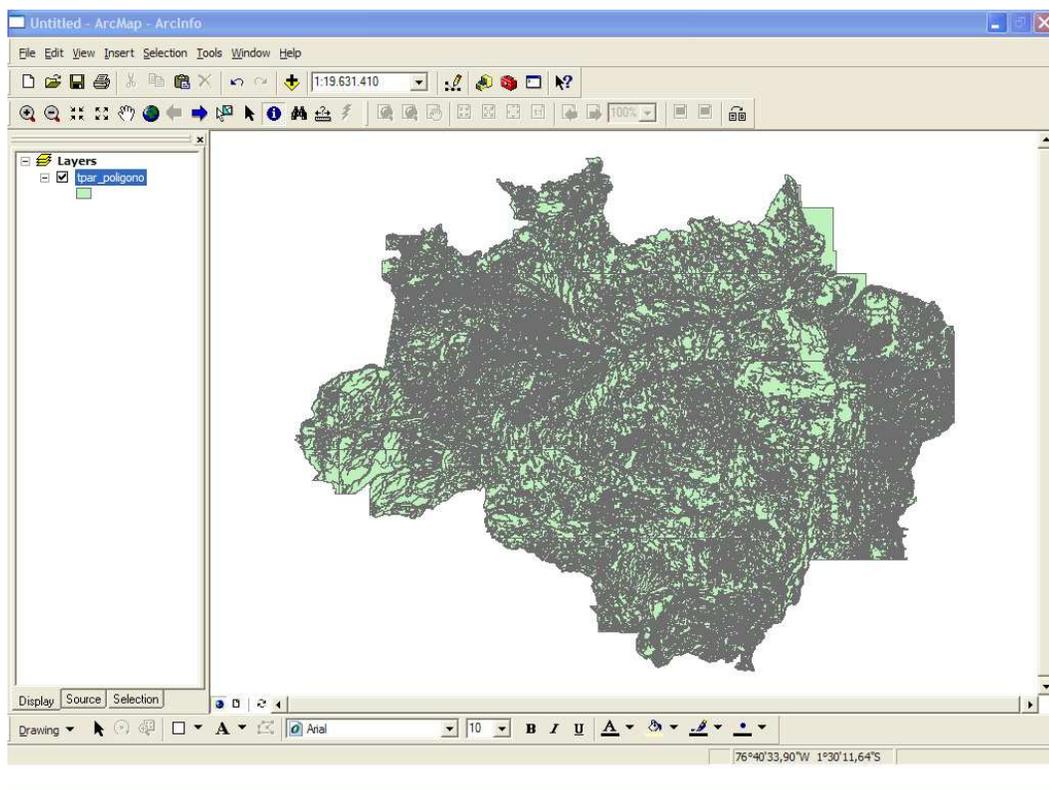


Figura 4.1 Mapa de solos do SIPAM apresentada no ArcMap (ESRI)

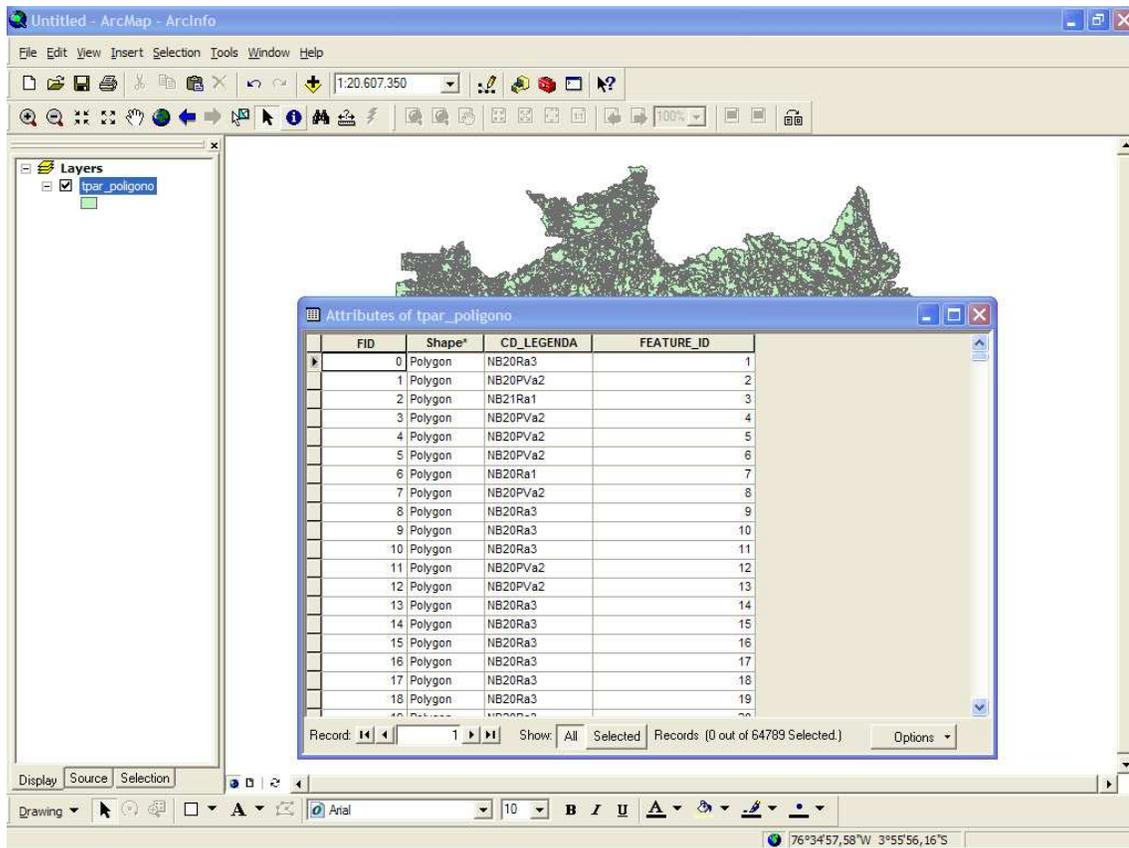


Figura 4.2 Tabela de atributos do mapa de solos do SIPAM

Nota-se que a tabela de atributos do mapa de solos possui uma coluna chamada CD_LEGENDA que faz o relacionamento entre os dados espaciais e não-espaciais, já que este campo faz referência à coluna CD_SIMBOLO_UNID da tabela TP_CAT_UNID_SOLO do banco de dados relacional (figura 4.3). A tabela TP_CAT_UNID_SOLO é a tabela que armazena os dados de categoria de unidade de mapeamento, enquanto a sua chave primária, a coluna CD_SIMBOLO_UNID, representa o símbolo da unidade de mapeamento (figura 4.3).

Para facilitar a publicação dos dados espaciais em serviços WFS, optou-se por modificar a estrutura de armazenamento dos dados, de uma arquitetura dual para um banco de dados espacial, incorporando as tabelas espaciais ao esquema conceitual do banco de dados do SIPAM. O SGBD Relacional *Postgresql* (<http://www.postgresql.org/>) com a extensão espacial *Postgis* (<http://postgis.refractor.net/>) foi escolhido para armazenar a nova estrutura, por ser um software livre e atender às necessidades do estudo de caso. Foram criadas as tabelas espaciais TPAR_Poligono e TPPT_Ponto que armazenam, respectivamente, as informações espaciais de polígonos de solos e pontos de coleta de informações de solos. A figura 4.3 demonstra como estas tabelas se relacionam com as outras tabelas do modelo relacional.

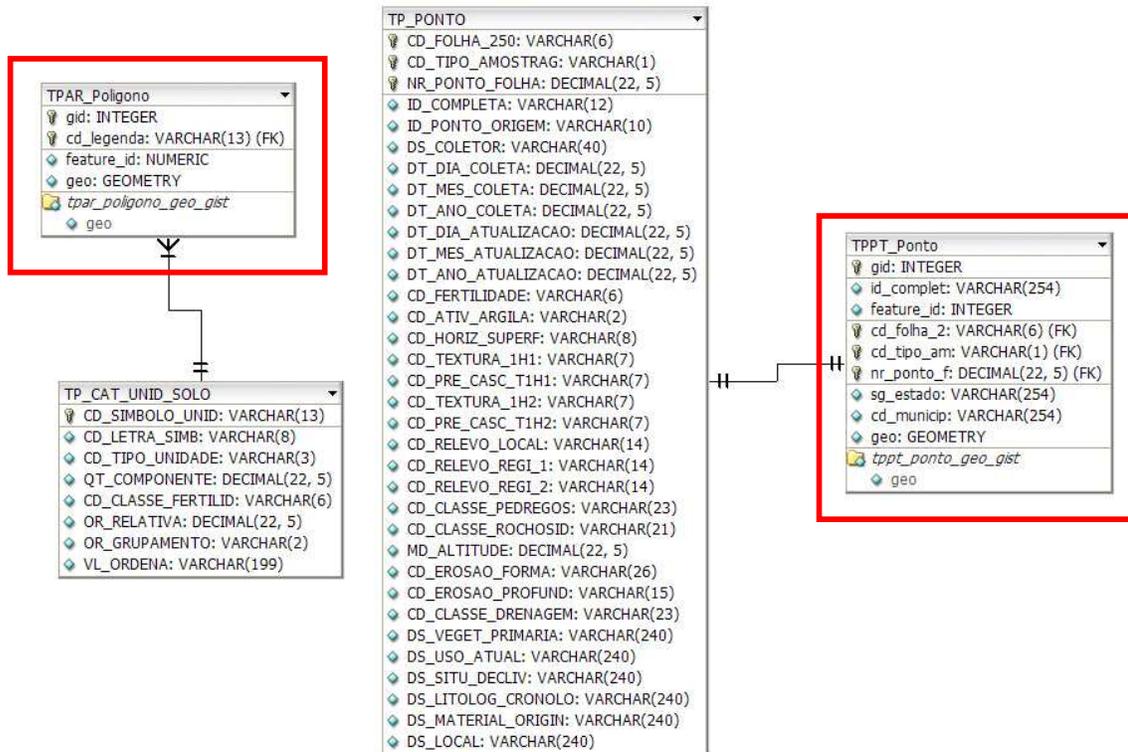


Figura 4.3 Relacionamento entre as tabelas espaciais e as tabelas do modelo relacional.

A partir das alterações realizadas no banco de dados de solos do SIPAM, criou-se a primeira fonte de dados do ambiente de simulação representada em uma estrutura de banco de dados espacial.

4.1.2 A fonte de dados de solos do IBGE

O IBGE vem realizando uma atividade permanente de sistematização das informações de recursos naturais. Segundo IBGE (28), o objetivo básico deste projeto é a produção de informações atualizadas sobre os recursos naturais em todo o território nacional, nas áreas de geologia, geomorfologia, solos e vegetação, utilizando uma metodologia padronizada. O IBGE absorveu o projeto RADAMBRASIL em 1986 e com o seu conteúdo iniciou a sistematização das informações de recursos naturais. Hoje em dia, este material encontra-se representado na escala de 1:250.000 e foi praticamente todo publicado na escala de 1:1.000.000.

Atualmente, este mapeamento é atualizado periodicamente por informações secundárias de outras instituições, por dados interpretados de imagens de sensores remotos ou por levantamentos de campo. O material vem sendo gerado para meio digital e

disponibilizado para a sociedade através do site http://www.ibge.gov.br/servidor_arquivos_geo. No banco de dados de recursos naturais, as informações que compreendem a região da Amazônia Legal estão parcialmente compatibilizadas, atualizadas e publicadas.

O banco de dados armazena informações gráficas e alfanuméricas atualizadas de solos com nível de detalhamento compatível com a escala de 1:250.000. Todas estas informações estão disponíveis para *download* no site do IBGE, sendo as informações espaciais representadas na forma de arquivos em formato *shapefile* e os dados alfanuméricos armazenados em arquivos no formato *dbf* (*database file* - <http://www.dbase.com/>), contendo os metadados devidamente documentados. Atualmente, os dados disponíveis correspondem às seguintes cartas ao milionésimo: Pico da Neblina (NA-19), Içá (SA-19), Javari (SB-18), Juruá (SB-19), Contamana (SC-18), Rio Branco (SC-19). A figura 4.4 apresenta o mapa índice ao milionésimo das cartas do IBGE.

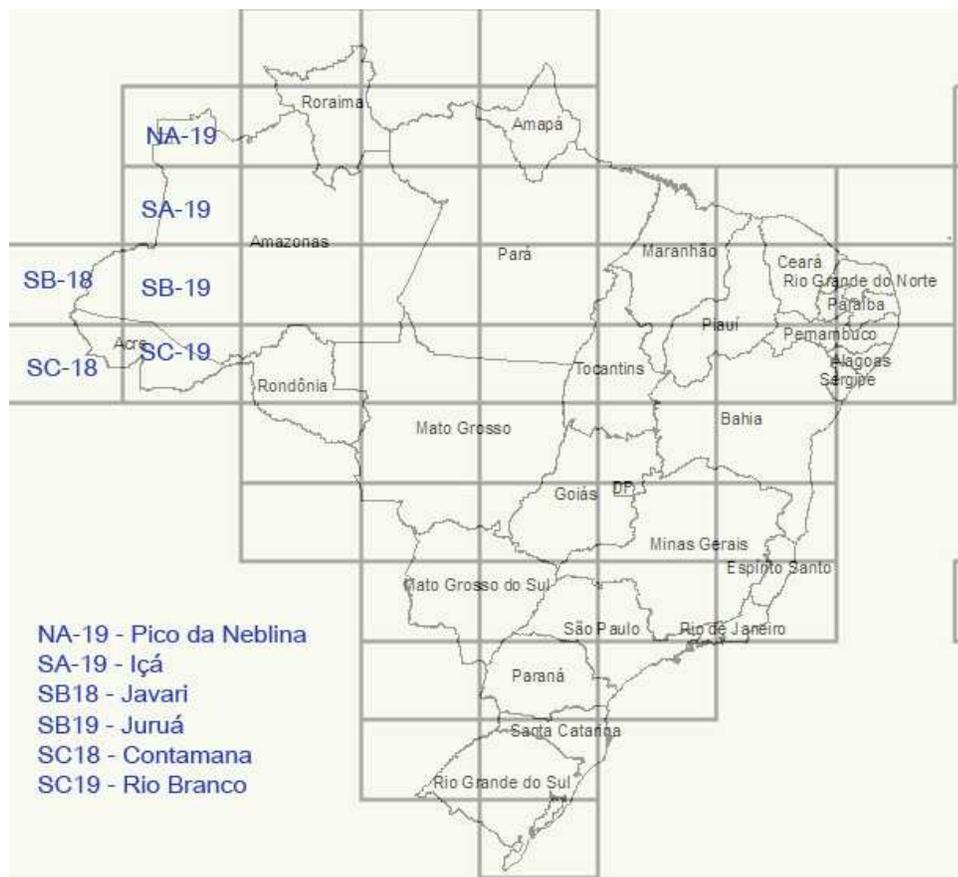


Figura 4.4 Mapa índice das cartas do IBGE (1:1.000.000)

A localização das cartas disponíveis para *download* corresponde à região de fronteira com países vizinhos, sendo assim, atendem aos objetivos do projeto de zoneamento agroecológico do dendê que enfatiza o seu estudo em regiões de fronteira (16).

Das cartas disponíveis para *download*, cinco possuem o mesmo esquema conceitual, ou seja, possuem a mesma especificação para os atributos, enquanto uma, a carta NA-19 (Pico da Neblina), possui um esquema conceitual diferente dos demais, o que caracterizaria uma nova fonte de dados, portanto, não será utilizada no estudo de caso.

As cartas utilizadas foram trabalhadas para que possuíssem somente os dados necessários para os cálculos de aptidão agrícola para o cultivo do dendê. Os campos irrelevantes foram excluídos e as cartas foram unidas em um único arquivo *shapefile* com a estrutura apresentada na tabela 4.1 para facilitar a publicação dos dados no servidor WFS.

Distribuição dos tipos de solo na folha SC.19 (escala 1:1.000.000) – Projeção Geográfica – Datum SAD-69			
Descrição dos campos da tabela gráfica			
Nome	Descrição	Tipo	Tamanho
SHAPE	Tipo de feição gráfica	Shapepoly	8
LETRA_SIMB	Letra-símbolo da unidade de mapeamento	Char	8
ORDEM	Ordem da unidade	Char	20
SUB_ORDEM	Subordem da unidade	Char	30
GDE_GRUPO	Grande grupo da unidade	Char	20
LEGENDA	Ordem e Subordem da classe de solo	Char	50
SIMB_UNID	Letra-símbolo de carga da unidade de mapeamento	Texto	13
SUBST_ROCH	Fase de Substrato Rochoso	Texto	25
SUB_GRUPO	Sub grupo	Texto	255
GRUP_TEXT	Grupamento de Textura	Texto	255
HOR_SUPERF	Horizonte Superficial	Texto	255
SAT_ALUM	Saturação por Alumínio	Texto	255
FS_VEGET	Fase de Vegetação	Texto	255
FS_RELEVO	Fase de Relevô	Texto	255
FS_PEDREG	Fase de Pedregosidade	Texto	255
FS_ROCH	Fase de Rochosidade	Texto	255
FS_EROSAO	Fase de Erosão	Texto	255

Tabela 4.1 Esquema conceitual da fonte de dados do IBGE

A figura 4.5 apresenta o mapa de solos que representa a região de fronteira com as cartas do IBGE unificadas em um mesmo arquivo *shapefile*, seguindo o esquema conceitual da tabela 4.1.

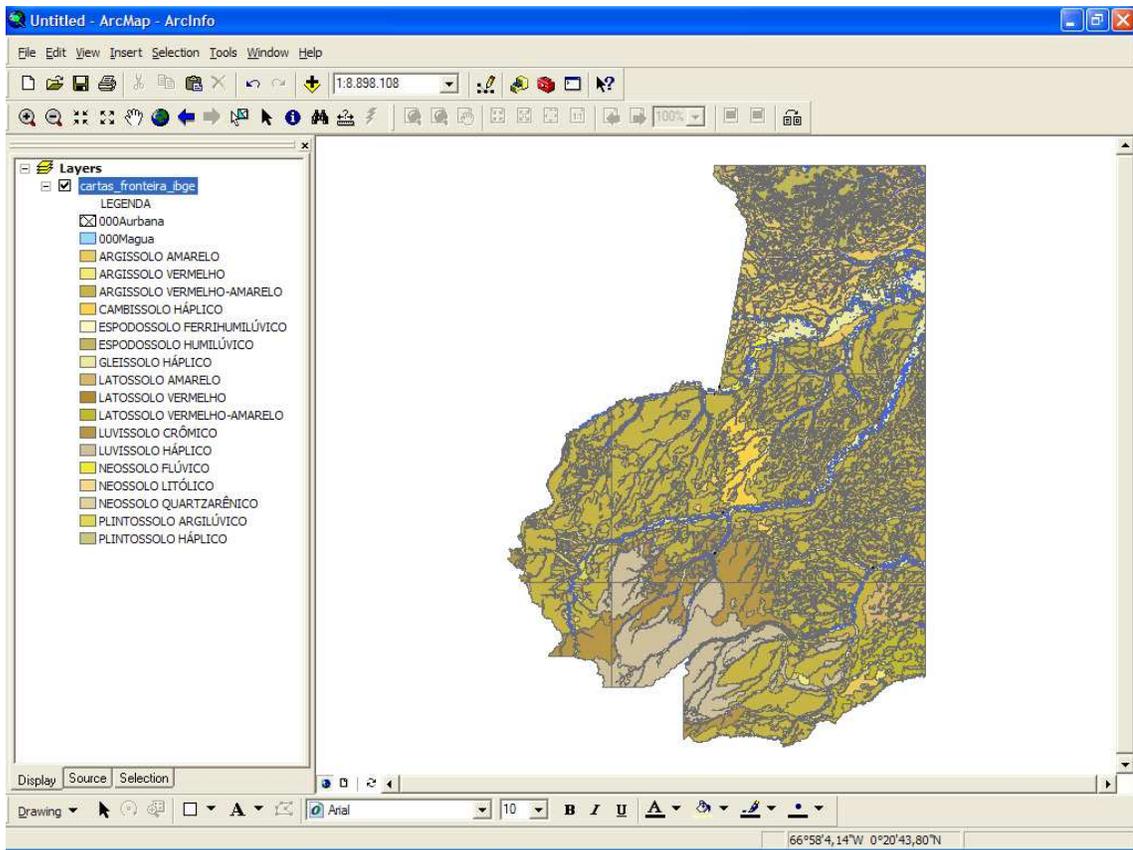


Figura 4.5 Mapa de solos do IBGE.

Desta maneira, criou-se o ambiente para a aplicação da metodologia com duas fontes de dados de instituições diferentes, com estruturas de dados, sintaxe e semântica distintas. As duas fontes de dados de solos utilizadas (SIPAM e IBGE) apresentam heterogeneidades em seus esquemas conceituais, tanto em termos de formato de dados (*postgresql/postgis* e *shapefile*) quanto a nível sintático e semântico. A Tabela 4.2 exemplifica estas diferenças. As duas primeiras linhas apresentam heterogeneidade sintática apenas, enquanto as três últimas linhas da tabela apresentam heterogeneidade sintática e semântica. Na última linha da tabela 4.2, por exemplo, a descrição semântica do atributo não deixa claro para um leigo que se trata de atributos com uma forte correlação.

Heterogeneidade	SIPAM	IBGE
Formato	Banco de Dados Espacial Postgis	Arquivo <i>ESRI Shapefile</i>
Sintática	Geo	SHAPE
Semântica	Polígono de unidade de mapeamento de solo na tabela espacial.	Polígono de unidade de mapeamento de solo no shapefile.
Sintática	cd_letra_simb	SIMB_UNID
Semântica	Símbolo da Unidade de Mapeamento.	Símbolo da Unidade de Mapeamento.
Sintática	cd_textura_1h1	GRUP_TEXT
Semântica	Textura do Horizonte Superficial	Grupamento de Textura
Sintática	cd_relevo_1	FS_RELEVO
Semântica	Relevo Predominante	Fase de Relevo
Sintática	cd_pre_casc_1_t1h1	FS_PEDREG
Semântica	Presença de cascalho no horizonte Superficial	Fase de Pedregosidade

Tabela 4.2 Exemplos de heterogeneidades encontradas nas bases de dados de solos

4.2 Aplicação da Metodologia

4.2.1 Definição da arquitetura utilizada

A arquitetura utilizada neste estudo de caso é semelhante a apresentada na figura 3.2, com apenas um servidor de integração para todas as fontes de dados e um servidor de objetos geográficos para cada fonte.

O primeiro servidor de objetos geográficos, denominado SOG-SIPAM, é responsável por fornecer os dados geográficos da fonte de dados do SIPAM, que contém os dados de solos de toda a Amazônia Legal em um banco de dados geográfico. Apesar de conter objetos geográficos de toda a Amazônia, no estudo de caso só foram utilizados os dados da carta SC18 (Contamana) do IBGE.

O segundo servidor, denominado SOG-IBGE, fornece os objetos geográficos da fonte de dados do IBGE. Esta fonte contém os dados de solos da região de fronteira da Amazônia com os países vizinhos, porém somente a carta SB18 (Javari) foi utilizada neste estudo.

O servidor de integração será responsável por realizar a integração entre as duas fontes de dados que compõem o ambiente de simulação. Este servidor utilizará os serviços WFS disponíveis nos servidores SOG-SIPAM e SOG-IBGE.

A seguir, a figura 4.6 apresenta o diagrama de implantação da arquitetura proposta para realizar a integração dos objetos geográficos fornecidos pelas fontes de dados apresentadas.

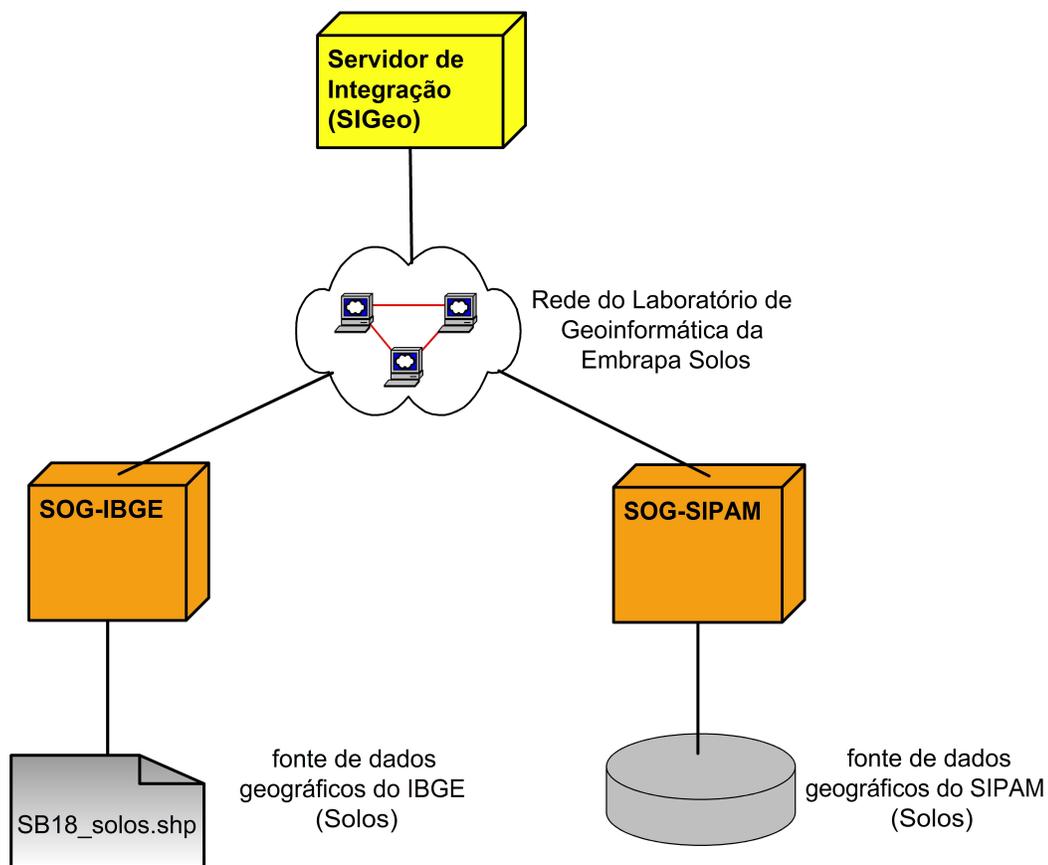


Figura 4.6 Arquitetura do estudo de caso

4.2.2 Configuração dos servidores

Com a arquitetura definida e os servidores de objetos geográficos determinados, foi feita a preparação dos servidores para que eles forneçam os dados geográficos através de serviços WFS. Cada servidor fornece o serviço WFS que permitirá o acesso aos objetos geográficos no formato GML.

Para fornecer os serviços utilizou-se a ferramenta *geoserver* (29). O *geoserver* é um servidor WFS e WMS de código aberto compatível com as especificações do consórcio OGC. Através do *geoserver* é possível publicar os dados geográficos utilizando os serviços WFS e WMS do consórcio OGC apenas realizando algumas configurações na ferramenta.

Os dados publicados em serviços WFS serão as informações de solos relevantes para o processo de definição da aptidão agrícola dos solos para o cultivo do dendê. A seguir, são

apresentadas as ações realizadas em cada servidor para que os objetos geográficos fossem publicados.

Configuração do SOG-SIPAM

Antes de publicar os dados armazenados na fonte de dados do SIPAM, criou-se uma visão no banco de dados espacial com as informações relevantes para a análise da aptidão agrícola para o cultivo do dendê. A tabela 4.2 apresenta a relação de alguns campos retornados pela visão criada no banco de dados espacial, e que foi utilizada para publicar os dados no servidor WFS.

Nome	Descrição	Tabela	Formato
feature_id	Código do objeto geográfico	tpar_poligono	NUM(20)
Geo	Geometria do objeto geográfico	tpar_poligono	MULTIPOLYGON
cd_simbolo_unid	Letra-símbolo de carga da unidade de mapeamento	tp_cat_unid_solo	CHAR(8)
cd_letra_simb	Letra-símbolo da unidade de mapeamento	tp_cat_unid_solo	CHAR(3)
cd_tipo_unidade	Tipo de unidade de mapeamento	tp_cat_unid_solo	CHAR(3)
cd_classe_fertilid	Classe de fertilidade	tp_cat_unid_solo	CHAR(6)
cd_classe_solo	Símbolo da classe de solo / tipo de terreno	tp_componente	CHAR(5)
nm_classe_solo	Nome da Classe de Solo	tp_ref_classe_solo	CHAR(80)
cd_carater_1	Tipo de caráter 1	tp_carater_co	CHAR(23)
cd_relevo_1	Relevo predominante 1	tp_relevo_co	CHAR(7)
cd_textura_1h1	Textura 1 do 1º horizonte (superficial)	tp_textura_co	CHAR(7)
cd_pre_casc_1_t1h1	Presença de cascalho 1 da textura 1 do 1º horizonte	tp_textura_co	CHAR(7)

Tabela 4.3 Relação de campos da visão utilizada para publicar os dados de solos do SIPAM

No banco de dados do SIPAM existem informações de perfis de solos (ponto) e polígonos de solo (unidades de mapeamento de solo) de toda a região da Amazônia Legal. Somente as informações relacionadas à legenda dos polígonos de solo dos mapas foram utilizadas para a análise da aptidão agrícola. Na visão de banco de dados criada, foram selecionadas somente as unidades de mapeamento que pertencem à carta SC18. Portanto, a visão criada no banco de dados retorna as informações relacionadas à geometria do polígono, classe de solos, fertilidade, relevo e caráter da região de estudo.

Para publicar as informações encontradas na visão criada no banco de dados, é preciso realizar as devidas configurações no servidor WFS. O apêndice A apresenta as configurações realizadas no servidor geoserver para a publicação dos dados espaciais de solos.

Configuração do SOG-IBGE

A configuração do servidor de objetos geográficos que fornece os dados espaciais de solos oferecidos pelo IBGE, denominado SOG-IBGE, é muito semelhante à configuração do SOG-SIPAM. A principal diferença fica por conta do tipo de fonte de dados, já que as informações de solos da fonte de dados do IBGE estão armazenadas em formato *shapefile*. Sendo os objetos espaciais da fonte de dados do IBGE previamente trabalhados e armazenados em um único arquivo *shapefile*, não é necessário fazer nenhum tratamento nos dados antes de publicá-los.

O apêndice A apresenta a configuração de um servidor WFS utilizando o geoserver. A tabela 4.4 apresenta as operações que podem ser realizada no SOG-IBGE e os seus respectivos endereços de acesso (URL).

Operação	URL
getCapabilities	http://<endereço>:<porta>/geoserver/wfs?request=getCapabilities
describeFeatureType	http://<endereço>:<porta>/geoserver/wfs/DescribeFeatureType? typeName=sb18_par
getFeature	http://<endereço>:<porta>/geoserver/wfs?request=getFeature& service=wfs&version=1.0.0&typename=sb18_par

Tabela 4.4 Operações executadas no SOG-IBGE

Com os dois servidores de objetos geográficos configurados e fornecendo as informações necessárias para realizar a integração, pode-se perceber que tanto os esquemas conceituais quanto os dados são providos em um formato padrão possibilitando assim a integração automática dos dados espaciais fornecidos pelos serviços. Porém, os dois esquemas conceituais são distintos, o que caracteriza uma diferença sintática e semântica entre as duas fontes de dados tornando inviável a integração entre elas.

Para realizar a integração completa, ou seja, abrangendo tanto os aspectos estruturais dos dados quanto os aspectos sintáticos e semânticos das informações, é preciso elaborar

ainda um esquema conceitual de referência no âmbito da aptidão agrícola dos solos para o cultivo do dendê, como descrito no item 3.3 da metodologia.

4.2.3 O Processo de Engenharia do Conhecimento – Criação da Ontologia no domínio de aptidão agrícola dos solos para cultivo de dendê

O processo de engenharia do conhecimento, como descrito no item 3.3 da metodologia, visa criar uma base de conhecimento no âmbito da aptidão agrícola dos solos para o cultivo do dendê, com o objetivo de estabelecer um esquema conceitual de referência para ser utilizado na integração sintática e semântica das duas fontes de dados existentes. Esta base de conhecimento é formada por um conjunto de ontologias criadas em formato OWL (14) que se relacionam através do reuso.

A base de conhecimento é formada pelas ontologias de três domínios: atributos de solos para aptidão, aptidão agrícola dos solos e aptidão para o dendê. Na ontologia de atributos de solos para aptidão, são descritas as características do domínio de solos que influenciam na análise da aptidão agrícola, suas classes e atributos e o relacionamento entre eles. A ontologia no domínio da aptidão agrícola visa elaborar uma base formal de conhecimento para a técnica de avaliação da aptidão agrícola das terras descrita por Ramalho Filho e Beek (30). Na definição dos termos da aptidão agrícola, os atributos do solo foram reutilizados a partir da ontologia que define este domínio. A ontologia para o domínio de aptidão para o dendê tem como objetivo estabelecer a base de conhecimento formal para avaliar a aptidão dos solos visando cultivo do dendê, reutilizando a base de conhecimento de aptidão agrícola das terras. Nesta ontologia são definidas as regras específicas para definição da aptidão pedológica para a cultura do dendê, visando manter um repositório de informação sobre o assunto e assim poder compartilhar este conhecimento com a comunidade. Outro objetivo da base de conhecimento é poder utilizar as regras definidas para realizar inferências sobre o modelo, e com isso gerar automaticamente a avaliação da aptidão agrícola utilizando uma máquina de inferência.

Atualmente está sendo desenvolvida uma ontologia para classificação de solos para sistemas de zoneamento agrícola no âmbito do agrissolos (31), baseada no Sistema Brasileiro de Classificação de Solos (SIBCS) (32), que poderá ser reutilizada pela ontologia de atributos de solos para aptidão agrícola da base de conhecimentos para o zoneamento do dendê. Objetivando relacionar os atributos de solos com a linguagem GML, foi criada uma ontologia que representa parte da especificação criada pelo consórcio OGC. Este é um aspecto

importante, já que o relacionamento da ontologia do domínio com a ontologia da GML definirá qual elemento do domínio representa o objeto espacial. Para isso, é preciso definir a classe do domínio que representa os objetos espaciais como subclasse da classe *FeaturePropertyType* da ontologia GML, e a propriedade da classe que representa a geometria deve ser do tipo *Geometry* da GML (*MuliPolygon*, *MultiLine*, *Point*). A figura 4.7 apresenta as ontologias propostas e como elas se relacionam.

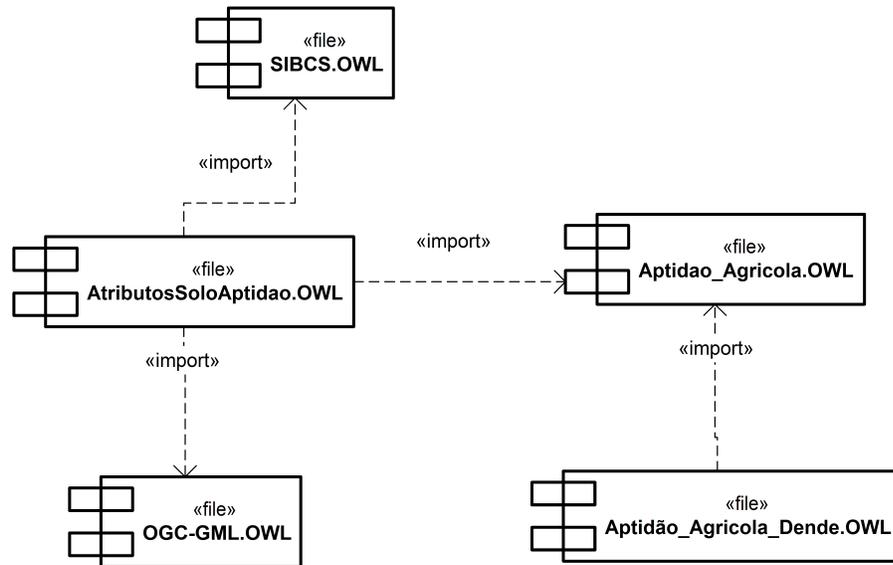


Figura 4.7 Diagrama de componentes da base de conhecimento

A elaboração das ontologias citadas é o resultado do processo de engenharia do conhecimento, que envolve tanto os engenheiros do conhecimento quanto os especialistas nos domínios de solos, agricultura e cultivo do dendê. Algumas tarefas realizadas durante o projeto auxiliaram o processo de modelagem do conhecimento:

- pesquisa bibliográfica à documentação técnica do sistema de banco de dados de solos da Embrapa Solos (SIGSOLOS);
- reuniões de trabalho (*workshops*) com os especialistas em solos da Embrapa Solos;
- reunião técnica com os especialistas em cultivo do dendê;
- reunião técnica com os especialistas em avaliação da aptidão agrícola;
- pesquisa bibliográfica ao sistema de avaliação da aptidão agrícola das terras (30);
- pesquisa bibliográfica aos manuais normativos de levantamento pedológico (33) e classificação de solos (34).

As informações obtidas durante o processo de engenharia do conhecimento foram utilizadas na elaboração da base de conhecimento sobre aptidão agrícola para o cultivo do dendê utilizando o editor de ontologias *protégé* com *plugin* OWL.

4.2.4 Metodologia de Engenharia do Conhecimento

Neste trabalho optou-se por utilizar a metodologia sugerida por Noy e McGuinness (13), que descreve uma abordagem interativa para desenvolver a ontologia. Inicialmente, são apresentadas algumas regras fundamentais que devem ser lembradas durante todo o processo de elaboração do modelo. Elas ajudam a tomar as decisões de elaboração quando surgirem dúvidas durante o processo de modelagem. As regras sugeridas são:

- Não existe uma maneira correta de modelar o domínio - sempre existem alternativas viáveis. A melhor solução quase sempre depende da aplicação e o reuso que se consegue prever;
- O desenvolvimento da ontologia é um processo interativo;
- Os conceitos na ontologia assemelham-se a objetos (físicos e lógicos) e relacionamentos no domínio de interesse. Estes são encarados como nomes (objetos) e verbos (relacionamentos) nas sentenças que descrevem o domínio;
- Após a definição da versão inicial, serão realizados testes para validar a ontologia e discutir a solução com os especialistas no domínio. A revisão da ontologia é inevitável e deve ser feita durante todo o ciclo de vida da base de conhecimento.

A seguir são definidos os sete passos sugeridos na metodologia para a realização do processo de engenharia do conhecimento, adaptados às necessidades encontradas durante a elaboração da base de conhecimento:

- Determinar o domínio e o escopo da ontologia;
- Considerar o reuso de ontologias existentes;
- Enumerar termos importantes na ontologia;
- Definir as classes e a hierarquia de classes;
- Definir as propriedades das classes;
- Criar os indivíduos;
- Definir as regras que definem as inferências que poderão ser realizadas na base de conhecimento a fim de gerar novo conhecimento;

4.2.5 Ontologia de Atributos do Solo para Aptidão Agrícola

A ontologia de atributos do solo para a aptidão modela as características do solo relevantes para a avaliação da aptidão agrícola. Esta ontologia foi utilizada na integração das informações heterogêneas de solo provenientes das fontes de dados distribuídas. Os servidores de objetos geográficos fornecem mais informações sobre solos do que as definidas nesta modelagem, porém, somente as informações da ontologia são relevantes para a avaliação da aptidão agrícola das terras para o dendê.

A partir da revisão bibliográfica realizada sobre mapeamento pedológico (33) e classificação de solos (34) e utilizando um processo iterativo com a participação de especialistas em pedologia, a ontologia foi elaborada utilizando o editor de ontologias Protégé OWL (figura 4.8).

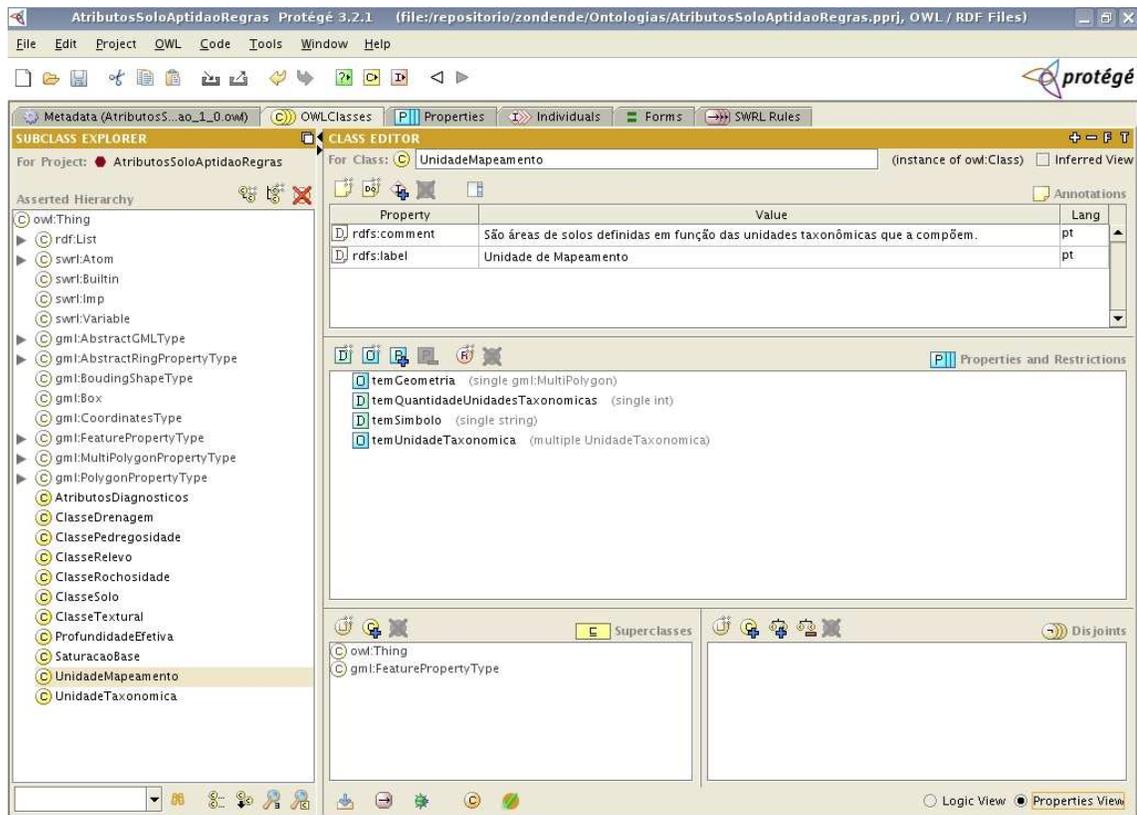
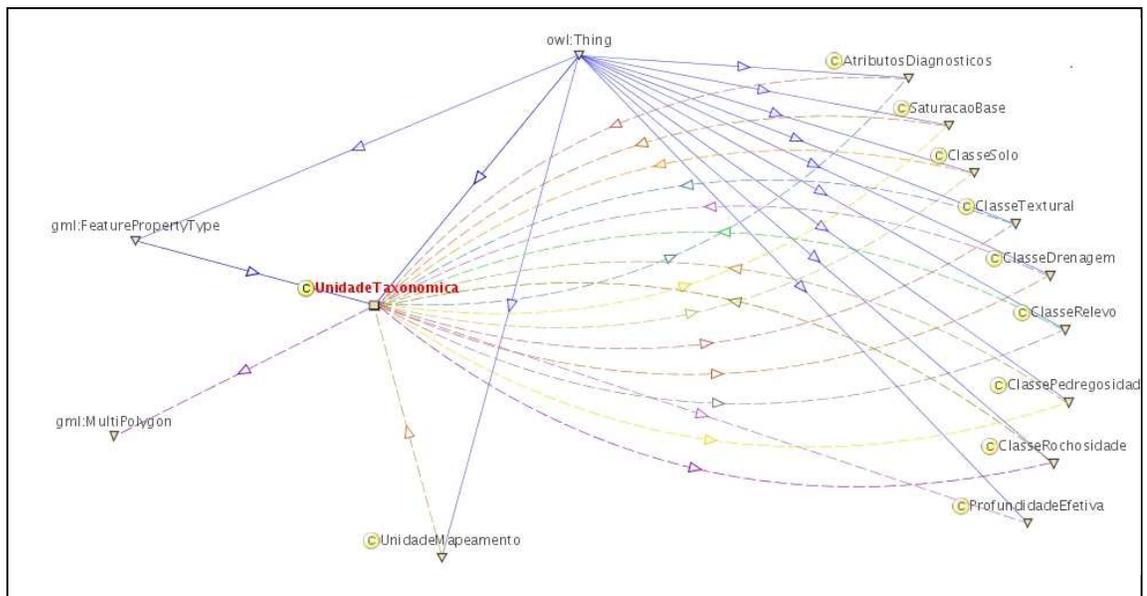


Figura 4.8 Ontologia de atributos do solo para aptidão agrícola no editor Protégé OWL

A ontologia que modela a especificação GML foi reutilizada no domínio de atributos de solo, desta forma, foi possível definir qual classe da ontologia representa um objeto geográfico. A classe que define a feição geográfica, neste caso a classe *UnidadeMapeamento*

da ontologia de atributos de solo, deve ser uma subclasse da classe *FeaturePropertyType* da ontologia da GML como mostra a figuras 4.9. Os detalhes da ontologia de atributos de solos



podem ser verificados na documentação da ontologia disponível no apêndice B.

Figura 4.9 Relacionamentos da unidade de mapeamento de solo na ontologia.

Com a elaboração da ontologia de atributos de solo para aptidão agrícola, foi definido um esquema de referência para a integração semântica. Outro objetivo de utilização desta modelagem é a realização de inferências após a integração, utilizando as regras em *semantic web rule language* (SWRL) criadas nas ontologias do projeto de zoneamento agro-ecológico do dendê. Com a possibilidade de utilização de uma máquina de inferência capaz de interpretar as regras criadas, novas informações podem ser geradas automaticamente após a integração. Segundo Rector (35), de forma pragmática, a definição e descrição das classes através de regras de restrição devem ser feitas, quando existe a necessidade de realizar classificações automáticas de indivíduos da ontologia. Futuramente, ontologias para outras culturas poderão ser criadas utilizando o arcabouço elaborado nesta base de conhecimento, possibilitando a avaliação da aptidão agrícola para outras culturas.

4.2.6 O Servidor de Integração

Com o objetivo de realizar a integração de objetos geográficos distribuídos fornecidos por fontes de dados heterogêneas, foi desenvolvida uma biblioteca em linguagem java

(<http://java.sun.com/>) denominada OpenOGI. Esta biblioteca é uma *API (Application Programming Interface)* de código aberto capaz de realizar a interoperabilidade semântica entre feições geográficas baseada em ontologias.

Para realizar o tratamento de informações geográficas, foi utilizada a *API geotools* versão 2.3. O *geotools* é uma biblioteca *java* de código aberto (*open source*) para manipulação de dados geoespaciais compatível com as especificações do consórcio OGC. Ela foi utilizada na aplicação tanto para ler e interpretar o esquema XML que define os objetos geográficos de cada fonte, quanto para manipular os objetos geográficos na integração dos dados.

A interação com a base de conhecimento foi feita utilizando a *API Protégé-OWL*. Esta *API* é uma biblioteca *java* de código aberto para manipulação de informações em linguagem OWL. Com ela é possível carregar e salvar arquivos OWL manipulando e consultando as ontologias definidas. Esta *API* é utilizada pela OpenOGI para ler as ontologias criadas e armazenar o resultado em arquivos OWL.

A OpenOGI utiliza ainda a biblioteca de métricas de similaridade entre strings SimMetrics (9) para realizar as integrações de informações de natureza qualitativa por similaridade. A biblioteca SimMetrics possui diversos algoritmos de medida de similaridade entre cadeias de caracteres que informam o grau de similaridade entre duas seqüências de caracteres. Depois de analisadas todas as possibilidades, resolveu-se adotar o algoritmo de Jaro-Winkler para fazer a associação entre uma informação de natureza qualitativa das fontes de dados com um determinado indivíduo da ontologia. Este algoritmo foi escolhido por apresentar os melhores resultados para os dados encontrados nas fontes de dados disponíveis. A tabela 4.5 apresenta um quadro comparativo entre alguns algoritmos de medida de similaridade entre cadeias de caracteres disponíveis na biblioteca SimMetrics.

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>Maior Diferença</i>	<i>Menor Diferença</i>
Levenshtein	0,88	0,63	0,42	0,19	0,69	0,21
Needleman-Wunch	0,90	0,79	0,71	0,53	0,37	0,08
Smith-Waterman	0,87	0,89	1,0	0,5	0,76	-1,3
Smith-Waterman-Gotoh	0,85	0,86	1,0	0,56	0,29	-0,15
Monge-Elkan	0,85	1,0	1,0	0,44	0,41	-0,15
Jaro	0,82	0,82	0,4	0,5	0,42	0,32
Jaro-Winkler	0,89	0,82	0,4	0,5	0,49	0,32
Q-Gram	0,70	0,60	0,52	0,25	0,45	0,08

Termos A: Podzólíco Vermelho-Amarelo / PodzolicVermelhoAmarelo
Termos B: Suave Ondulado / RelevoSuaveOndulado
Termos C: Ondulado / RelevoSuaveOndulado
Termos D: Latossolo Vermelho-Amarelo / Cambissolo

Tabela 4.5 Comparação entre algoritmos de medida de similaridade

A documentação completa da biblioteca OpenOGI pode ser consultada em formato *javadoc* (<http://java.sun.com/j2se/javadoc/>) no apêndice C desta dissertação.

Um protótipo foi criado com o objetivo de realizar as tarefas do servidor de integração (figura 4.10). Este aplicação foi desenvolvida em *java* e utiliza a biblioteca OpenOGI para integrar os dados espaciais.

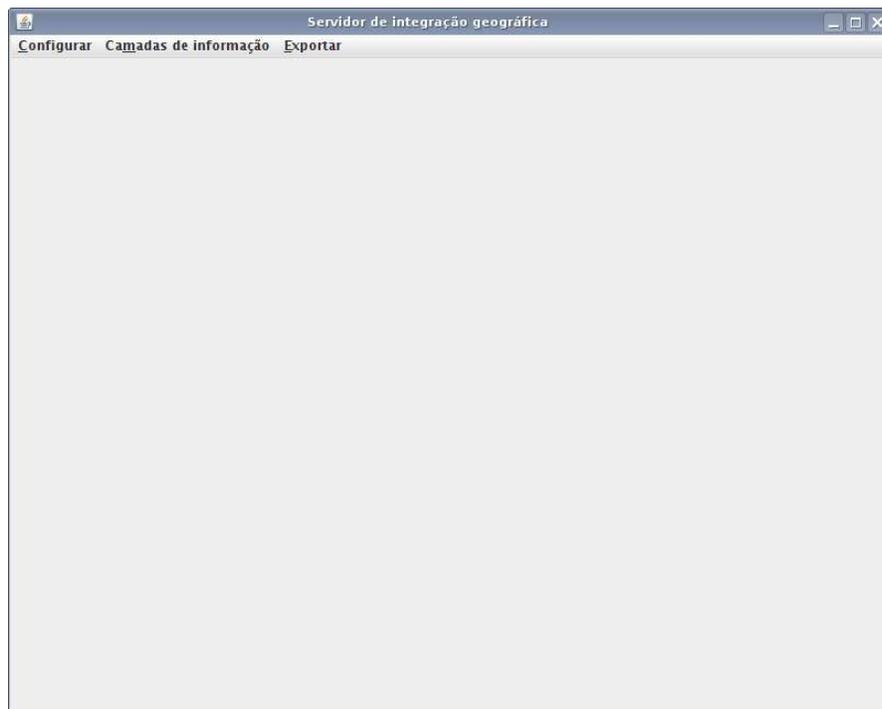


Figura 4.10 Protótipo do Servidor de Integração

O servidor de integração possui duas funcionalidades de configuração para que seja possível realizar a interoperabilidade entre as fontes de dados espaciais heterogêneas: o cadastramento de domínios e o cadastramento das fontes de dados que deverão ser integradas. A figura 4.11 apresenta a funcionalidade de cadastramento de domínios.

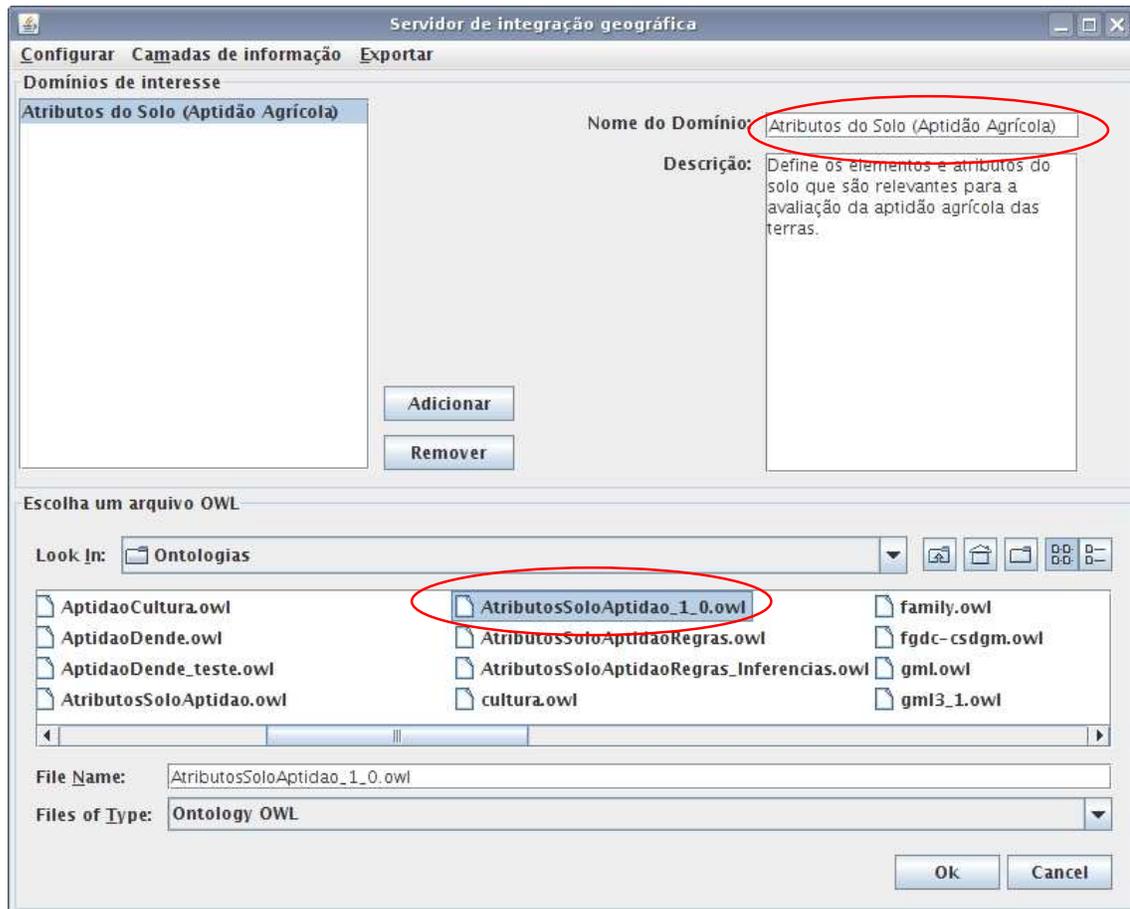


Figura 4.11 Cadastramento de Domínios

No cadastramento do domínio, um nome, uma breve descrição e o arquivo OWL contendo a ontologia é atribuído ao domínio. O arquivo OWL que contém a ontologia do domínio é armazenado no servidor para ser utilizado posteriormente como referência semântica na integração.

O cadastramento das fontes de dados geográficos (figura 4.12) armazena informações sobre cada fonte disponível para a integração. Cada fonte é relacionada diretamente a um domínio de interesse e possui uma relação de metadados que a definem. Os metadados geográficos devem ser escolhidos de acordo com a necessidade de cada integração e devem seguir o padrão ISO 19115. Os metadados apresentados na figura 4.12 foram escolhidos

especificamente para o estudo de caso. A tabela 4.6 apresenta os itens de metadados escolhidos e o item correspondente na norma da ISO.

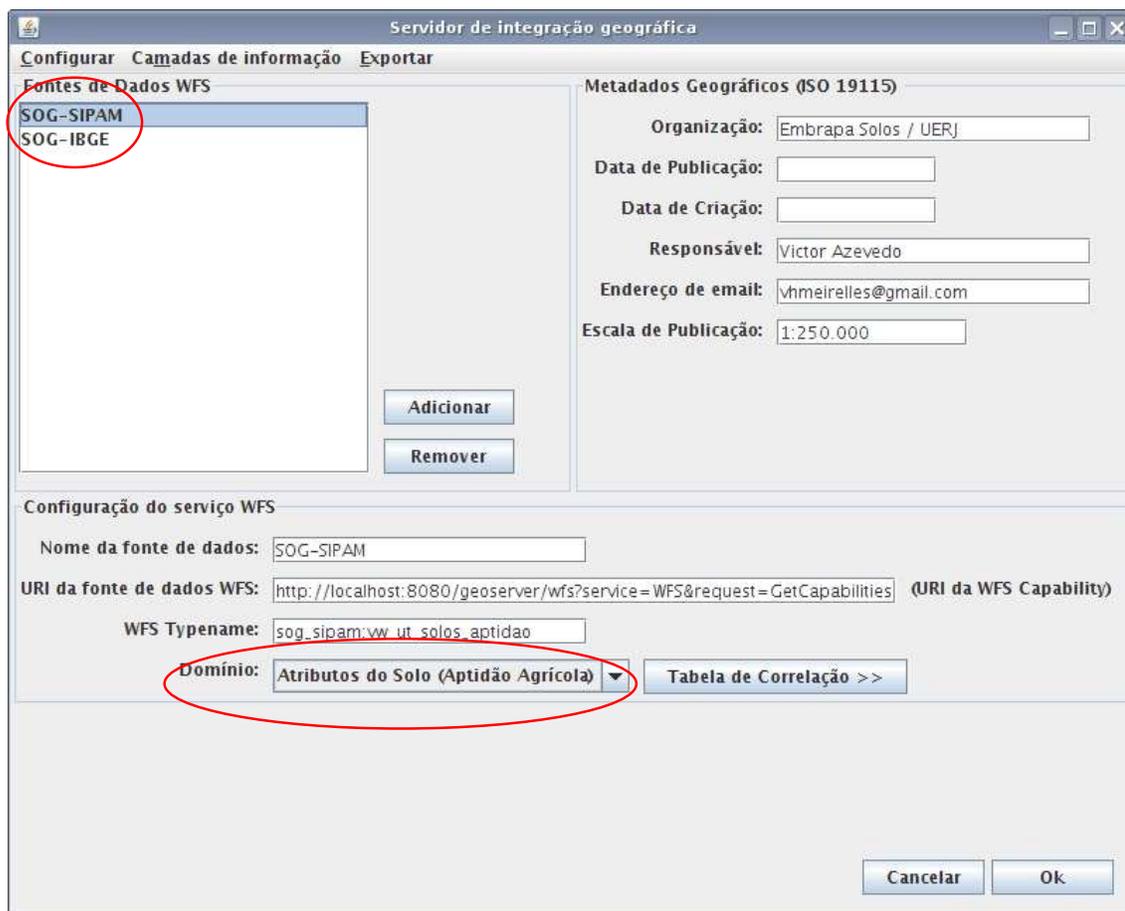


Figura 4.12 Cadastramento das fontes de dados geográficos

Estudo de Caso	ISO 19115
Organização	CI_ ResponsibleParty.organisationName
Data de publicação	CI_DateTypeCode.publication
Data de criação	CI_DateTypeCode.creation
Responsável	CI_ ResponsibleParty.individualName
Endereço de email	CI_Address.electronicMailAddress
Escala de publicação	MD_Resolution.equivalentScale

Tabela 4.6 Correspondência entre os itens de metadados utilizados e os itens da ISO 19115.

As informações do endereço do serviço WFS (*getCapabilities*) e do tipo de feição geográfica (*FeatureType*) cadastrada no servidor WFS informam a localização do serviço para a correlação com o esquema semântico e para a integração dos dados. A correlação sintática e semântica é feita em uma tabela de correlação (figura 4.13) que o responsável pela publicação

do serviço deve preencher. Os termos do esquema XML retornado pela função *getFeatureType* do serviço WFS da fonte de dados devem ser correlacionados um a um aos termos da ontologia do domínio ao qual a fonte está associada.

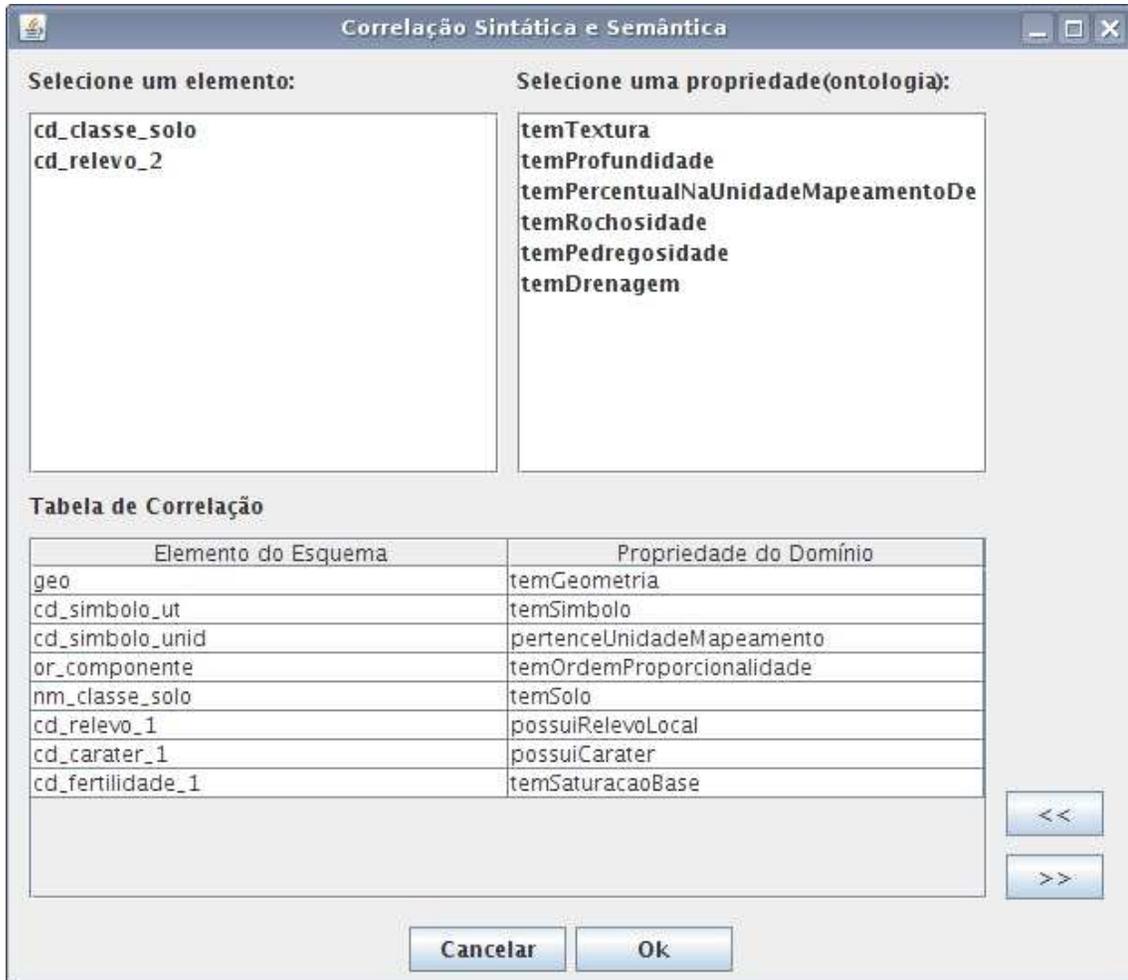


Figura 4.13 Tabela de correlação sintática e semântica

O servidor de integração utiliza a biblioteca OpenOGI, que implementa os conceitos abordados pela metodologia proposta. Esta biblioteca é capaz de buscar as propriedades dos objetos geográficos da fonte de dados no servidor WFS. Os termos da ontologia são adquiridos através da API *Protégé-OWL*, que lê o arquivo cadastrado no domínio, identifica qual classe da ontologia é subclasse de *FeaturePropertyType* e busca as suas propriedades. No caso da ontologia de atributos de solo para aptidão, a classe que representa as unidades taxônomicas foi definida como subclasse *FeaturePropertyType*, portanto as propriedades da ontologia que tenham em seu domínio a classe *UnidadeTaxonomica* poderão ser correlacionadas com os atributos dos objetos geográficos distribuídos.

O cadastramento das fontes de dados do IBGE e do SIPAM foram realizados no servidor de integração. As duas fontes foram associadas ao mesmo domínio criado, o domínio de atributos de solo para a avaliação da aptidão agrícola, e o esquema dos objetos geográficos de cada fonte correlacionado às propriedades da classe de unidade de mapeamento de solo definida na ontologia. Com estas configurações realizadas no servidor de integração os aspectos sintático, semântico e de formato e estrutura dos dados espaciais heterogêneos foram contemplados, permitindo que a interoperabilidade seja realizada.

4.2.7 Publicação das Informações Integradas de Atributos de Solos

A solução adotada neste trabalho utiliza o servidor de integração para integrar os dados espaciais distribuídos. O protótipo possui a funcionalidade de geração de camadas de informação baseadas no domínio de interesse definido pela base de conhecimento (figura 4.14). À camada, é atribuído um nome, o domínio de interesse ao qual ela está associada, e as fontes de dados geográficos associadas a este domínio que deverão fornecer os objetos geográficos (figura 4.15).

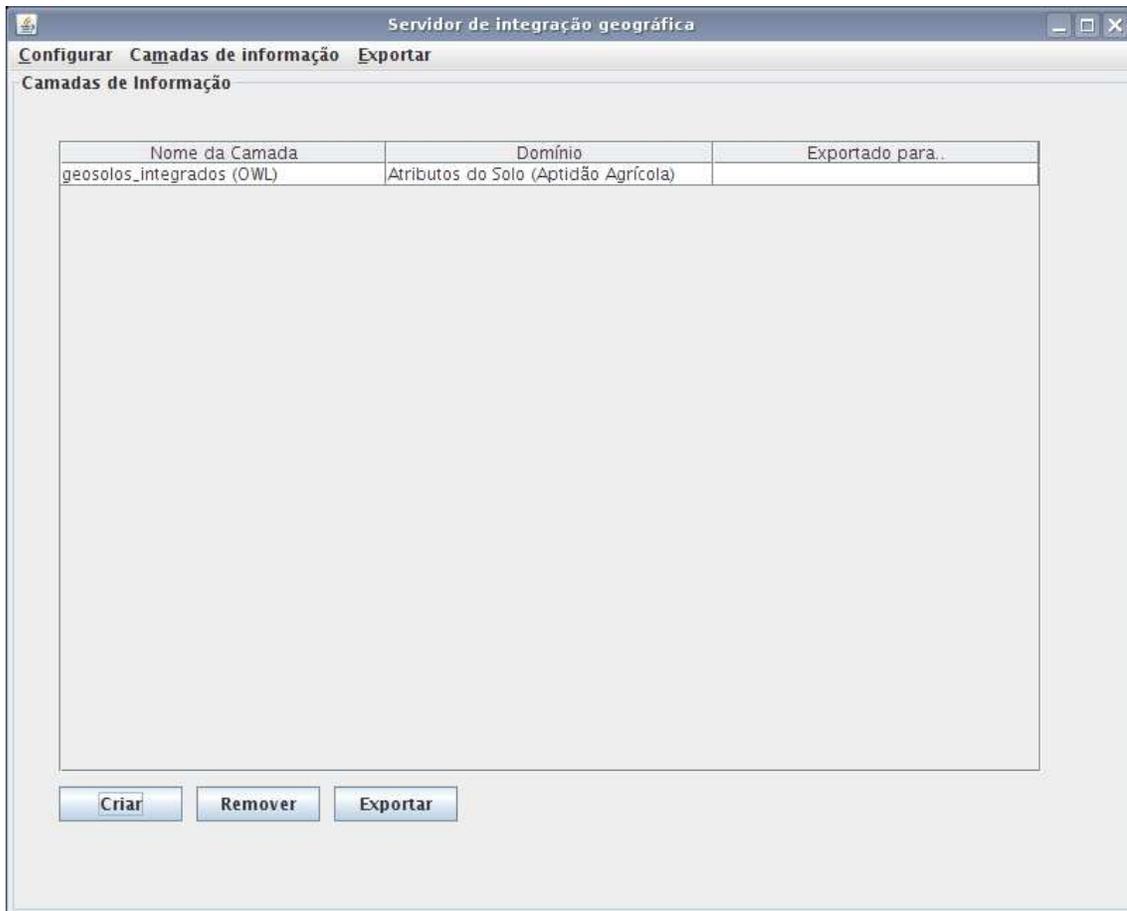


Figura 4.14 Funcionalidade de geração de mapa

A criação do mapa em si integra os objetos geográficos heterogêneos utilizando todas as configurações realizadas no servidor de integração. Utilizando a API *geotools*, o aplicativo gera esquema conceitual baseado na ontologia e busca os objetos heterogêneos armazenados em cada fonte armazenando-os em um arquivo seguindo o novo esquema criado.

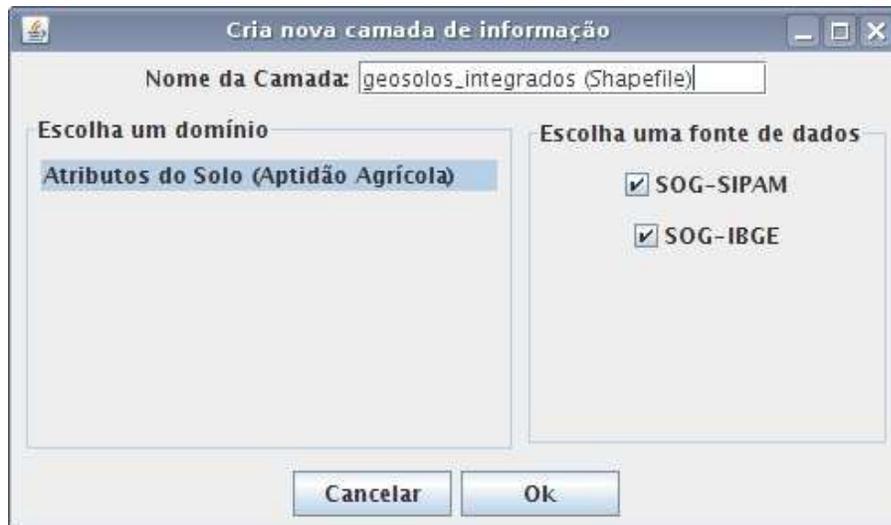


Figura 4.15 Funcionalidade de geração de mapa

Neste estudo de caso, os objetos geográficos das fontes de dados do SIPAM (carta SC18) e do IBGE (carta SB18) foram integrados utilizando o esquema definido na ontologia de atributos de solo para aptidão. O resultado gerado foi uma camada de informação de solos contendo as unidades de mapeamento das duas fontes de dados com um único esquema conceitual definido pela ontologia. Esta camada foi exportada para um arquivo em formato *shapefile* e posteriormente publicada como um serviço WFS. Este serviço fornece os dados integrados e um esquema conceitual baseado na representação do conhecimento sobre o domínio.

4.3 Resultados

O resultado obtido foi uma camada de informação com as informações integradas de unidade de mapeamento de solos da área de estudo (figura 4.16), cartas SB-18 e SC-18 do mapa índice do IBGE. As informações da carta SB-18 foram adquiridas a partir da fonte de dados do IBGE configurada no servidor de integração (SOG-IBGE), enquanto as informações da carta SC-18 foram obtidas a partir da fonte de dados do SIPAM (SOG-SIPAM). A figura 4.17 apresenta o mapa de solos integrado com os objetos geográficos provenientes das duas fontes de dados.

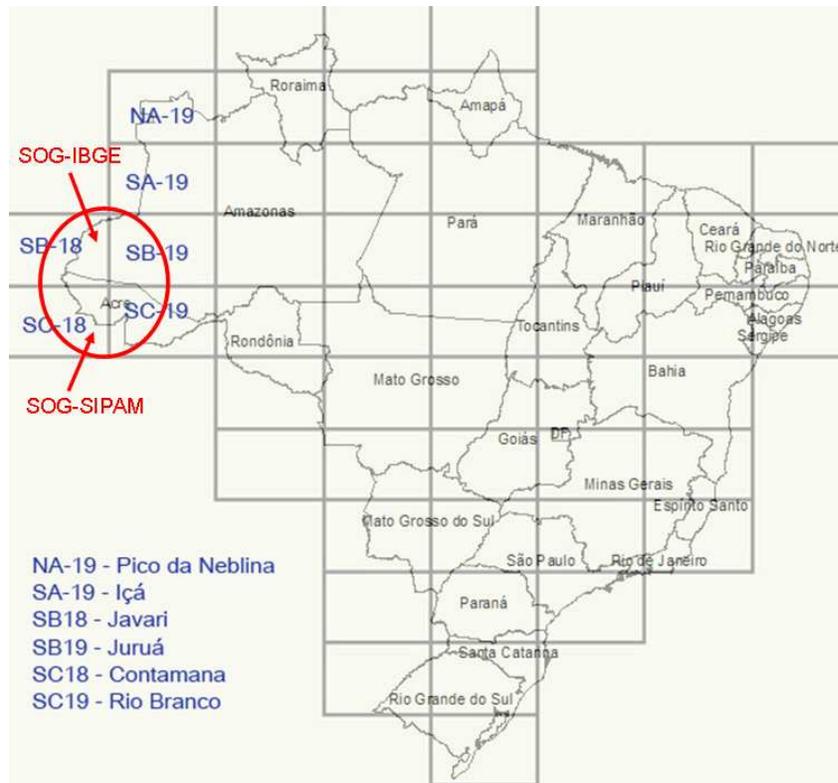


Figura 4.16 Área de estudo

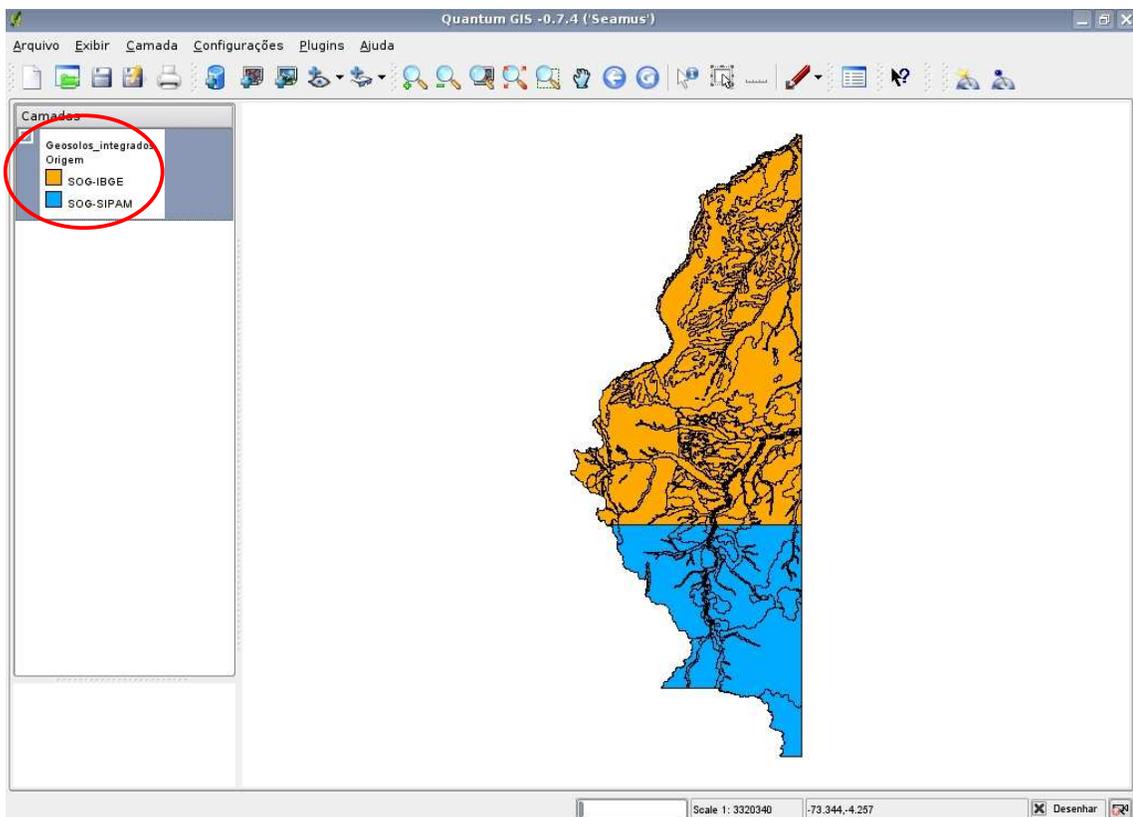


Figura 4.17 Camada de informação de solos formada por feições geográficas integradas

Os dados espaciais, que nas suas fontes de origem eram heterogêneos, agora estão armazenados sob um mesmo esquema conceitual. A figura 4.18 apresenta os atributos de uma unidade de mapeamento selecionada no SIG. Nota-se que o objeto possui um esquema conceitual baseado nas propriedades da classe unidade de mapeamento da ontologia de atributos de solo para a aptidão agrícola. Como demonstra a figura, a origem do objeto selecionado foi a fonte de dados do IBGE (SOG-IBGE).

Feição	Valor
temSimbolo	SB18PVAa11
origem	SOG-IBGE
pertenceUn	
possuiCara	Alítico
possuiRele	suave ondulado e ondulado
temOrdemPr	0
temPedregoso	
temRochoso	
temSaturac	
temSimbolo	SB18PVAa11
temSolo	ARGISSOLO VERMELHO-AMARELO
temTextura	média sem cascalho/argilosa sem cascalho e argilosa sem cascalho/argilosa sem cascalho

Figura 4.18 Atributos de uma unidade de mapeamento da camada de informação integrada

Os atributos dos objetos da carta SC18, fornecida pela fonte de dados do SIPAM, estão armazenados sob a mesma estrutura, com o mesmo esquema conceitual baseado na ontologia do domínio. A figura 4.19 mostra a tabela de atributos de um objeto proveniente da fonte de dados SOG-SIPAM (objeto selecionado). Nota-se que os atributos do objeto utilizam o mesmo modelo dos objetos da fonte SOG-IBGE, ou seja, as propriedades da classe unidade taxonômica na ontologia de atributos do solo. Pode ser observado na figura, que o esquema não está idêntico ao apresentado pela ontologia (apêndice B) devido à limitação de número de caracteres para nomeação de atributos imposta pelo formato shapefile.

A fonte de dados do IBGE fornece as informações consolidadas das unidades de mapeamento, enquanto a fonte de dados do SIPAM fornece as informações detalhadas por cada unidades taxonômica de solo que compõe as unidades de mapeamento. Por este motivo, nota-se na figura 4.19 que existem dois elementos associados a mesma geometria, cada um com características distintas. Posteriormente, serão realizadas inferências no mapa visando obter a informação do percentual de cada unidade taxonômica pertencente à unidade de mapeamento.

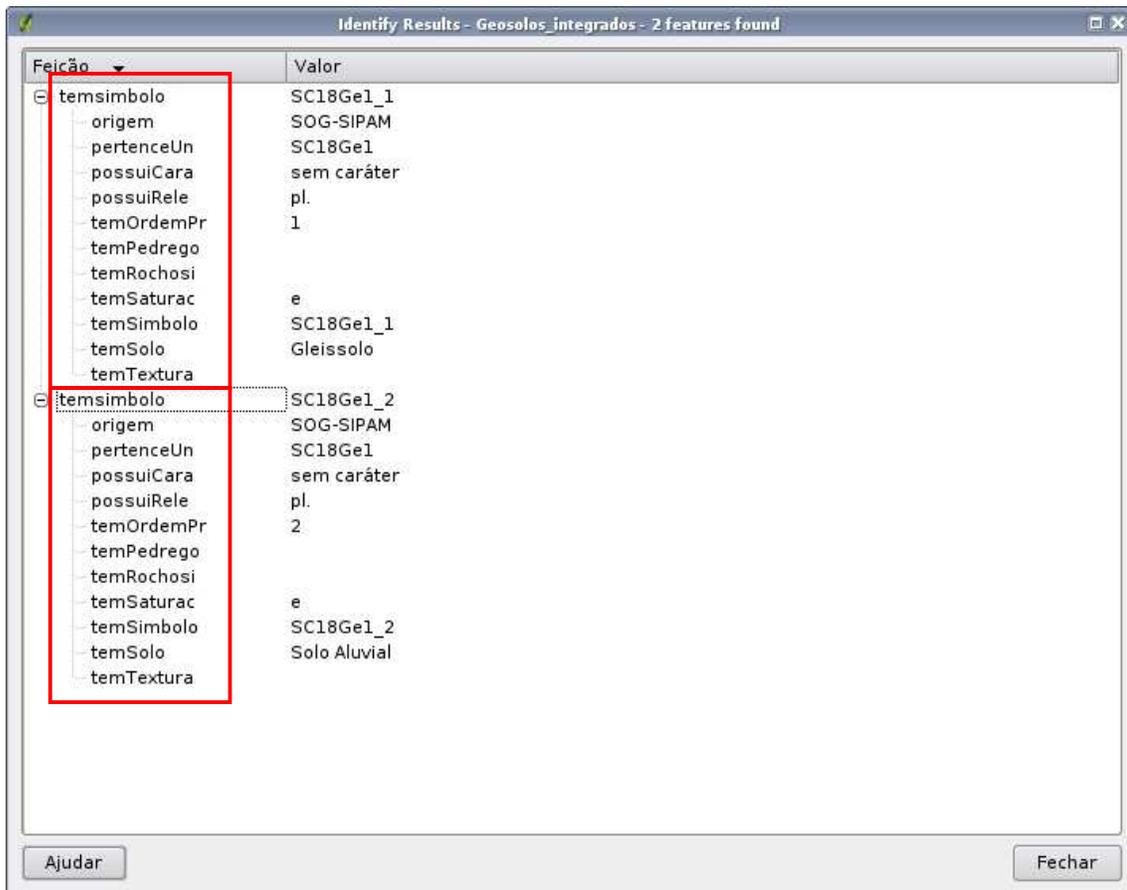


Figura 4.19 Seleção da unidade de mapeamento do SIPAM no mapa integrado

4.4 Utilização no projeto Zondende

Este estudo de caso está sendo aplicado no projeto de zoneamento agro-ecológico do dendê, o Zondende, para exportação dos dados de múltiplas fontes para a base de conhecimento do projeto. As informações das unidades de mapeamento e unidades taxonômicas fornecidas pelas fontes de dados estão sendo armazenadas como indivíduos da ontologia de atributos de solos na base de conhecimento (figura 4.20).

O objetivo principal é utilizar estas informações para inferir o valor da aptidão agrícola de cada elemento a partir das suas informações originais vindas de cada fonte. Esta inferência será feita utilizando uma máquina de inferências (*reasoner*) que, a partir da base de conhecimento criada e das suas regras cadastradas em linguagem SWRL, irá gerar novo conhecimento, neste caso, a classe de aptidão do solo. As regras levam em consideração as informações de relevo, textura, classe, fertilidade e outras do solo para inferir a classe de aptidão de cada unidade taxonômica e cada unidade de mapeamento.

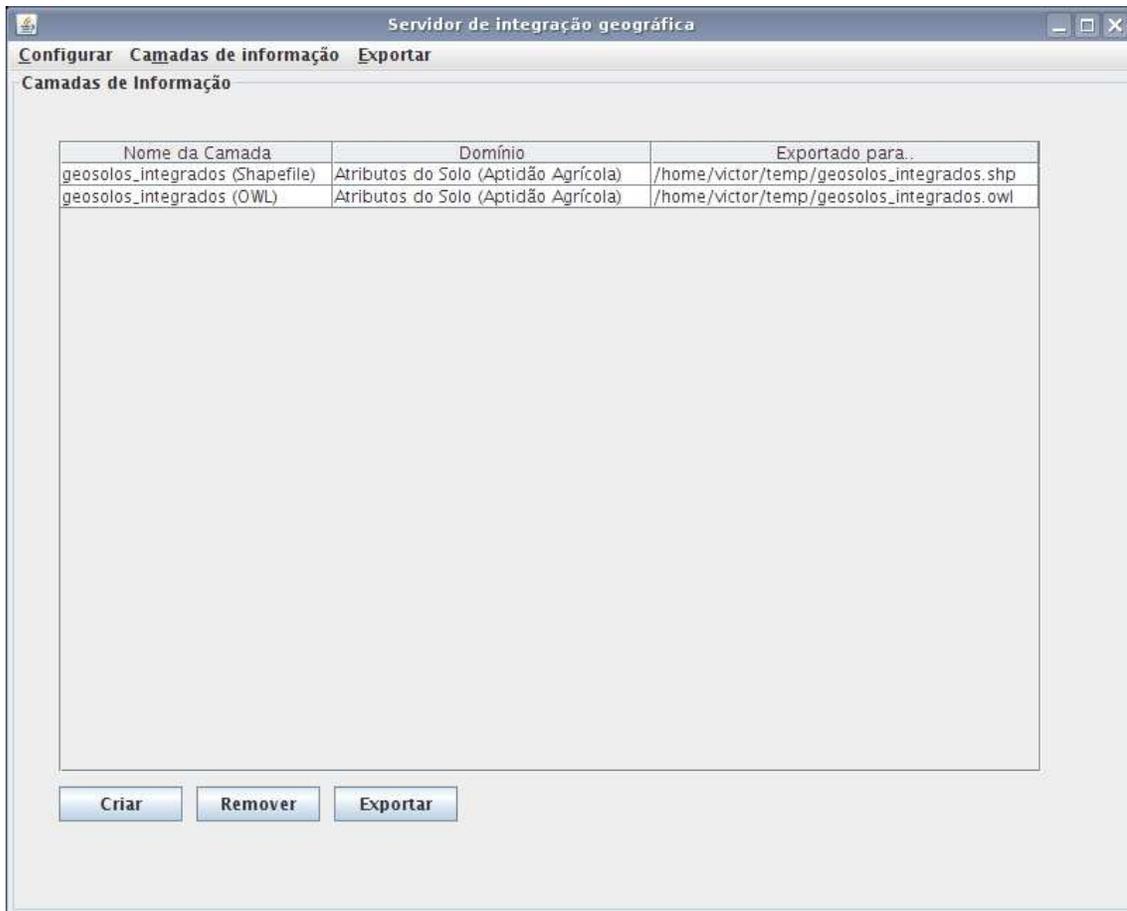


Figura 4.20 Exportação de dados para a ontologia OWL

Para este caso, foi necessário estabelecer uma correlação entre os dados e os indivíduos do *range* de cada propriedade das unidades taxonômicas. Para isso, foi utilizado o algoritmo de similaridade entre cadeias de caracteres de Jaro-Winkler e, quando necessário, a tabela de correlação de informações qualitativas definida na metodologia (item 3.4.2). A figura 4.21 apresenta os indivíduos cadastrados na ontologia a partir da integração dos dados. Os indivíduos das propriedades da unidade taxonômica (destacados na figura 4.21) foram atribuídos a partir da correlação entre informações qualitativas.

As informações do percentual que a unidade taxonômica ocupa na unidade de mapeamento é atribuído à base de conhecimento através de um conjunto de regras SWRL. A figura 4.22 apresenta o conjunto de todas as regras e as inferências realizadas pela máquina de inferências enquanto a figura 4.23 apresenta uma das regras.

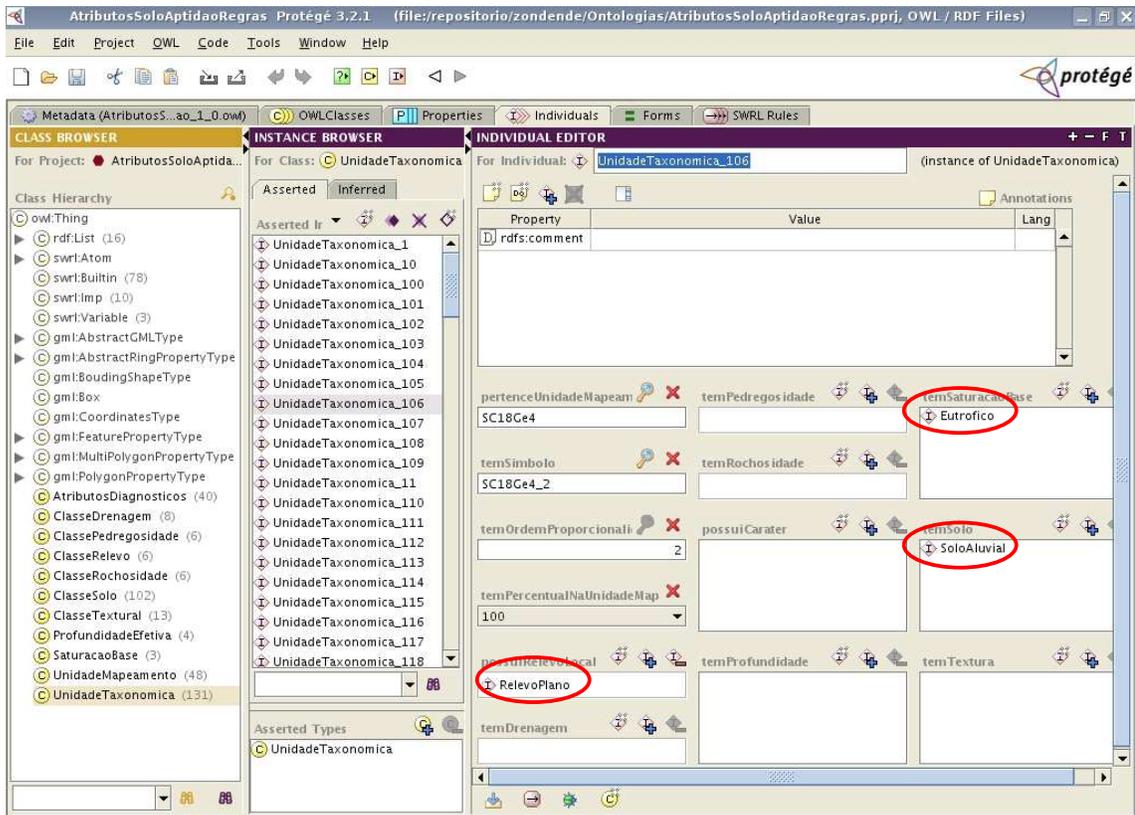


Figura 4.21 Indivíduos (unidades taxonômicas) da ontologia

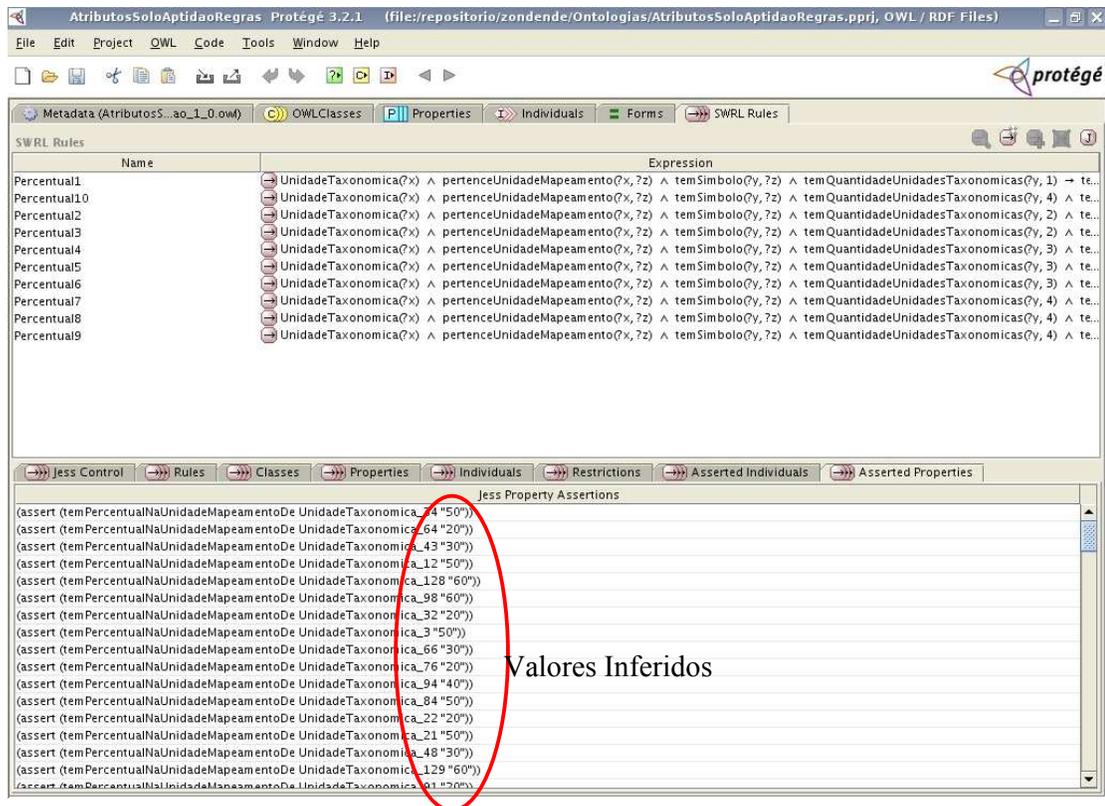


Figura 4.22 Conjunto de regras e as inferências realizadas pelo reasoner

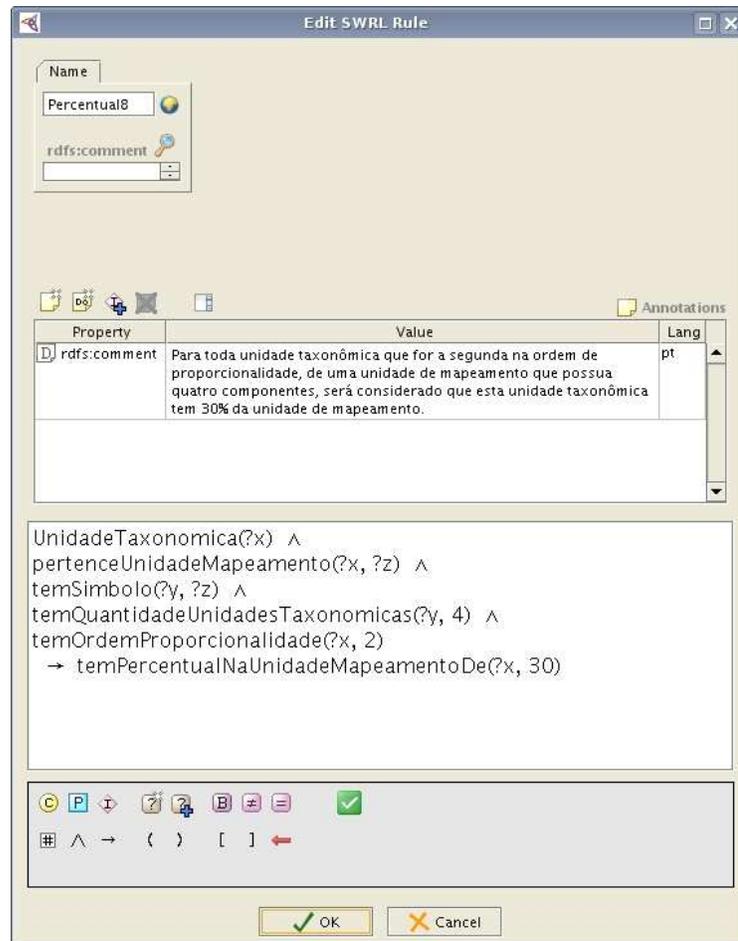


Figura 4.23 Regra SWRL que define o percentual da unidade taxonômica

4.5 Considerações finais

O estudo de caso descrito apresenta o ambiente de teste da metodologia proposta nesta dissertação. Nele, foram apresentadas duas fontes de dados geográficos heterogêneos no domínio de solos, onde os métodos e procedimentos sugeridos no capítulo 3 foram aplicados com o objetivo de permitir a interação entre os dados espaciais das duas fontes. Os resultados de cada fase da metodologia foram apresentados ao longo da descrição do estudo de caso. O resultado final apresentou o mapa de solos da região de estudo com estrutura homogênea, contendo as informações de solos necessárias para a avaliação da aptidão agrícola do dendê no âmbito do projeto de zoneamento agro-ecológico do dendê na Amazônia Legal. As hipóteses levantadas foram avaliadas e comprovadas. A utilização dos padrões do consórcio OGC, principalmente a utilização de GML e WFS, conjuntamente com a definição de ontologias de domínio possibilitaram uma maior interoperabilidade entre fontes de objetos geográficos heterogêneos distribuídas.

Neste estudo não foi dada ênfase à utilização de metadados porque o objetivo foi apenas integrar as informações das fontes de dados disponíveis para o projeto. Para situações onde exista a necessidade de se realizar buscas diretamente nas fontes de dados de origem, pode ser necessário um estudo mais detalhado dos metadados, baseando-se na norma ISO 19115. As buscas serão tão precisas quanto melhor for a escolha dos metadados. Para a busca dos dados em si, devem-se utilizar consultas baseadas em ontologia (*ontology-based query*).

A utilização de ontologias como esquema de referência para a interoperabilidade, além de garantir a homogeneidade sintática e semântica, possibilita a agregação de novo conhecimento após a integração dos dados. Esta característica mostrou-se bastante útil para este estudo de caso, visto que havia necessidade de inferir novas informações a partir dos dados existentes.

5 Conclusões

Neste trabalho foi proposta uma metodologia para alcançar a interoperabilidade entre objetos geográficos heterogêneos e distribuídos levando em consideração os aspectos sintáticos, semânticos, de estrutura e forma que determinam a incompatibilidade entre os dados geográficos disponíveis. O uso de padrões nas diversas áreas do conhecimento, e principalmente na área das Geociências, pode ser uma maneira de alcançar a interoperabilidade.

A metodologia proposta mostrou resultados satisfatórios utilizando padrões para a integração sintática, semântica e de formato de dados, quando aplicada aos dados de solos da Amazônia disponíveis para o projeto de zoneamento agro-ecológico do dendê. A utilização das especificações de padrões do consórcio OGC aliada ao uso de ontologias de domínio, utilizando a linguagem de ontologia padrão OWL definida pelo consórcio W3C, permitiu que os dados espaciais de solos do SIPAM e do IBGE fossem integrados.

Os servidores de serviços WFS, que estão em crescente evolução ultimamente, mostraram bons resultados na tarefa de unificação de formatos de dados. O fato de realizarem a conversão automática de diversos formatos diferentes para o formato GML permitiu que os dados fossem transformados para um formato único com algumas poucas configurações. O ganho de produtividade que a utilização deste tipo de ferramenta pode trazer para o processo de interoperabilidade de dados espaciais é um fato bastante relevante.

Um elemento relevante, e que não foi abordado no estudo de caso, é a avaliação da performance da publicação de dados geográficos em WFS. Como os dados espaciais geralmente são muito grandes, é preciso avaliar a viabilidade para cada caso, já que as informações são publicadas em GML, uma extensão do XML, que é um formato que proporciona uma performance insatisfatória.

O processo de engenharia do conhecimento proposto nesta metodologia apresentou bons resultados para a elaboração da ontologia do domínio de interesse. O processo contou com a colaboração expressiva de pesquisadores pedólogos da EMBRAPA SOLOS. A interação entre estes especialistas e os engenheiros do conhecimento foi importante para a elaboração de um modelo mais próximo da realidade.

Consultas a termos já consagrados na bibliografia foram amplamente utilizadas melhorando a representação do conhecimento na área de solos e aptidão agrícola. Contudo, elaborar modelos computacionais que representem o conhecimento não é uma tarefa fácil, de

forma que a utilização de um processo formal de engenharia do conhecimento facilita a modelagem.

A utilização de uma ferramenta gráfica de edição de ontologias que utilize uma interface de entrada de dados intuitiva, com geração de gráficos e documentação, facilita a interação entre os envolvidos no processo de representação cognitiva. Neste sentido, o editor de ontologias *Protégé-OWL* foi eficaz ao ser utilizado para modelar as ontologias que compõem a base de conhecimento do projeto Zon-dende. Com esta ferramenta, as dificuldades provocadas pela troca de conhecimento de especialistas de diferentes áreas foram minimizadas.

A criação de uma biblioteca em java, de código aberto, que implementa os conceitos abordados na metodologia proposta nesta dissertação, tem como principal objetivo contribuir com a evolução da pesquisa nesta área de conhecimento.

6 Trabalhos Futuros

Este capítulo apresenta os futuros trabalhos que serão realizados no âmbito do projeto de zoneamento agro-ecológico do dendê na Amazônia e sugestões para melhorias metodológicas no processo de realização da interoperabilidade entre dados geográficos heterogêneos em ambiente distribuído.

Para o projeto Zon-dende, os próximos passos que serão seguidos ampliam a aplicação da metodologia a outras fontes de dados geográficos e a toda a área da Amazônia Legal, posto que o estudo de caso foi aplicado somente em duas cartas da região da fronteira. A experiência com um volume de dados maior pode trazer novas conclusões e resultados.

Deseja-se realizar inferências na ontologia de aptidão agrícola do dendê visando fornecer uma maneira de classificar automaticamente as unidades de mapeamento nas classes de aptidão definidas na base de conhecimento. Para isto, deve-se incluir as funcionalidades de geração automática de inferências na biblioteca OpenOGI. A inclusão desta funcionalidade permitirá a criação automática de um mapa de aptidão da terra a partir dos dados e da base de conhecimento elaborada.

Entre as sugestões de melhoria da metodologia, é possível propor a utilização mais efetiva do serviço de catálogo do consórcio OGC. Esta mudança poderá significar a especificação de um mecanismo de busca eficiente baseada em metadados com o auxílio das especificações de consulta do *OpenGIS Catalog Service*.

A busca semântica baseada na ontologia de domínio poderia ser mais bem especificada na metodologia proposta tendo por objetivo a consulta de objetos geográficos distribuídos. Isto seria muito útil se for considerada a utilização da metodologia para criação de uma rede cooperativa de informações geográficas distribuídas, ou consórcios de dados espaciais.

Apêndice A - Configuração do Servidor Geoserver (29)

O Servidor geoserver é um software livre capaz de interpretar os dados espaciais e disponibiliza-los através de serviços web, seguindo as especificações Web Feature Services (WFS) e Web Map Service (WMS) do consórcio OGC. A seguir são apresentados os passos para a configuração do serviço WFS utilizado na integração dos dados realizada no estudo de caso (capítulo 4).

A.1 Configuração do serviço WFS

Primeiramente, são realizadas as configurações de definição do servidor WFS. A figura A.1 apresenta as configurações do servidor WFS SOG-SIPAM no *geoserver*.

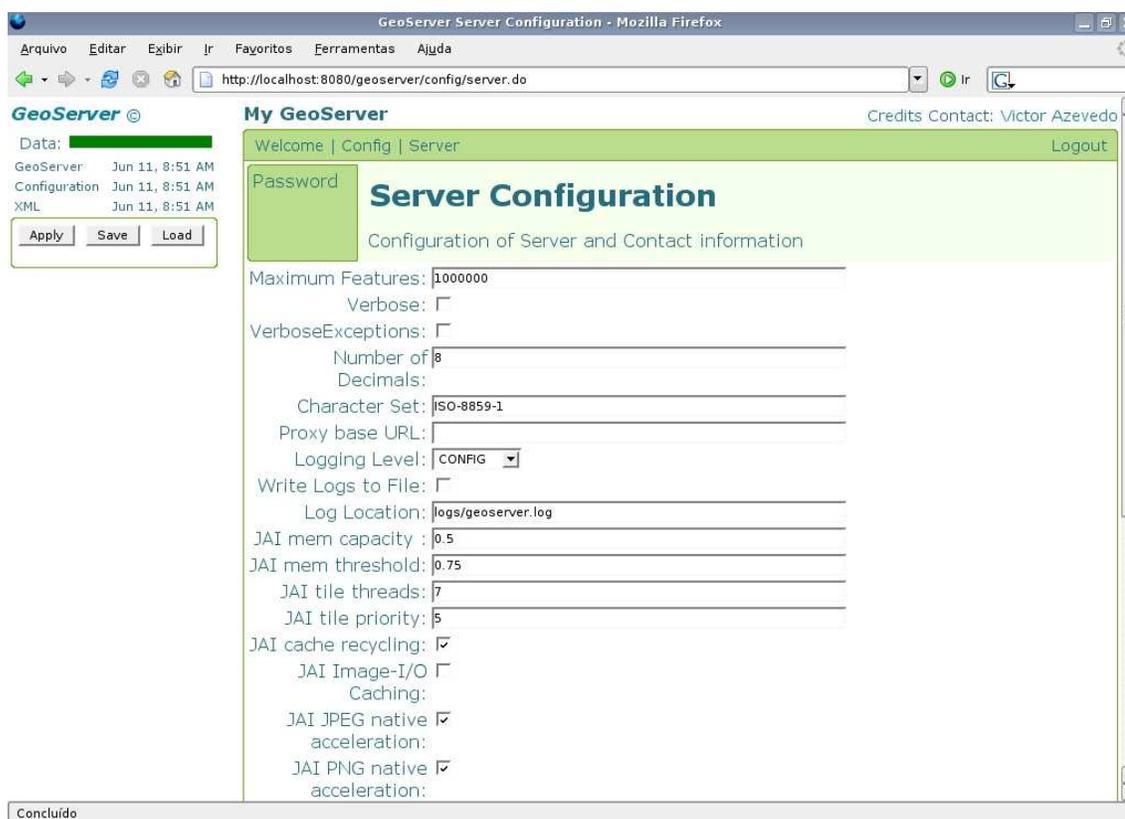


Figura A.1 Configuração das informações sobre o servidor WFS

A configuração do servidor é composta por uma área de definição de parâmetros e uma área de metadados do servidor como demonstra a figura A.2.

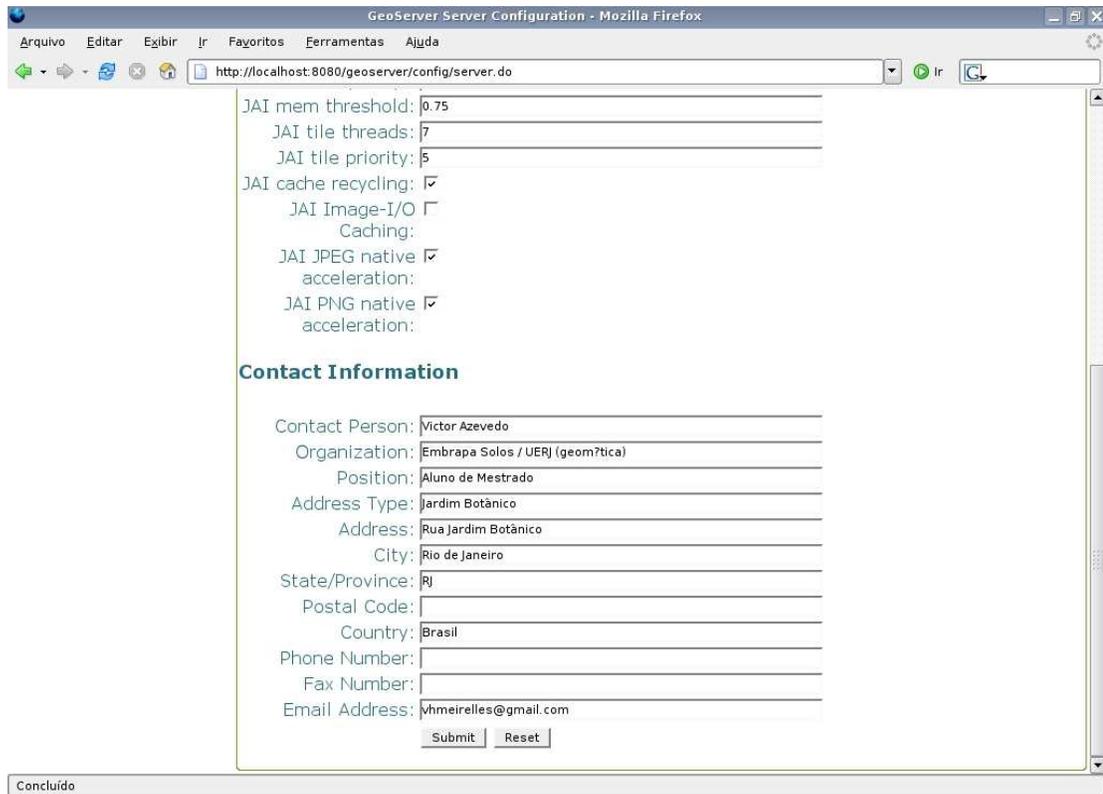


Figura A.2 Metadados do servidor

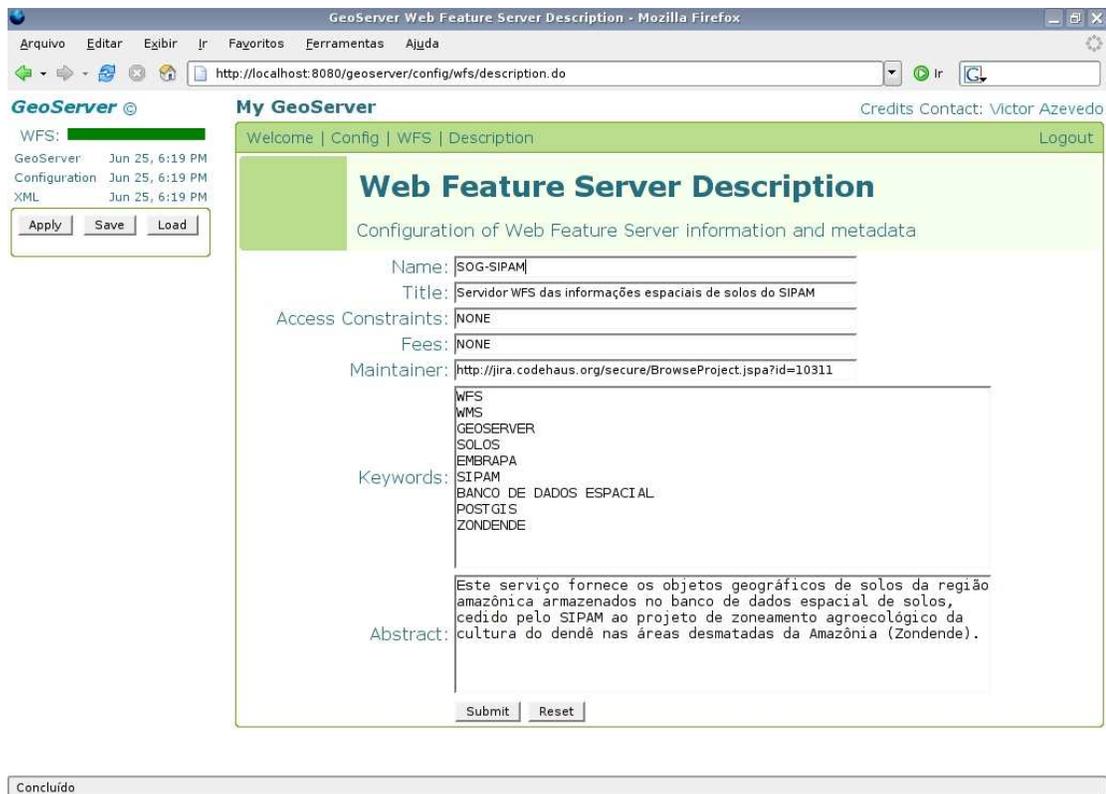


Figura A.3 Configuração da descrição do serviço WFS.

Posteriormente, são configuradas as informações de descrição do serviço que podem ser consultadas quando a função *getCapabilities* (25) do serviço WFS for requerida. A figura A.3 apresenta a tela de configuração no *geoserver* com as informações de metadados do serviço.

A.2 Configuração das fontes de dados

O próximo passo é realizar a configuração, da fonte de dados que será utilizada para fornecer os dados que serão convertidos para o formato GML e disponibilizados através da função *getFeature* do serviço WFS. Para isto, é preciso criar uma nova fonte de dados no servidor e posteriormente configurar as informações de acesso ao banco de dados. As figuras A.4 e A.5 demonstram a configuração de uma fonte de dados em postgis.



Concluído

Figura A.4 Criação da fonte de dados no servidor WFS *geoserver*

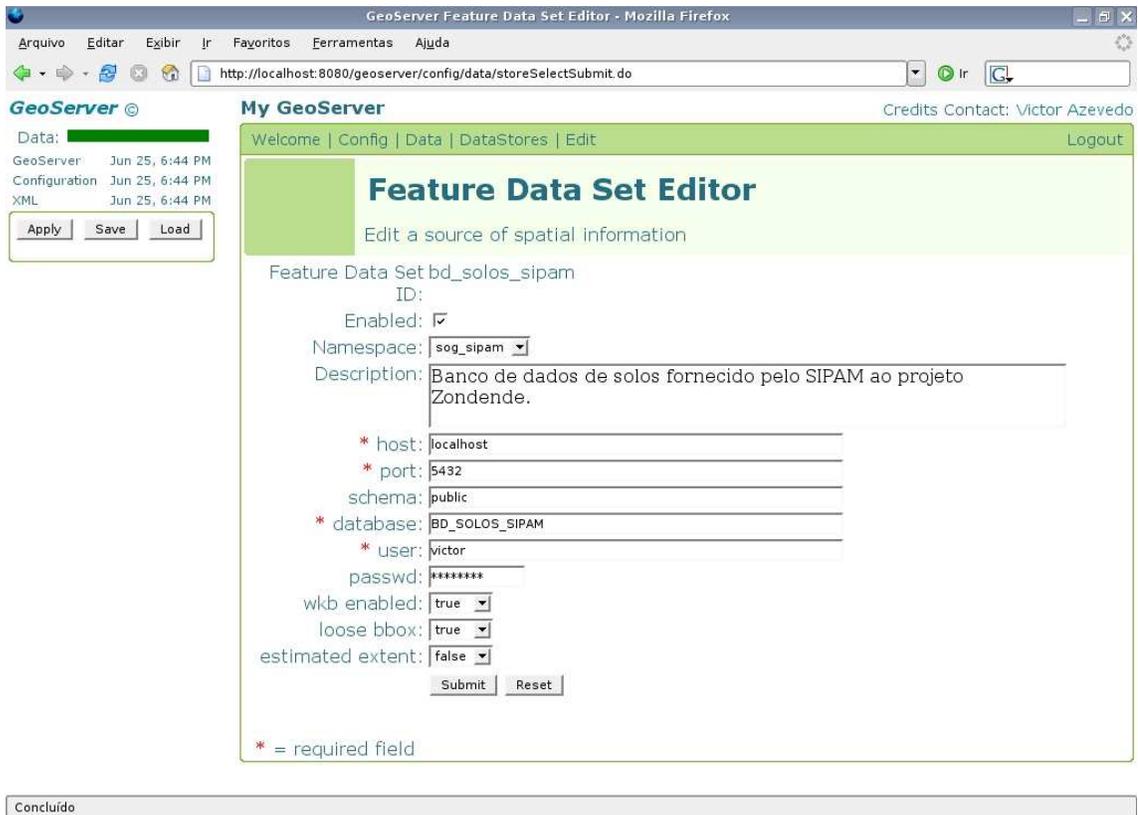


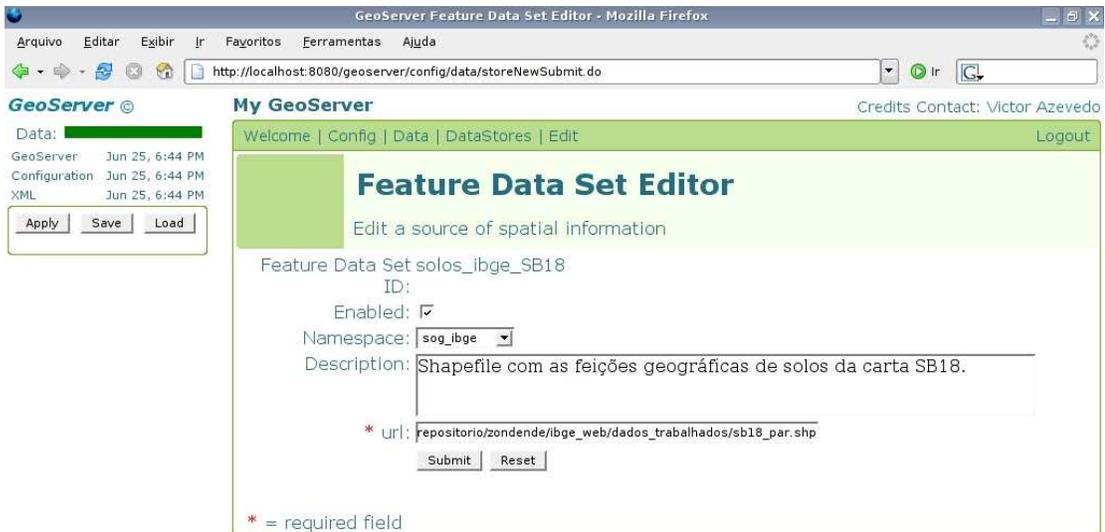
Figura A.5 Configuração da fonte de dados SOLOS_SIPAM no *geoserver*

Dependendo do tipo de fonte de dados, a sua configuração pode ser ligeiramente diferente. As figuras A.6 e A.7 apresentam os passos que devem ser realizados no *geoserver* para que uma fonte de dados do tipo *shapefile* seja cadastrada no servidor.



Concluído

Figura A.6 Criação da fonte de dados solos_ibge_SB18 no servidor SOG-IBGE



Concluído

Figura A.7 Configuração da fonte de dados solos_ibge_SB18 no servidor SOG-IBGE.

A.3 Criação de um novo tipo de feição geográfica (WFS FeatureType)

Com a fonte de dados configurada, são publicados os dados (*features*) configurando os serviços disponibilizados. Para isto, são criados e configurados os tipos de feições geográficas (*featureTypes*) informando os campos do esquema conceitual, quadrante, sistema de referência e etc. A figura A.8 apresenta a tela de criação do tipo de feição utilizado para publicar os dados de solos do SIPAM no *geoserver*. A fonte de dados SOLOS_SIPAM fornece a visão criada como opção de tipo de feição.



Concluído

Figura A.8 Criação de um tipo de feição geográfica no *geoserver*

As figuras A.9 e A.10 apresentam as configurações do tipo de feição. A figura A.9 apresenta a parte da configuração que define o estilo da camada que será utilizado no serviço WMS e o sistema de referência de coordenadas geodésicas utilizado pela fonte de dados. Pode-se notar, que devemos informar o código do sistema de referência (campo SRS) utilizado pelos objetos espaciais contidos na fonte de dados, e a partir desta informação gerar o retângulo envolvente dos dados. Ao gerar o retângulo envolvente, o servidor apresenta as informações do sistema de referência para que sejam conferidas.

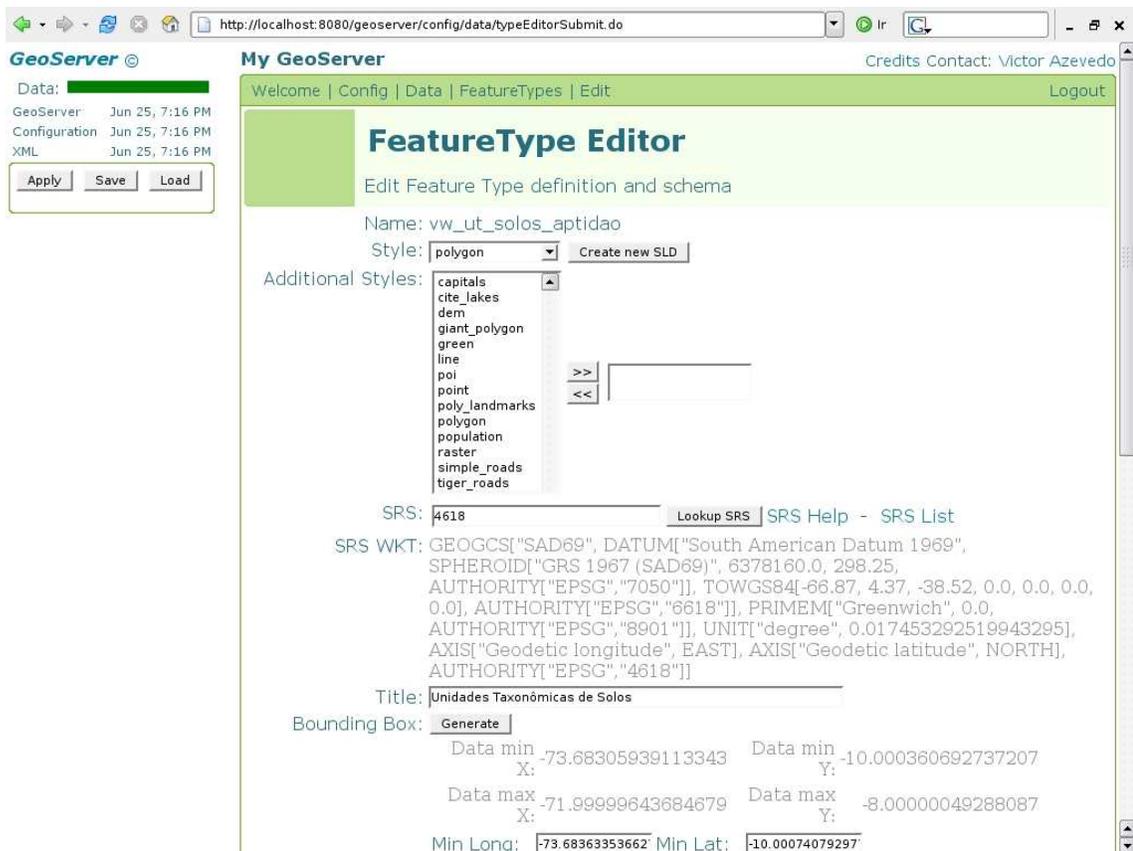


Figura A.9 Configuração do tipo de feição no *geoserver* – estilo e sistema de referência

A figura A.10 apresenta a descrição do tipo de feição geográfica e o esquema que define os dados nela contidos como o nome do campo, o seu tipo, valores máximo e mínimo permitidos e se aceita valores nulos. O esquema segue as especificações da linguagem Geographic Markup Language (GML).

A.4 Verificando o resultado

Realizada a configuração do serviço, é possível acessar as informações disponíveis utilizando-se as funções definidas na especificação WFS: *getCapabilities*, *describeFeatureType*, *getFeature*.

Min Long: -73.68363353662 Min Lat: -10.00074079297
 Max Long: -72.00056106773 Max Lat: -8.000372839583

Keywords: Unidades Taxônomicas, Solos, Banco de Dados Espacial, SIPAM, Zondende.

Abstract: Visão do banco de dados espacial de solos que fornece as informações das unidades taxônomicas associadas aos objetos geográficos de unidade de mapeamento.

WMS Path: /ut

Metadata URL	Type	Format	URL
	FGDC	text/plain	
	FGDC	text/plain	

Caching Enabled?:

Cache Time:

Schema Base: gml:AbstractFeatureType

geo: multiPolygonProperty nillable:true min:0 max:1

cd_simbolo_ut: string nillable:true min:0 max:1

cd_simbolo_unid: string nillable:true min:0 max:1

or_componente: decimal nillable:true min:0 max:1

cd_classe_solo: string nillable:true min:0 max:1

nm_classe_solo: string nillable:true min:0 max:1

cd_carater_1: string nillable:true min:0 max:1

cd_relevo_1: string nillable:true min:0 max:1

cd_relevo_2: string nillable:true min:0 max:1

cd_fertilidade_1: string nillable:true min:0 max:1

Figura A.10 Configuração do tipo de feição no *geoserver* – descrição e esquema de dados

A função *getCapabilities* descreve as características do servidor WFS fornecendo, em XML, informações como o nome, a descrição, palavras chave e a lista de tipos de feição cadastradas no servidor. Esta função também é responsável por informar quais serviços estão disponíveis no servidor. O endereço utilizado para executar a função *getCapabilities* é `http://<endereço>:<porta>/geoserver/WFS?request=getCapabilities`, onde:

- `http://<endereço>:<porta>/` é o endereço do servidor onde o *geoserver* está instalado e o serviço está publicado;
- *geoserver* é o nome do servidor;
- *WFS* é o serviço que está sendo solicitado;
- `request=getCapabilities` define a função que deverá ser executada.

A figura A.11 apresenta o retorno da operação *getCapabilities* no servidor SOG-SIPAM. Nota-se que o retorno da operação *getCapabilities* é uma descrição em formato XML das características do servidor que fornece os serviços OGC. Através desta função pode-se, por exemplo, identificar qual serviço encontra-se disponível para ser utilizado, ou quais tipos de feições podem ser requisitados pelo cliente ao servidor.

```

--<WFS_Capabilities version="1.0.0" xsi:schemaLocation="http://www.opengis.net/wfs
http://localhost:8080/geoserver/schemas/wfs/1.0.0/WFS-capabilities.xsd">
- <Service>
  <Name>SOG-IBGE</Name>
  <Title>
    Servidor WFS das informa??es espaciais de solos do IBGE
  </Title>
  <Abstract>
    Este servi?o fornece os objetos geogr?ficos de solos da regi?o amaz?nica armazenados dispon?veis no banco de dados
    de recursos naturais do IBGE (http://www.ibge.gov.br/servidor_arquivos_geo/).
  </Abstract>
  <Keywords>
    WFS , , WMS , , GEOSERVER , , SOLOS , , EMBRAPA , , IBGE , , SHAPEFILE , , ZONDENDE ,
  </Keywords>
  <OnlineResource>http://geoserver.sourceforge.net/html/index.php</OnlineResource>
  <Fees>NONE</Fees>
  <AccessConstraints>NONE</AccessConstraints>
</Service>
- <Capability>
+ <Request></Request>
</Capability>
- <FeatureTypeList>
+ <Operations></Operations>
- <FeatureType>
  <Name>sog_ibge:sb18_par</Name>
  <Title>Solos - Carta SB18</Title>
  <Abstract>
    Solos IBGE Carta SB18 Zondende. Banco de dados de recursos naturais Regi?o Amaz?nica
  </Abstract>
  <Keywords>
    Solos, Shapefile, IBGE, Banco de dados de Recursos Naturais, Zondende
  </Keywords>
  <SRS>EPSG:4618</SRS>
  <LatLongBoundingBox minx="-73.99105371353413" miny="-8.000040236669047" maxx="-71.99999894799302"
  maxy="-4.6216961790207725"/>
</FeatureType>
+ <FeatureType></FeatureType>
- <FeatureType>

```

Figura A.11 Retorno da opera?o *getCapabilities* no servidor SOG-SIPAM

A opera?o *describeFeatureType* fornece o esquema conceitual dos objetos geogr?ficos providos pelo servi?o WFS no formato de *XMLSchema*.

O esquema conceitual pode ser adquirido executando a opera?o *describeFeatureType* no endere?o `http://<endere?o>:<porta>/geoserver/wfs/DescribeFeatureType?typeName=<nome da fei?o>`, onde:

- `http://<endere?o>:<porta>/` ? o endere?o do servidor onde o *geoserver* est? instalado e o servi?o SOG-SIPAM est? publicado;
- *geoserver* ? o nome do servidor;
- *wfs* ? o servi?o que est? sendo solicitado;
- *describeFeatureType* ? o nome da opera?o que deve ser executada;
- *typeName=<nome da fei?o>* ? o nome do tipo de fei?o geogr?fica que deve ser retornado.

A figura A.12 apresenta o retorno da operação *describeFeatureType*, solicitada ao servidor SOG-SIPAM através do *browser*.

```

--<xs:schema targetNamespace="http://www.cnps.embrapa.br/zondende/sog_sipam" elementFormDefault="qualified"
attributeFormDefault="unqualified" version="1.0">
  <xs:import namespace="http://www.opengis.net/gml"
  schemaLocation="http://localhost:8080/geoserver/schemas/gml/2.1.2/feature.xsd"/>
  --<xs:complexType name="vw_ut_solos_aptidao_Type">
  --<xs:complexContent>
  --<xs:extension base="gml:AbstractFeatureType">
  --<xs:sequence>
  <xs:element name="geo" minOccurs="0" nillable="true" type="gml:MultiPolygonPropertyType"/>
  --<xs:element name="cd_simbolo_ut" minOccurs="0" nillable="true">
  --<xs:simpleType>
  --<xs:restriction base="xs:string">
  <xs:maxLength value="2147483647"/>
  </xs:restriction>
  </xs:simpleType>
  </xs:element>
  +<xs:element name="cd_simbolo_unid" minOccurs="0" nillable="true"></xs:element>
  <xs:element name="or_componente" minOccurs="0" nillable="true" type="xs:decimal"/>
  +<xs:element name="cd_classe_solo" minOccurs="0" nillable="true"></xs:element>
  +<xs:element name="nm_classe_solo" minOccurs="0" nillable="true"></xs:element>
  +<xs:element name="cd_carater_1" minOccurs="0" nillable="true"></xs:element>
  +<xs:element name="cd_relevo_1" minOccurs="0" nillable="true"></xs:element>
  +<xs:element name="cd_relevo_2" minOccurs="0" nillable="true"></xs:element>
  --<xs:element name="cd_fertilidade_1" minOccurs="0" nillable="true">
  --<xs:simpleType>
  --<xs:restriction base="xs:string">
  <xs:maxLength value="2147483647"/>
  </xs:restriction>
  </xs:simpleType>
  </xs:element>
  </xs:sequence>
  </xs:extension>
  </xs:complexContent>
  </xs:complexType>
  <xs:element name="vw_ut_solos_aptidao" type="sog_sipam:vw_ut_solos_aptidao_Type"
  substitutionGroup="gml:_Feature"/>
</xs:schema>

```

Figura A.12 Retorno da operação *DescribeFeatureType* solicitada ao servidor SOG-SIPAM.

Pode ser observado que a operação *describeFeatureType* retorna o esquema conceitual em *XMLSchema* (10) do tipo de feição geográfica disponível no servidor. No caso do SOG-IBGE, o tipo de feição geográfica foi configurado no *geoserver* com o nome de *sb18_par*. Este nome deve ser informado através do parâmetro *typename* no endereço utilizado para executar a operação.

Esta operação retorna a estrutura sintática dos objetos contidos na fonte de dados, fornecendo informações sobre os campos, tais como nome do campo, número mínimo de ocorrências, se aceita valores nulos, tipo do campo e tamanhos mínimo e máximo permitidos para o campo. A figura A.13 apresenta o retorno para o servidor SOG-IBGE.

```

- <xs:schema targetNamespace="http://www.cnps.embrapa.br/zondende/sog_ibge" elementFormDefault="qualified"
  attributeFormDefault="unqualified" version="1.0">
  <xs:import namespace="http://www.opengis.net/gml"
    schemaLocation="http://localhost:8080/geoserver/schemas/gml/2.1.2/feature.xsd"/>
  - <xs:complexType name="sb18_par_Type">
  - <xs:complexContent>
  - <xs:extension base="gml:AbstractFeatureType">
  - <xs:sequence>
  <xs:element name="the_geom" minOccurs="0" nillable="true" type="gml:MultiPolygonPropertyType"/>
  - <xs:element name="LETRA_SIMB" minOccurs="0" nillable="true">
  - <xs:simpleType>
  - <xs:restriction base="xs:string">
  <xs:maxLength value="8"/>
  </xs:restriction>
  </xs:simpleType>
  </xs:element>
  + <xs:element name="ORDEM" minOccurs="0" nillable="true"></xs:element>
  + <xs:element name="SUB_ORDEM" minOccurs="0" nillable="true"></xs:element>
  + <xs:element name="GDE_GRUPO" minOccurs="0" nillable="true"></xs:element>
  + <xs:element name="LEGENDA" minOccurs="0" nillable="true"></xs:element>
  + <xs:element name="SIMB_UNID" minOccurs="0" nillable="true"></xs:element>
  + <xs:element name="SUBST_ROCH" minOccurs="0" nillable="true"></xs:element>
  + <xs:element name="SUB_GRUPO" minOccurs="0" nillable="true"></xs:element>
  + <xs:element name="GRUP_TEXT" minOccurs="0" nillable="true"></xs:element>
  + <xs:element name="HOR_SUPERF" minOccurs="0" nillable="true"></xs:element>
  + <xs:element name="SAT_ALUM" minOccurs="0" nillable="true"></xs:element>
  + <xs:element name="FS_VEGET" minOccurs="0" nillable="true"></xs:element>
  + <xs:element name="FS_RELEVO" minOccurs="0" nillable="true"></xs:element>
  + <xs:element name="FS_PEDREG" minOccurs="0" nillable="true"></xs:element>
  + <xs:element name="FS_ROCH" minOccurs="0" nillable="true"></xs:element>
  + <xs:element name="FS_EROSAO" minOccurs="0" nillable="true"></xs:element>
  </xs:sequence>
  </xs:extension>
  </xs:complexContent>
  </xs:complexType>
  <xs:element name="sb18_par" type="sog_ibge:sb18_par_Type" substitutionGroup="gml:_Feature"/>
</xs:schema>

```

Figura A.13 Retorno da operação *describeFeatureType* no servidor SOG-IBGE

Com estas informações, o cliente poderá avaliar quais os tipos de objetos serão retornados na operação *getFeature*. A operação *getFeature* é a função que realmente fornece os objetos geográficos armazenados na fonte de dados, no formato definido pelo esquema conceitual descrito pela operação *describeFeatureType*. Para adquirir os objetos geográficos em formato GML é necessário executar a função *getFeature* através do endereço `http://<endereço>:<porta>/geoserver/wfs?request=getFeature&service=wfs&version=1.0.0&typename=<nome da feição>`, onde:

- `http://<endereço>:<porta>/` é o endereço do servidor onde o *geoserver* está instalado e o serviço está publicado;
- *geoserver* é o nome do servidor;
- *wfs* é o serviço que está sendo solicitado;
- *request=getFeature* é o nome da operação que deve ser executada;
- *service=wfs* é o serviço que será retornado;
- *version=1.0.0* é a versão da especificação do serviço WFS;


```

--<wfs:FeatureCollection xsi:schemaLocation="http://www.cnps.embrapa.br/zondende/sog_sipam
http://localhost:8080/geoserver/wfs/DescribeFeatureType?typeName=sog_sipam:vw_um_solos_aptidao
http://www.opengis.net/wfs http://localhost:8080/geoserver/schemas/wfs/1.0.0/WFS-basic.xsd">
--<gml:boundedBy>
--<gml:Box srsName="http://www.opengis.net/gml/srs/epsg.xml#4618">
--<gml:coordinates decimal="." cs="," ts=" " >-73.68305939,-10.00036069
-71.99999644,-8.00000049</gml:coordinates>
</gml:Box>
</gml:boundedBy>
--<gml:featureMember>
--<sog_sipam:vw_um_solos_aptidao fid="vw_um_solos_aptidao.nfm--39313e7c_11364f06f51_-7f1d">
--<sog_sipam:geo>
--<gml:MultiPolygon srsName="http://www.opengis.net/gml/srs/epsg.xml#4618">
--<gml:polygonMember>
--<gml:Polygon>
--<gml:outerBoundaryIs>
--<gml:LinearRing>
--<gml:coordinates decimal="." cs="," ts=" " >
-73.68305939,-8.000004 -73.65876223,-8.00000893 -73.64860368,-8.00001148
-73.64959613,-8.00112742 -73.65109913,-8.00136861 -73.65338677,-8.00087805
-73.65482313,-8.00106189 -73.65545357,-8.00146939 -73.65802237,-8.00401765
-73.65859425,-8.004425 -73.66066411,-8.00455779 -73.66314661,-8.00331705
-73.6645279,-8.00321353 -73.66516,-8.00333388 -73.66733267,-8.00513247 -73.66995336,-8.00854317
-73.67140733,-8.01082404 -73.67427671,-8.00960913 -73.67557359,-8.00906028
-73.68005521,-8.00554081 -73.68305939,-8.000004
</gml:coordinates>
</gml:LinearRing>
</gml:outerBoundaryIs>
</gml:Polygon>
</gml:polygonMember>
</gml:MultiPolygon>
</sog_sipam:geo>
<sog_sipam:cd_simbolo_unid>SC18PVe6</sog_sipam:cd_simbolo_unid>
<sog_sipam:total_componentes>3</sog_sipam:total_componentes>
</sog_sipam:vw_um_solos_aptidao>
</gml:featureMember>
--<gml:featureMember>
--<sog_sipam:vw_um_solos_aptidao fid="vw_um_solos_aptidao.nfm--39313e7c_11364f06f51_-7f1c">

```

Figura A.15 Feição geográfica representada em GML (retorno da função *getFeature*)

Apêndice B - Documentação da ontologia de atributos do solo para aptidão agrícola

A ontologia desenvolvida nesta dissertação foi utilizada como esquema de referência na integração dos dados espaciais de solos. Esta ontologia representa o domínio dos atributos do solo relevantes para a avaliação da aptidão agrícola das terras para uma determinada cultura. A seguir, são apresentados os elementos desta representação do conhecimento, modelado em linguagem Ontology Web Language (OWL).

B.1 Unidade Taxonômica

Classe: UnidadeTaxonomica

"É conceituada segundo um conjunto de características e propriedades pedológicas conhecidas por meio de estudos de pedons e polipedons e corresponde à unidade de classificação mais homogênea em qualquer nível categórico de sistemas taxonômicos."
[lang: pt]

owl:Thing
UnidadeTaxonomica

Super-classes:

owl:Thing

Indivíduos:

UnidadeTaxonomica_83	UnidadeTaxonomica_62	UnidadeTaxonomica_129
UnidadeTaxonomica_125	UnidadeTaxonomica_122	UnidadeTaxonomica_108
UnidadeTaxonomica_100	UnidadeTaxonomica_124	UnidadeTaxonomica_119
UnidadeTaxonomica_107	UnidadeTaxonomica_123	UnidadeTaxonomica_118
UnidadeTaxonomica_106	UnidadeTaxonomica_130	UnidadeTaxonomica_105
UnidadeTaxonomica_103	UnidadeTaxonomica_101	UnidadeTaxonomica_104
UnidadeTaxonomica_102	UnidadeTaxonomica_120	UnidadeTaxonomica_121
UnidadeTaxonomica_115	UnidadeTaxonomica_110	UnidadeTaxonomica_109
UnidadeTaxonomica_128	UnidadeTaxonomica_127	UnidadeTaxonomica_113
UnidadeTaxonomica_117	UnidadeTaxonomica_126	UnidadeTaxonomica_112
UnidadeTaxonomica_131	UnidadeTaxonomica_116	UnidadeTaxonomica_114
UnidadeTaxonomica_111	UnidadeTaxonomica_52	UnidadeTaxonomica_95
UnidadeTaxonomica_13	UnidadeTaxonomica_69	UnidadeTaxonomica_27
UnidadeTaxonomica_91	UnidadeTaxonomica_3	UnidadeTaxonomica_80
UnidadeTaxonomica_72	UnidadeTaxonomica_81	UnidadeTaxonomica_12
UnidadeTaxonomica_32	UnidadeTaxonomica_19	UnidadeTaxonomica_18
UnidadeTaxonomica_5	UnidadeTaxonomica_25	UnidadeTaxonomica_8
UnidadeTaxonomica_7	UnidadeTaxonomica_9	UnidadeTaxonomica_74

UnidadeTaxonomica_4	UnidadeTaxonomica_10	UnidadeTaxonomica_75
UnidadeTaxonomica_71	UnidadeTaxonomica_23	UnidadeTaxonomica_11
UnidadeTaxonomica_84	UnidadeTaxonomica_73	UnidadeTaxonomica_58
UnidadeTaxonomica_41	UnidadeTaxonomica_89	UnidadeTaxonomica_50
UnidadeTaxonomica_53	UnidadeTaxonomica_99	UnidadeTaxonomica_63
UnidadeTaxonomica_36	UnidadeTaxonomica_57	UnidadeTaxonomica_6
UnidadeTaxonomica_17	UnidadeTaxonomica_15	UnidadeTaxonomica_31
UnidadeTaxonomica_76	UnidadeTaxonomica_39	UnidadeTaxonomica_37
UnidadeTaxonomica_98	UnidadeTaxonomica_65	UnidadeTaxonomica_35
UnidadeTaxonomica_93	UnidadeTaxonomica_56	UnidadeTaxonomica_16
UnidadeTaxonomica_14	UnidadeTaxonomica_30	UnidadeTaxonomica_38
UnidadeTaxonomica_21	UnidadeTaxonomica_97	UnidadeTaxonomica_40
UnidadeTaxonomica_61	UnidadeTaxonomica_34	UnidadeTaxonomica_55
UnidadeTaxonomica_49	UnidadeTaxonomica_77	UnidadeTaxonomica_20
UnidadeTaxonomica_90	UnidadeTaxonomica_28	UnidadeTaxonomica_59
UnidadeTaxonomica_60	UnidadeTaxonomica_51	UnidadeTaxonomica_54
UnidadeTaxonomica_24	UnidadeTaxonomica_64	UnidadeTaxonomica_1
UnidadeTaxonomica_92	UnidadeTaxonomica_26	UnidadeTaxonomica_29
UnidadeTaxonomica_78	UnidadeTaxonomica_79	UnidadeTaxonomica_87
UnidadeTaxonomica_88	UnidadeTaxonomica_2	UnidadeTaxonomica_82
UnidadeTaxonomica_44	UnidadeTaxonomica_33	UnidadeTaxonomica_45
UnidadeTaxonomica_46	UnidadeTaxonomica_42	UnidadeTaxonomica_47
UnidadeTaxonomica_43	UnidadeTaxonomica_48	UnidadeTaxonomica_86
UnidadeTaxonomica_85	UnidadeTaxonomica_70	UnidadeTaxonomica_68
UnidadeTaxonomica_67	UnidadeTaxonomica_22	UnidadeTaxonomica_66
UnidadeTaxonomica_96	UnidadeTaxonomica_94	

Domínio de (propriedades):

pertenceUnidadeMapeamento	possuiCarater	possuiRelevoLocal
temDrenagem	temOrdemProporcionalidade	temPedregosidade
temPercentualNaUnidadeMapeamentoDe	temProfundidade	temRochosidade
temSaturacaoBase	temSimbolo	temSolo
temTextura		

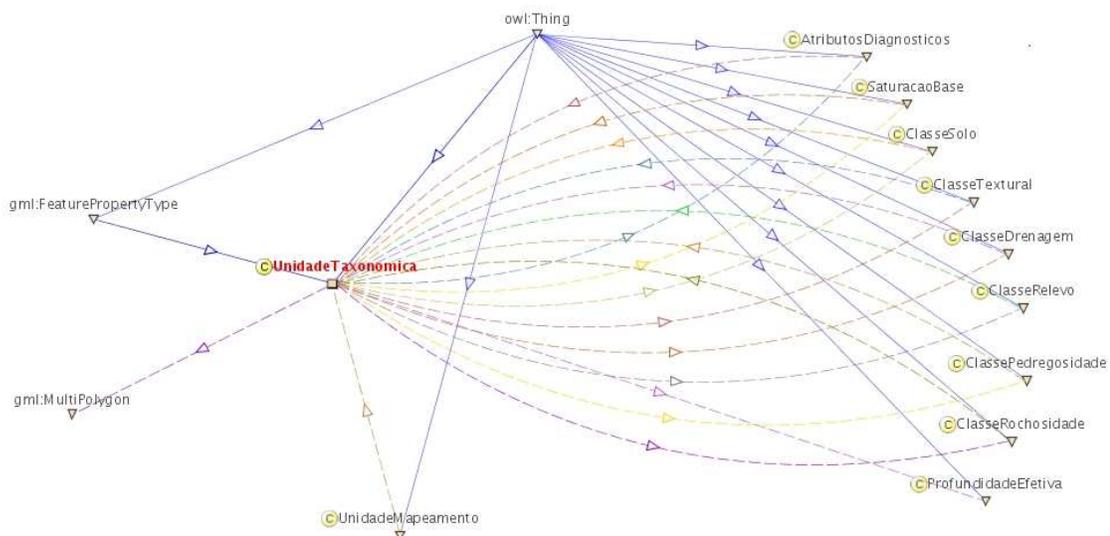


Figura B.1 Unidade Taxonômica

B.2 Unidade de Mapeamento

Classe: UnidadeMapeamento

"São áreas de solos definidas em função das unidades taxonômicas que a compõem." [lang: pt]

owl:Thing
 gml:FeaturePropertyType
 UnidadeMapeamento
 UnidadeMapeamento

Super-classes:

gml:FeaturePropertyType
 owl:Thing

Indivíduos:

UnidadeMapeamento_21	UnidadeMapeamento_20	UnidadeMapeamento_13
UnidadeMapeamento_23	UnidadeMapeamento_12	UnidadeMapeamento_22
UnidadeMapeamento_11	UnidadeMapeamento_25	UnidadeMapeamento_10
UnidadeMapeamento_24	UnidadeMapeamento_27	UnidadeMapeamento_17
UnidadeMapeamento_26	UnidadeMapeamento_16	UnidadeMapeamento_29
UnidadeMapeamento_15	UnidadeMapeamento_28	UnidadeMapeamento_14
UnidadeMapeamento_38	UnidadeMapeamento_1	UnidadeMapeamento_39
UnidadeMapeamento_2	UnidadeMapeamento_18	UnidadeMapeamento_19
UnidadeMapeamento_34	UnidadeMapeamento_48	UnidadeMapeamento_35
UnidadeMapeamento_36	UnidadeMapeamento_46	UnidadeMapeamento_37
UnidadeMapeamento_47	UnidadeMapeamento_44	UnidadeMapeamento_9
UnidadeMapeamento_30	UnidadeMapeamento_45	UnidadeMapeamento_31

UnidadeMapeamento_41	UnidadeMapeamento_32	UnidadeMapeamento_7
UnidadeMapeamento_42	UnidadeMapeamento_43	UnidadeMapeamento_8
UnidadeMapeamento_33	UnidadeMapeamento_5	UnidadeMapeamento_40
UnidadeMapeamento_6	UnidadeMapeamento_3	UnidadeMapeamento_4

Domínio de (propriedades):

temGeometria temQuantidadeUnidadesTaxonomicas
temSimbolo temUnidadeTaxonomica



Figura B.2 Unidade de Mapeamento

B.3 Saturação de Bases

Classe: SaturacaoBase

"Refere-se à proporção (taxa percentual, V%=100. S/T) de cátions básicos trocáveis em relação à capacidade de troca determinada a pH 7." [lang: pt]

owl:Thing
SaturacaoBase

Super-classes:

owl:Thing

Indivíduos:

Eutrofico	Distrofico	Alico
-----------	------------	-------

Domínio de (propriedades):

ehSaturacaoBaseDe

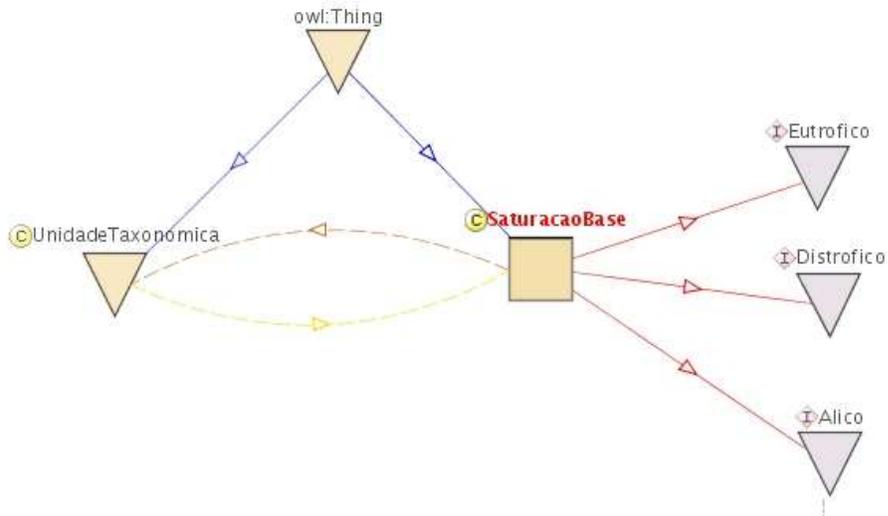


Figura B.3 Saturação de Bases

B.4 Classe de Relevô

Classe: ClasseRelevo

Qualificam condições de declividade, comprimento de encostas e configuração superficial dos terrenos, que afetam as formas de modelado (formas topográficas) de áreas de ocorrência das unidades de solo.

owl:Thing
 ClasseRelevo

Super-classes:

owl:Thing

Indivíduos

RelevoOndulado	RelevoPlano	RelevoForteOndulado
RelevoSuaveOndulado	RelevoEscarpado	RelevoMontanhoso

Domínio de (propriedades):

ehRelevoLocalDe

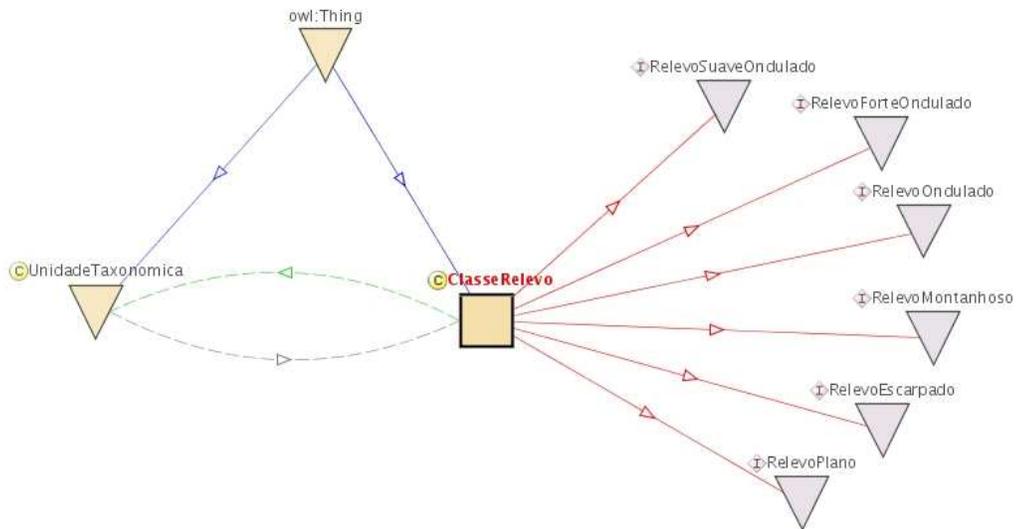


Figura B.4 Classe de Relevo

B.5 Classe de Drenagem

Classe: ClasseDrenagem

"Referem-se à quantidade e rapidez com que a água recebida pelo solo se escoar por infiltração e escoamento, afetando as condições hídricas do solo - duração de período em que permanece úmido, molhado ou encharcado." [lang: pt]

owl:Thing

ClasseDrenagem

Super-classes:

owl:Thing

Indivíduos:

FortementeDrenado	MuitoMalDrenado	ImperfeitamenteDrenado
ModeradamenteDrenado	ExcessivamenteDrenado	BemDrenado
AcentuadamenteDrenado	MalDrenado	

Domínio de (propriedades):

ehDrenagemDe

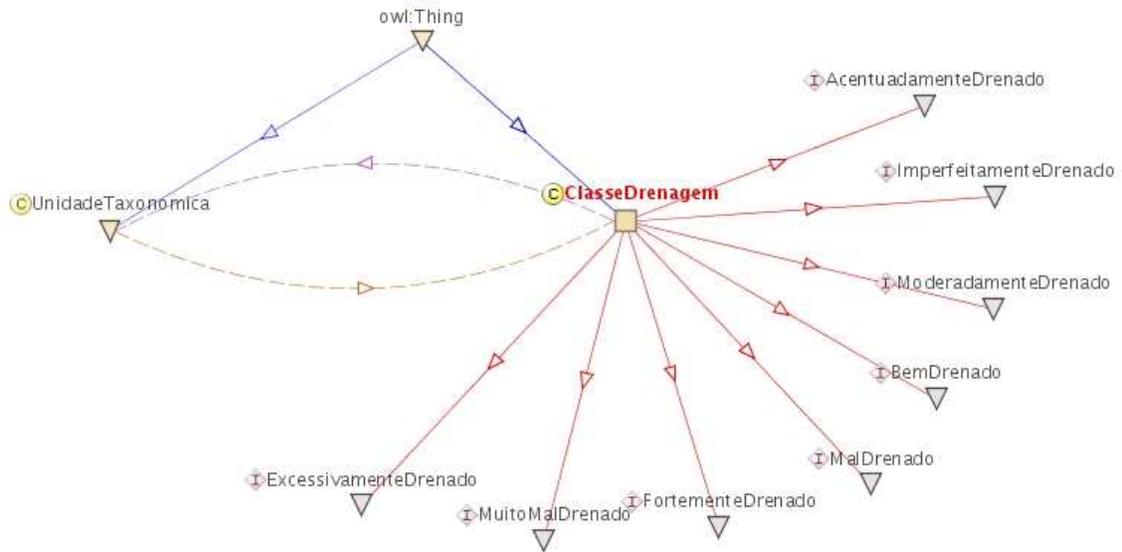


Figura B.5 Classe de Drenagem

B.6 Classe de Pedregosidade

Classe: ClassePedregosidade

"Qualificam áreas em que a presença superficial ou subsuperficial de quantidades expressivas de calhaus (2 a 20cm de diâmetro) e matações (20 a 100cm de diâmetro) interfere no uso das terras, sobretudo no referente ao emprego de máquinas e equipamentos agrícolas, ou seja, 3% ou mais de material macroclástico em apreço." [lang: pt]

owl:Thing
ClassePedregosidade

Super-classes:

owl:Thing

Indivíduos:

Pedregosa	ExtremamentePedregosa	ModeradamentePedregosa
LigeiramentePedregosa	MuitoPedregosa	NaoPedregosa

Domínio de (propriedades):

ehPedregosidadeDe

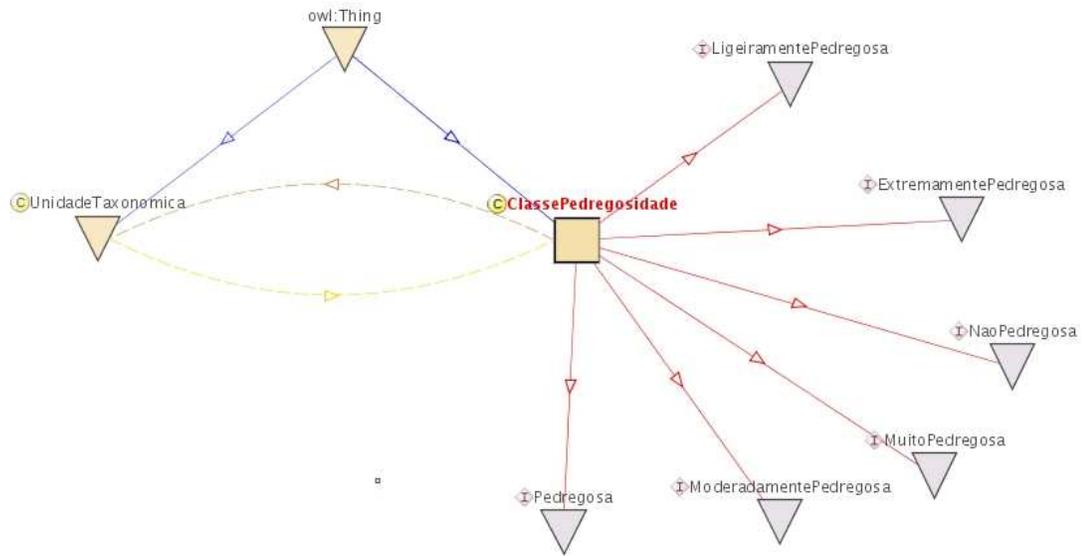


Figura B.6 Classe de Pedregosidade

B.7 Classe de Rochosidade

Classe: ClasseRochosidade

"Refere-se à exposição do substrato rochoso, lajes de rochas, parcelas de camadas delgadas de solos sobre rochas e/ou predominância de matações (?boulders?) com diâmetro médio maior que 100cm, na superfície ou na massa do solo, em quantidades tais, que tornam impraticável o uso de máquinas agrícolas." [lang: pt]

owl:Thing
ClasseRochosidade

Super-classes:

owl:Thing

Indivíduos:

MuitoRochosa	NaoRochosa	LigeiramenteRochosa
Rochosa	ExtremamenteRochosa	ModeradamenteRochosa

Domínio de (propriedades):

ehRochosidadeDe

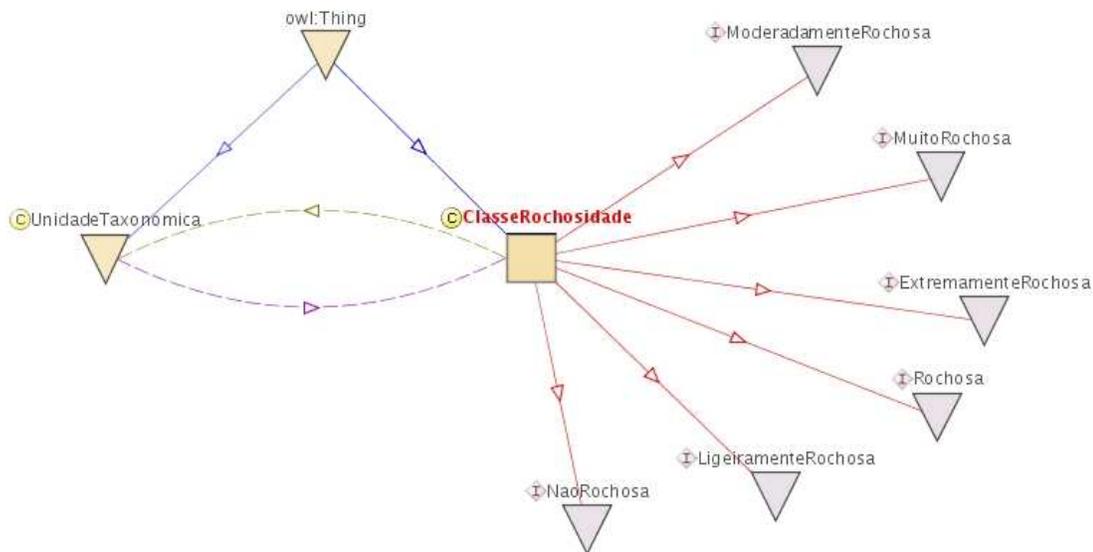


Figura B.7 Classe de Rochosidade

B.8 Classe Textural

Classe: ClasseTextural

"Refere-se à composição granulométrica, ou seja, a quantidade percentual das frações argila (0,05 à 2 mm) de uma determinada amostra de solo." [lang: pt]

owl:Thing
ClasseTextural

Super-classes:

owl:Thing

Indivíduos:

ClasseTexturalFrancoArenoso	ClasseTexturalFrancoArgiloso
ClasseTexturalArgilosa	ClasseTexturalMuitoArgilosa
ClasseTexturalFrancoSiltoso	ClasseTexturalFrancoArgiloArenoso
ClasseTexturalFrancoArgiloSiltoso	ClasseTexturalAreia
ClasseTexturalSilte	ClasseTexturalArgilaArenosa
ClasseTexturalAreiaFranca	ClasseTexturalFranco
ClasseTexturalArgilaSiltosa	

Domínio de (propriedades):

ehTexturaDe

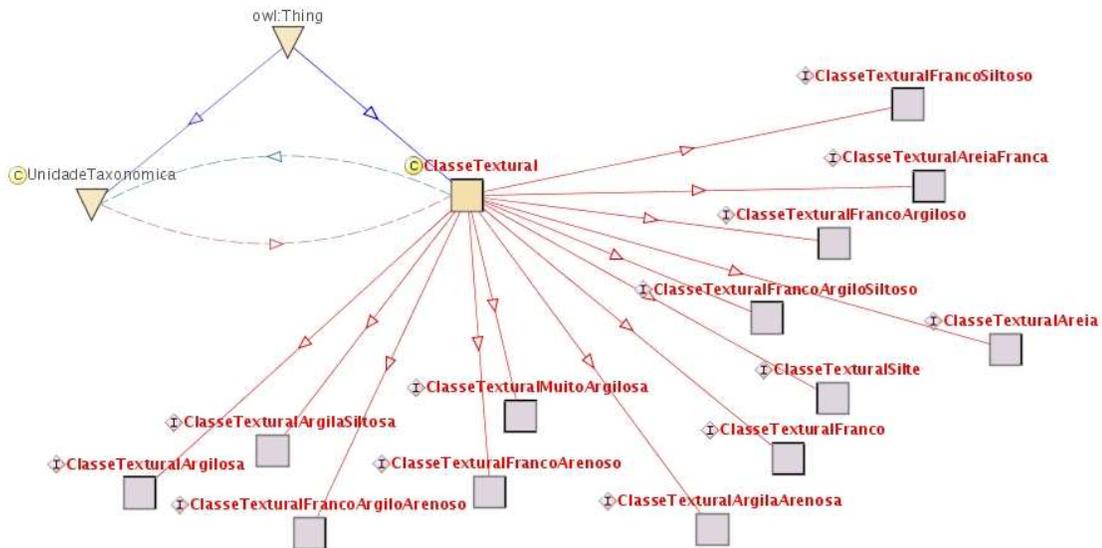


Figura B.8 Classe Textural

B.9 Atributos Diagnósticos

Classe: AtributosDiagnosticos

"São atributos indicativos de características pedológicas distintas, referentes à características físicas, químicas e mineralógicas dos solos e/ou seus horizontes, tendo função diagnóstica para fins de classificação." [lang: pt]

owl:Thing
AtributosDiagnosticos

Super-classes:

owl:Thing

Indivíduos:

CaraterSolodico	CaraterPlintico	CaraterComCarbonato
SaturacaoporBases	CaraterCromico	CaraterAluminico
Petroplintita	CaraterSalico	TeordeOxidosdeFerro
CaraterEutrico	AtividadedaFracaoArgila	MaterialOrganicoSaprico
SuperficiesdeFriccao	CaraterVertico	CaraterEpiquico
CaraterTiomorfico	CaraterCalcico	CaraterPlanico
CaraterLitoplintico	CaraterCarbonatico	MateriaisSulfidricos
CaraterAcrico	MaterialOrganicoHemico	CaraterAlitico
CaraterCoeso	CaraterSalino	CaraterEbanico
CaraterRubrico	CaraterSodico	MaterialOrganico
CaraterArgiluvico	ContatoLiticoFragmentario	CaraterFluvico
CaraterConcrecionario	MaterialMineral	ContatoLitico
Plintita	CaraterDurico	MudancaTexturalAbrupta
MaterialOrganicoFibrico		

Domínio de (propriedades):

ehCaraterDe

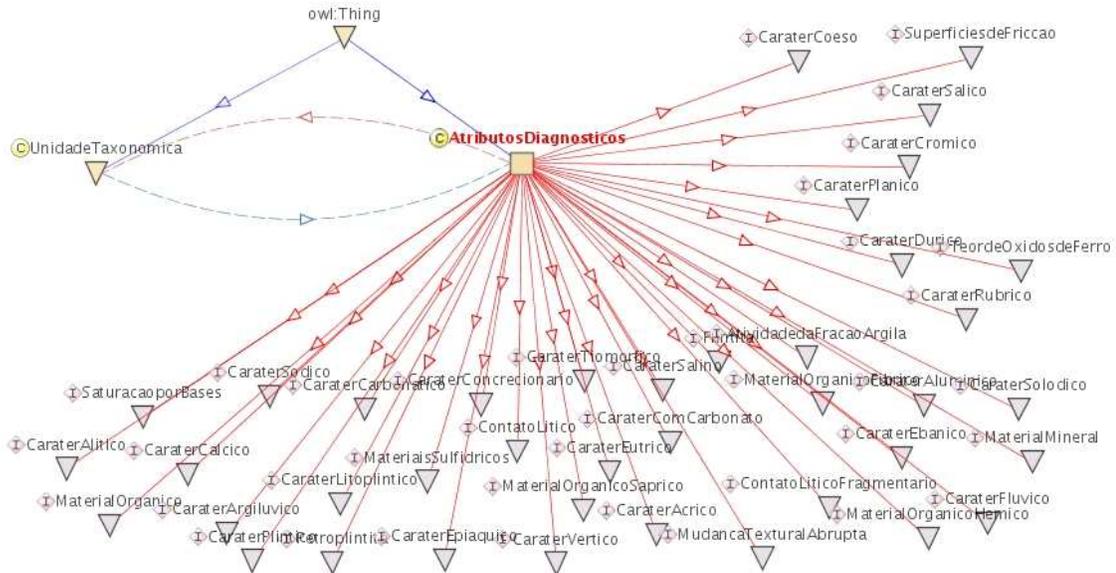


Figura B.9 Atributos Diagnosticos

B.10 Classe de Solo

Classe: ClasseSolo

"É definida como um grupamento de indivíduos semelhantes em características diagnósticas de natureza morfológica, física, química e mineralógica, com apoio de um sistema taxonômico organizado, constituindo-se a unidade fundamental na composição de unidades de mapeamento." [lang: pt]

owl:Thing
ClasseSolo

Super-classes:

owl:Thing

Indivíduos:

Podzolico Bruno Acinzentado
Terra Bruna Estruturada Intermediária para Terra Roxa Estruturada
Solo Aluvial
Gleissolo
Podzolico Vermelho Escuro
Areia Quartzosa
Podzolico Vermelho Amarelo

Cambissolo
LatossoloVermelhoAmarelo
LatossoloAmarelo
Vertissolo
Plintossolo
PodzolHidromorfico
Planossolo
CambissoloHistico
GleiHumico
GleissoloHumicoVertico
Podzol
SoloIndiscriminadoMangueManguezal
LatossoloAmareloHumico
PodzolicoAmareloLatossolico
CambissoloHumico
LatossoloBrunoIntermediarioParaLatossoloRoxo
SoloTiomorficoHumico
LatossoloAmareloPodzolico
Regossolo
LatossoloVermelhoEscuro
GleissoloVertico
Solonchak
SoloLitologicoSubstratoRochasFerriferas
LatossoloVermelhoAmareloHumico
CambissoloSubstratoRochasFerriferas
SoloTiomorfico
AreiaQuartzosaMarinhaHidromorfica
TerraRoxaEstruturadaLatossolica
SoloPetroplintico
SolosHidromorficosIndiscriminados
CambissoloIntermediarioParaPodzolicoBrunoAcinzentado
SolonetzSolodizado
SoloOrganicoTiomorfico
GleissoloTiomorfico
AreiaQuartzosaHidromorficaHumica
PodzolicoVermelhoEscuroLatossolico
TerraBrunaEstruturadaSimilar
TerraBrunaEstruturadaHumica
LatossoloFerriferoHumico
PodzoisIndiscriminados
Rendzina
LatossoloFerrifero
CambissoloLatossolico
Brunizem
CambissoloHaplico
BrunizemAvermelhado
TerraBrunaEstruturadaIntermediariaParaPodzolicoBrunoAcinzentadoHumico
TerraBrunaEstruturadaLatossolica
LatossoloVariacaoUnaHumico
TerraVermelhoBrunadaEstruturada
AreiaQuartzosaHidromorfica

LatossoloBrunoHumico
LatossoloVariacaoUna
PodzolicoAmarelo
SoloPetroplinticoIndiscriminado
LatossoloVermelhoEscuroPodzolico
SoloLitolico
Petroplintitas
LatossoloBruno
TerraVermelhoBrunadaEstruturadaHumica
PodzolicoAcinzentado
TerraBrunaEstruturadaIntermediariaParaPodzolicoVermelhoEscuro
GleissoloHumicoSolodico
LatossoloBrunoIntermediarioParaLatossoloRoxoHumico
SoloOrganico
PlanossoloVertico
AfloramentoRocha
LatossoloRoxo
LatossoloVermelhoAmareloPodzolico
GleissoloSolodico
BrunoNaoCalcicoPlanossolico
PlanossoloSolodico
VertissoloSolodico
PodzolicoBrunoAcinzentadoPlanossolico
TerraRoxaEstruturada
BrunizemVertico
Rubrozem
CambissoloBruno
CambissoloBrunoHumico
AreiaQuartzosaMarinha
SolonchakSodico
HidromorficoCinzento
SoloLitolicoHumico
TerraBrunaEstruturada
SolosTiomorficosIndiscriminados
LatossoloVermelhoEscuroHumico
GleiPoucoHumico
Dunas
LatossoloBrunoCambico
LatossoloVermelhoEscuroCambico
PlintossoloHumico
BrunoNaoCalcico
Solonetz
PodzolicoVermelhoAmareloLatossolico
GleissoloHumico

Domínio de (propriedades):

formaUnidadeMapeamento

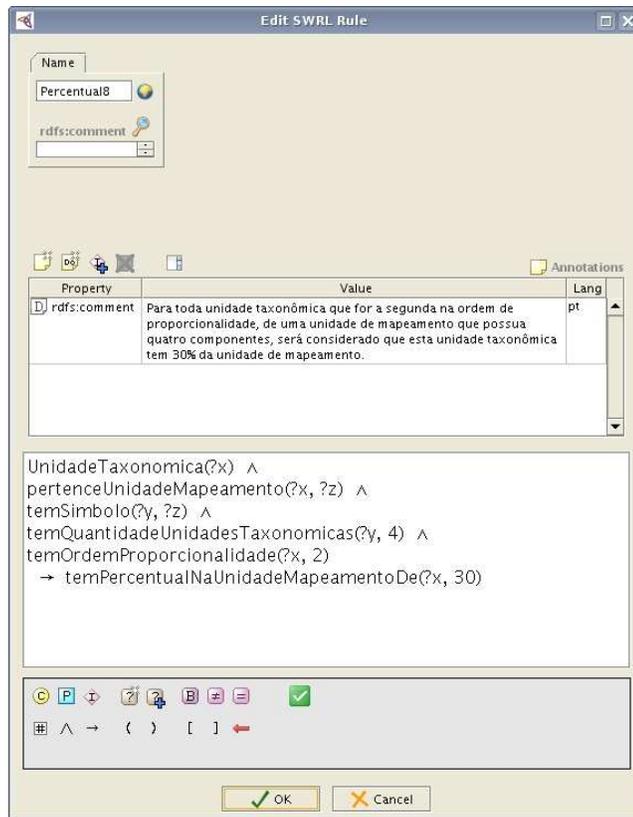


Figura B.11 Regra de percentual da unidade taxonômica na unidade de mapeamento

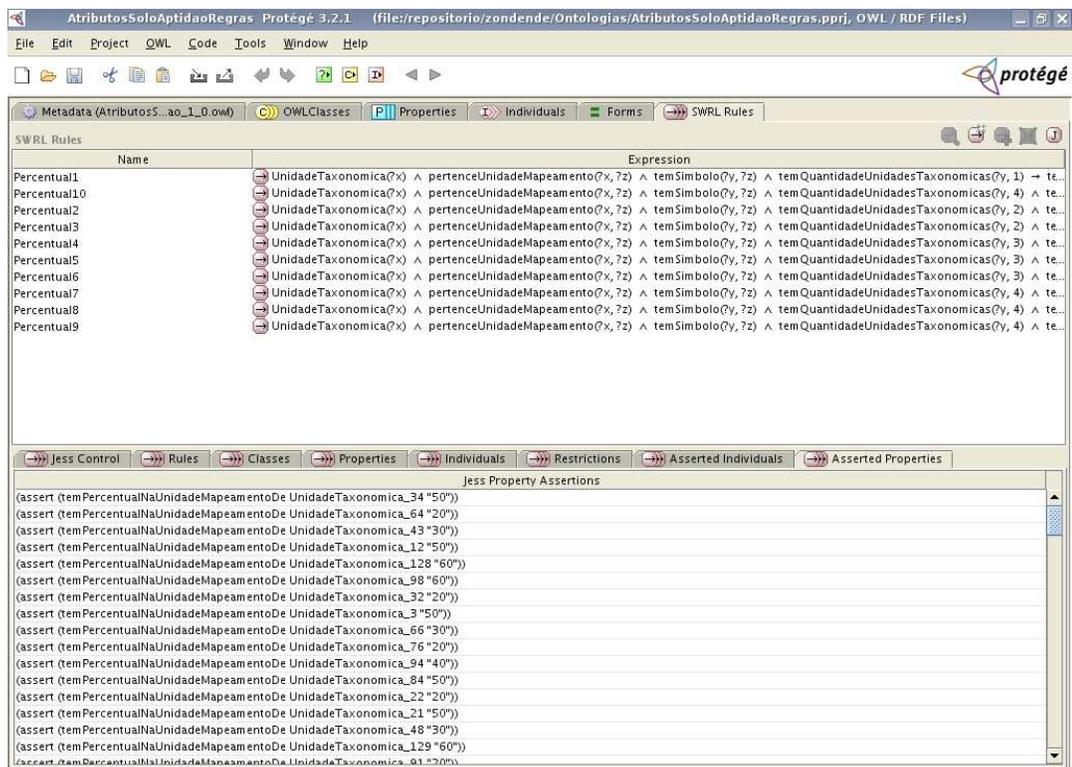


Figura B.12 Conjunto de regras e valores inferidos

Apêndice C - Documentação da biblioteca de interoperabilidade entre objetos geográficos baseada em ontologias (OpenOGI)

OpenOGI 1.0

Packages	
br.uerj.eng.geomatica.interoperability.exception	Contém as classes de exceção da biblioteca.
br.uerj.eng.geomatica.interoperability.model	Este pacote contém as classes fundamentais que definem o modelo de interoperabilidade sintática e semântica entre feições geográficas.
br.uerj.eng.geomatica.interoperability.service	Este pacote contém os serviços disponíveis na biblioteca para interagir com as fontes de dados, os domínios, as ontologias, os serviços WFS, etc.
br.uerj.eng.geomatica.interoperability.unitTest	Contém os testes unitários (junit tests) que validam as classes da biblioteca.

C.1 Pacote de tratamento de Exceção

Package **br.uerj.eng.geomatica.interoperability.exception**

Contém as classes de exceção da biblioteca.

See:

[Description](#)

Exception Summary	
DataSourceException	Exceção disparada quando ocorre algum problema relacionado a configuração ou uso das fontes de dados.
ObjectNotValidException	Exceção disparada quando um objeto não é válido.
XMLObjectNotFoundException	Exceção disparada quando um objeto não é encontrado no XML de configuração.
XMLStoreException	Exceção disparada tentar armazenar um objeto em formato XML.

Package **br.uerj.eng.geomatica.interoperability.exception** Description

Contém as classes de exceção da biblioteca.

Since:

1.0

Author:

Victor Azevedo (reference Azevedo, V. H. M.)

See Also:

Exception

C.2 Classe DataSourceException

`br.uerj.eng.geomatica.interoperability.exception`

Class DataSourceException

```
java.lang.Object
    java.lang.Throwable
        java.lang.Exception
br.uerj.eng.geomatica.interoperability.exception.DataSourceException
```

All Implemented Interfaces:

java.io.Serializable

```
public class DataSourceException extends java.lang.Exception
```

Exceção disparada quando ocorre algum problema relacionado a configuração ou uso das fontes de dados.

Since:

1.0

Author:

Victor Azevedo (reference Azevedo, V. H. M.)

See Also:

[Serialized Form](#)

Constructor Summary	
DataSourceException (java.lang.String ex)	
Construtor da classe DataSourceException	

Method Summary

Methods inherited from class java.lang.Throwable
<code>fillInStackTrace</code> , <code>getCause</code> , <code>getLocalizedMessage</code> , <code>getMessage</code> , <code>getStackTrace</code> , <code>initCause</code> , <code>printStackTrace</code> , <code>printStackTrace</code> , <code>printStackTrace</code> , <code>setStackTrace</code> , <code>toString</code>

Methods inherited from class java.lang.Object
<code>equals</code> , <code>getClass</code> , <code>hashCode</code> , <code>notify</code> , <code>notifyAll</code> , <code>wait</code> , <code>wait</code> , <code>wait</code>

Constructor Detail

DataSourceException

```
public DataSourceException(java.lang.String ex)
```

Construtor da classe DataSourceException

Parameters:

ex - mensagem de exceção

C.3 Classe ObjectNotValidException

br.uerj.eng.geomatica.interoperability.exception

Class ObjectNotValidException

```
java.lang.Object
    java.lang.Throwable
        java.lang.Exception
            br.uerj.eng.geomatica.interoperability.exception.ObjectNotValidException
```

All Implemented Interfaces:

java.io.Serializable

```
public class ObjectNotValidException extends java.lang.Exception
```

Exceção disparada quando um objeto não é válido.

Since:

1.0

Author:

Victor Azevedo (reference Azevedo, V. H. M.)

See Also:

[Serialized Form](#)

Constructor Summary

```
ObjectNotValidException(java.lang.String ex)  
    Construtor da classe ObjectNotValidException
```

Method Summary

Methods inherited from class java.lang.Throwable

```
fillInStackTrace, getCause, getLocalizedMessage, getMessage,  
getStackTrace, initCause, printStackTrace, printStackTrace,  
printStackTrace, setStackTrace, toString
```

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

ObjectNotValidException

```
public ObjectNotValidException(java.lang.String ex)
```

Construtor da classe ObjectNotValidException

Parameters:

ex - mensagem de exceção

C.4 Classe XMLObjectNotFoundException

br.uerj.eng.geomatica.interoperability.exception

Class XMLObjectNotFoundException

```
java.lang.Object
    java.lang.Throwable
        java.lang.Exception
            br.uerj.eng.geomatica.interoperability.exception.XMLObjectNotFoundException
```

All Implemented Interfaces:

java.io.Serializable

```
public class XMLObjectNotFoundException extends java.lang.Exception
```

Exceção disparada quando um objeto não é encontrado no XML de configuração.

Since:

1.0

Author:

Victor Azevedo (reference Azevedo, V. H. M.)

See Also:

[Serialized Form](#)

Constructor Summary

```
XMLObjectNotFoundException(java.lang.String ex)
```

Construtor da classe XMLObjectNotFoundException

Method Summary

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace`, `getCause`, `getLocalizedMessage`, `getMessage`,
`getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`,
`printStackTrace`, `setStackTrace`, `toString`

Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`,
`wait`

Constructor Detail

XMLObjectNotFoundException

```
public XMLObjectNotFoundException(java.lang.String ex)
```

Construtor da classe `XMLObjectNotFoundException`

Parameters:

`ex` - mensagem de exceção

C.5 Classe `XMLStoreException`

`br.uerj.eng.geomatica.interoperability.exception`

Class `XMLStoreException`

```
java.lang.Object  
    java.lang.Throwable  
        java.lang.Exception  
br.uerj.eng.geomatica.interoperability.exception.XMLStoreException
```

All Implemented Interfaces:

`java.io.Serializable`

```
public class XMLStoreException extends java.lang.Exception
```

Exceção disparada tentar armazenar um objeto em formato XML.

Since:

1.0

Author:

Victor Azevedo (reference Azevedo, V. H. M.)

See Also:

[Serialized Form](#)

Constructor Summary

[XMLStoreException](#)(java.lang.String ex)
Construtor da classe XMLStoreException

Method Summary

Methods inherited from class java.lang.Throwable

fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

XMLStoreException

```
public XMLStoreException(java.lang.String ex)
```

Construtor da classe XMLStoreException

Parameters:

ex - mensagem de exceção

C.6 Pacote Model

Package br.uerj.eng.geomatica.interoperability.model

Este pacote contém as classes fundamentais que definem o modelo de interoperabilidade sintática e semântica entre feições geográficas.

See:

[Description](#)

Class Summary

DataSource	Representa uma fonte de dados
Domain	Representa um domínio de interesse.
GeographicInformationLayer	Representa uma camada de informação geográfica baseada em um domínio de interesse e que armazena informações de um conjunto de fontes de dados associadas a este domínio

Metadata	Conjunto de metadados geográficos que identificam cada fonte de dados.
SchemaCorrelation	Representa a correlação entre um termo do esquema da fonte de dados e uma propriedade da classe que representa a feição geográfica na ontologia (esquema de referência).
SchemaElement	Representa um elemento de esquema conceitual.
WFSDataSource	Representa uma fonte de dados do tipo WFS

Package **br.uerj.eng.geomatica.interoperability.model** Description

Este pacote contém as classes fundamentais que definem o modelo de interoperabilidade sintática e semântica entre feições geográficas.

Since:

1.0

Author:

Victor Azevedo (reference Azevedo, V. H. M.)

C.7 Classe [DataSource](#)

br.uerj.eng.geomatica.interoperability.model

Class DataSource

`java.lang.Object`

`br.uerj.eng.geomatica.interoperability.model.DataSource`

Direct Known Subclasses:

[WFSDataSource](#)

```
public class DataSource extends java.lang.Object
```

Representa uma fonte de dados

Since:

1.0

Author:

Victor Azevedo (reference Azevedo, V. H. M.)

Constructor Summary	
DataSource ()	Cria uma nova instância de DataSource
DataSource (int code)	Cria uma nova instância de DataSource
DataSource (int code, java.lang.String name)	

Cria uma nova instância de DataSource

Method Summary	
boolean	equals (java.lang.Object obj)
int	getCode () Retorna o código da fonte de dados
java.util.Collection < SchemaCorrelation >	getCorrelationTable () Retorna a tabela de correlação entre o esquema da fonte de dados e o esquema conceitual.
Domain	getDomain () Retorna o domínio de interesse associado à fonte de dados
Metadata	getMetadata () Retorna os metadados geográficos associados à fonte de dados
java.lang.String	getName () Retorna o nome da fonte de dados
int	hashCode ()
void	setCode (int code) Cadastra o código da fonte de dados
void	setCorrelationTable (java.util.Collection < SchemaCorrelation > correlationTable) Cadastra a tabela de correlação entre o esquema da fonte de dados e o esquema conceitual.
void	setDomain (Domain domain) Associa um domínio de interesse à fonte de dados.
void	setMetadata (Metadata metadata) Cadastra os metadados geográficos da fonte de dados
void	setName (java.lang.String name) Cadastra o nome da fonte de dados

Methods inherited from class java.lang.Object

`getClass, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

DataSource

```
public DataSource()
```

Cria uma nova instância de DataSource

DataSource

```
public DataSource(int code, java.lang.String name)
```

Cria uma nova instância de DataSource

Parameters:

code - código da fonte de dados

name - nome da fonte de dados

DataSource

```
public DataSource(int code)
```

Cria uma nova instância de DataSource

Parameters:

code - código da fonte de dados

Method Detail

getCode

```
public int getCode()
```

Retorna o código da fonte de dados

Returns:

int código da fonte de dados

setCode

```
public void setCode(int code)
```

Cadastra o código da fonte de dados

Parameters:

code - código da fonte de dados

getCorrelationTable

```
public java.util.Collection<SchemaCorrelation> getCorrelationTable()
```

Retorna a tabela de correlação entre o esquema da fonte de dados e o esquema conceitual.

Returns:

Collection<SchemaCorrelation> tabela de correlação

See Also:

[Collection](#), SchemaCorrelation

setCorrelationTable

```
public void setCorrelationTable(java.util.Collection<SchemaCorrelation> correlationTable)
```

Cadastra a tabela de correlação entre o esquema da fonte de dados e o esquema conceitual.

Parameters:

correlationTable - tabela de correlação

See Also:

[Collection](#), SchemaCorrelation

getDomain

```
public Domain getDomain()
```

Retorna o domínio de interesse associado à fonte de dados

Returns:

[Domain](#) domínio de interesse

See Also:

Domain

setDomain

```
public void setDomain(Domain domain)
```

Associa um domínio de interesse à fonte de dados.

Parameters:

domain - domínio de interesse

See Also:

Domain

getMetadata

```
public Metadata getMetadata()
```

Retorna os metadados geográficos associados à fonte de dados

Returns:

[Metadata](#) metadados geográficos

See Also:

Metadata

setMetadata

```
public void setMetadata(Metadata metadata)
```

Cadastra os metadados geográficos da fonte de dados

Parameters:

metadata - metadados geográficos

See Also:

Metadata

getName

```
public java.lang.String getName()
```

Retorna o nome da fonte de dados

Returns:

String nome da fonte de dados

setName

```
public void setName(java.lang.String name)
```

Cadastra o nome da fonte de dados

Parameters:

name - nome da fonte de dados

equals

```
public boolean equals(java.lang.Object obj)
```

Overrides:

equals in class java.lang.Object

hashCode

```
public int hashCode()
```

Overrides:

hashCode in class java.lang.Object

C.8 Classe [Domain](#)

br.uerj.eng.geomatica.interoperability.model

Class Domain

```
java.lang.Object
```

```
br.uerj.eng.geomatica.interoperability.model.Domain
```

```
public class Domain extends java.lang.Object
```

Representa um domínio de interesse.

Since:

1.0

Author:

Victor Azevedo (reference Azevedo, V. H. M.)

Constructor Summary	
Domain ()	Cria uma nova instancia de um domínio
Domain (int code)	Cria uma nova instancia de um domínio
Domain (int code, java.lang.String name)	Cria uma nova instancia de um domínio

Method Summary	
boolean	equals (java.lang.Object obj)
int	getCode () Retorna o código do domínio
java.lang.String	getDescription () Retorna a descrição do domínio
java.lang.String	getName () Retorna o nome do domínio
java.lang.String	getOntologyFile () Retorna o caminho completo do arquivo da ontologia owl que armazena a base de conhecimento sobre o domínio
int	hashCode ()
void	setCode (int code) Cadastra o código do domínio
void	setDescription (java.lang.String description) Cadastra a descrição do domínio
void	setName (java.lang.String name) Cadastra o nome do domínio
void	setOntologyFile (java.lang.String ontologyFile) Cadastra o arquivo de ontologia owl que representa a base de conhecimento sobre o domínio.

Methods inherited from class java.lang.Object

`getClass, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

Domain

```
public Domain()
```

Cria uma nova instancia de um domínio

Domain

```
public Domain(intcode, java.lang.String name)
```

Cria uma nova instancia de um domínio

Parameters:

code - código do domínio

name - nome do domínio

Domain

```
public Domain(intcode)
```

Cria uma nova instancia de um domínio

Parameters:

code - código do domínio

Method Detail

getCode

```
public int getCode()
```

Retorna o código do domínio

Returns:

int código do domínio

setCode

```
public void setCode(intcode)
```

Cadastra o código do domínio

Parameters:

code - código do domínio

getDescription

```
public java.lang.String getDescription()
```

Retorna a descrição do domínio

Returns:

String descrição do domínio

setDescription

```
public void setDescription(java.lang.String description)
```

Cadastra a descrição do domínio

Parameters:

description - descrição do domínio

getName

```
public java.lang.String getName()
```

Retorna o nome do domínio

Returns:

String nome do domínio

setName

```
public void setName(java.lang.String name)
```

Cadastra o nome do domínio

Parameters:

name - nome do domínio

getOntologyFile

```
public java.lang.String getOntologyFile()
```

Retorna o caminho completo do arquivo da ontologia owl que armazena a base de conhecimento sobre o domínio

Returns:

String caminho completo do arquivo owl

setOntologyFile

```
public void setOntologyFile(java.lang.String ontologyFile)
```

Cadastra o arquivo de ontologia owl que representa a base de conhecimento sobre o domínio.

Parameters:

ontologyFile - caminho completo do arquivo owl

equals

```
public boolean equals(java.lang.Object obj)
```

Overrides:

equals in class java.lang.Object

hashCode
public int hashCode()

Overrides:

hashCode in class java.lang.Object

C.9 Classe [GeographicInformationLayer](#)

br.uerj.eng.geomatica.interoperability.model
Class GeographicInformationLayer

```
java.lang.Object  
    br.uerj.eng.geomatica.interoperability.model.GeographicInformationLayer  
  
public class GeographicInformationLaye extends java.lang.Object
```

Representa uma camada de informação geográfica baseada em um domínio de interesse e que armazena informações de um conjunto de fontes de dados associadas a este domínio

Since:

1.0

Author:

Victor Azevedo (reference Azevedo, V. H. M.)

See Also:

Domain, DataSource

Constructor Summary	
GeographicInformationLayer ()	
Cria uma nova instância para GeographicInformationLayer	

Method Summary	
java.util.Collection	getDataSources () Retorna o conjunto de fontes de dados que compõem a camada de informação
Domain	getDomain () Retorna o domínio de interesse associado a camada de informação geográfica
java.lang.String	getFileName () Retorna o caminho completo do arquivo que contém as feições geográficas da camada de informação geográfica.

java.lang.String	getName() Retorna o nome da camada de informação geográfica
void	setDataSources() (java.util.Collection dataSources) Cadastra um conjunto de fontes de dados na camada de informação geográfica
void	setDomain() (Domain domain) Associa a camada de informação geográfica domínio de interesse a um domínio.
void	setFileName() (java.lang.String fileName) Armazena o caminho completo do arquivo que contém as feições geográficas da camada de informação geográfica.
void	setName() (java.lang.String name) Fornece um nome à camada de informação geográfica

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Detail

GeographicInformationLayer

`public GeographicInformationLayer()`

Cria uma nova instância para `GeographicInformationLayer`

Method Detail

getDomain

`public Domain getDomain()`

Retorna o domínio de interesse associado a camada de informação geográfica

Returns:

[Domain](#) domínio de interesse

See Also:

`Domain`

setDomain

`public void setDomain(Domain domain)`

Associa a camada de informação geográfica domínio de interesse a um domínio.

Parameters:

domain - domínio de interesse

See Also:

Domain

getFileName

```
public java.lang.String getFileName()
```

Retorna o caminho completo do arquivo que contém as feições geográficas da camada de informação geográfica. Este arquivo pode ser um shapefile ou um arquivo owl

Returns:

String caminho completo do arquivo

setFileName

```
public void setFileName(java.lang.String fileName)
```

Armazena o caminho completo do arquivo que contém as feições geográficas da camada de informação geográfica. Formatos shapefile ou owl

Parameters:

fileName - caminho completo do arquivo

getName

```
public java.lang.String getName()
```

Retorna o nome da camada de informação geográfica

Returns:

String nome da camada

setName

```
public void setName(java.lang.String name)
```

Fornece um nome à camada de informação geográfica

Parameters:

name - nome da camada

getDataSources

```
public java.util.Collection getDataSources()
```

Retorna o conjunto de fontes de dados que compõem a camada de informação

Returns:

[Collection](#) fontes de dados geográficas

See Also:

[Collection](#), DataSource

setDataSources

```
public void setDataSources(java.util.Collection dataSources)
```

Cadastra um conjunto de fontes de dados na camada de informação geográfica

Parameters:

dataSources - conjunto de fontes de dados geográficas

See Also:

[Collection](#), DataSource

C.10 Classe [Metadata](#)

br.uerj.eng.geomatica.interoperability.model

Class Metadata

java.lang.Object

br.uerj.eng.geomatica.interoperability.model.Metadata

```
public class Metadata extends java.lang.Object
```

Conjunto de metadados geográficos que identificam cada fonte de dados. Segue o padrão da especificação [ISO 19115 \(Geographic Metadata\)](#)

Since:

1.0

Author:

Victor Azevedo (reference Azevedo, V. H. M.)

Constructor Summary

[Metadata](#)()

Cria uma nova instância dos metadados geográficos

Method Summary

java.util.Date	getCreationDate () Retorna a data de criação da fonte de dados geográfica. Representa o item CI_DateTypeCode.creation da ISO 19115:2003
java.lang.String	getEmailAddress () Retorna o endereço de correio eletrônico do responsável pela fonte de dados.Representa o item CI_Address.electronicMailAddress da ISO 19115:2003
java.lang.String	getOrganization () Retorna o nome da intuição responsável pelos dados.Representa o item CI_ResponsibleParty.organisationName da ISO 19115:2003

java.util.Date	getPublicationDate () Retorna a data de publicação dos dados no servidor de integração. Representa o item CI_DateTypeCode.publication da ISO 19115:2003
java.lang.String	getPublicationScale () Retorna a escala de publicação dos dados. Representa o item MD_Resolution.equivalentScale da ISO 19115:2003
java.lang.String	getResponsible () Retorna o nome da pessoa responsável pelos dados.Representa o item CI_ResponsibleParty.individualName da ISO 19115:2003
void	setCreationDate (java.util.Date creationDate) Cadastra a data de criação da fonte de dados geográfica.Representa o item CI_DateTypeCode.creation da ISO 19115:2003
void	setEmailAddress (java.lang.String email) Cadastra o endereço de correio eletrônico do responsável pela fonte de dados geográfica.Representa o item CI_Address.electronicMailAddress da ISO 19115:2003
void	setOrganization (java.lang.String organization) Cadastra o nome da instituição responsável pelos dados.Representa o item CI_ResponsibleParty.organisationName da ISO 19115:2003
void	setPublicationDate (java.util.Date publicationDate) Cadastra data de publicação dos dados no servidor de integração. Representa o item CI_DateTypeCode.publication da ISO 19115:2003
void	setPublicationScale (java.lang.String publicationScale) Cadastra escala de publicação dos dados. Representa o item MD_Resolution.equivalentScale da ISO 19115:2003
void	setResponsible (java.lang.String responsible) Cadastra o nome da pessoa responsável pelos dados.Representa o item CI_ResponsibleParty.individualName da ISO 19115:2003

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

Metadata

```
public Metadata()
```

Cria uma nova instância dos metadados geográficos

Method Detail

getCreationDate

```
public java.util.Date getCreationDate()
```

Retorna a data de criação da fonte de dados geográfica. Representa o item CI_DateTypeCode.creation da ISO 19115:2003

Returns:

Date data de criação da fonte de dados

See Also:

Date, DataSource

setCreationDate

```
public void setCreationDate(java.util.Date creationDate)
```

Cadastra a data de criação da fonte de dados geográfica. Representa o item CI_DateTypeCode.creation da ISO 19115:2003

Parameters:

creationDate - data de criação da fonte de dados

See Also:

Date, DataSource

getEmailAddress

```
public java.lang.String getEmailAddress()
```

Retorna o endereço de correio eletrônico do responsável pela fonte de dados. Representa o item CI_Address.electronicMailAddress da ISO 19115:2003

Returns:

String endereço de correio eletrônico do responsável

See Also:

DataSource

setEmailAddress

```
public void setEmailAddress(java.lang.String email)
```

Cadastra o endereço de correio eletrônico do responsável pela fonte de dados geográfica. Representa o item CI_Address.electronicMailAddress da ISO 19115:2003

Parameters:

email - endereço de correio eletrônico do responsável

See Also:

DataSource

getOrganization

```
public java.lang.String getOrganization()
```

Retorna o nome da instituição responsável pelos dados. Representa o item CI_ResponsibleParty.organisationName da ISO 19115:2003

Returns:

String nome da instituição responsável

See Also:

DataSource

setOrganization

```
public void setOrganization(java.lang.String organization)
```

Cadastra o nome da instituição responsável pelos dados. Representa o item CI_ResponsibleParty.organisationName da ISO 19115:2003

Parameters:

organization - nome da instituição responsável

See Also:

DataSource

getPublicationDate

```
public java.util.Date getPublicationDate()
```

Retorna a data de publicação dos dados no servidor de integração. Representa o item CI_DateTypeCode.publication da ISO 19115:2003

Returns:

Date data de publicação dos dados

See Also:

Date, DataSource

setPublicationDate

```
public void setPublicationDate(java.util.Date publicationDate)
```

Cadastra data de publicação dos dados no servidor de integração. Representa o item CI_DateTypeCode.publication da ISO 19115:2003

Parameters:

publicationDate - data de publicação dos dados

See Also:

Date, DataSource

getPublicationScale

```
public java.lang.String getPublicationScale()
```

Retorna a escala de publicação dos dados. Representa o item MD_Resolution.equivalentScale da ISO 19115:2003

Returns:

String escala de publicação dos dados

See Also:

DataSource

setPublicationScale

```
public void setPublicationScale(java.lang.String publicationScale)
```

Cadastra escala de publicação dos dados. Representa o item MD_Resolution.equivalentScale da ISO 19115:2003

Parameters:

publicationScale - escala de publicação dos dados

See Also:

DataSource

getResponsible

```
public java.lang.String getResponsible()
```

Retorna o nome da pessoa responsável pelos dados. Representa o item CI_ResponsibleParty.individualName da ISO 19115:2003

Returns:

String nome da pessoa responsável

See Also:

DataSource

setResponsible

```
public void setResponsible(java.lang.String responsible)
```

Cadastra o nome da pessoa responsável pelos dados. Representa o item CI_ResponsibleParty.individualName da ISO 19115:2003

Parameters:

responsible - nome da pessoa responsável

See Also:

DataSource

C.11 Classe [SchemaCorrelation](#)

br.uerj.eng.geomatica.interoperability.model

Class SchemaCorrelation

```
java.lang.Object
```

```
br.uerj.eng.geomatica.interoperability.model.SchemaCorrelation
```

```
public class SchemaCorrelation extends java.lang.Object
```

Representa a correlação entre um termo do esquema da fonte de dados e uma propriedade da classe que representa a feição geográfica na ontologia (esquema de referência).

Since:

1.0

Author:

Victor Azevedo (reference Azevedo, V. H. M.)

Constructor Summary	
SchemaCorrelation ()	Cria uma nova instância de SchemaCorrelation

Method Summary	
SchemaElement	getDomainProperty () Retorna a propriedade da classe que representa a feição geográfica na ontologia.
java.util.Map <java.lang.String, java.lang.Object>	getQualitativeDataCorrelations () Retorna uma tabela contendo a correlação entre os dados qualitativos da fonte de dados e os indivíduos da ontologia no domínio da propriedade cadastrada em <code>this#setDomainProperty(SchemaElement)</code>
SchemaElement	getSchemaElement () Retorna o elemento do schema da fonte de dados
Void	setDomainProperty (SchemaElement domainProperty) Cadastra a propriedade da classe que representa a feição geográfica na ontologia.
void	setQualitativeDataCorrelations (java.util.Map <java.lang.String,java.lang.Object> qualifiedC orrelations) Cadastra uma tabela contendo a correlação entre os dados qualitativos da fonte de dados e os indivíduos da ontologia no domínio da propriedade cadastrada em <code>this#setDomainProperty(SchemaElement)</code>
Void	setSchemaElement (SchemaElement schemaElement) Cadastra o elemento do schema da fonte de dados

Methods inherited from class java.lang.Object
<code>equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait</code>

Constructor Detail
SchemaCorrelation public SchemaCorrelation()

Cria uma nova instância de SchemaCorrelation

Method Detail

getDomainProperty

public [SchemaElement](#) **getDomainProperty()**

Retorna a propriedade da classe que representa a feição geográfica na ontologia.

Returns:

[SchemaElement](#) propriedade da classe que representa a feição geográfica na ontologia

See Also:

[SchemaElement](#), [Domain](#), [OntologyService](#)

setDomainProperty

public void **setDomainProperty**([SchemaElement](#) domainProperty)

Cadastra a propriedade da classe que representa a feição geográfica na ontologia.

Parameters:

domainProperty - propriedade da classe que representa a feição geográfica na ontologia

See Also:

[SchemaElement](#), [Domain](#), [OntologyService](#)

getSchemaElement

public [SchemaElement](#) **getSchemaElement()**

Retorna o elemento do schema da fonte de dados

Returns:

[SchemaElement](#) elemento do schema da fonte de dados

See Also:

[SchemaElement](#), [DataSource](#), [DataSourceService](#), [WFSService](#)

setSchemaElement

public void **setSchemaElement**([SchemaElement](#) schemaElement)

Cadastra o elemento do schema da fonte de dados

Parameters:

schemaElement - elemento do schema da fonte de dados

See Also:

[SchemaElement](#), [DataSource](#), [DataSourceService](#), [WFSService](#)

getQualitativeDataCorrelations

public java.util.Map<java.lang.String, java.lang.Object>
getQualitativeDataCorrelations()

Retorna uma tabela contendo a correlação entre os dados qualitativos da fonte de dados e os indivíduos da ontologia no domínio da propriedade cadastrada em `setDomainProperty(SchemaElement)`

Returns:

Map tabela de correlação de dados qualitativos

See Also:

[Map](#), `DataSource`, `Domain`, `DataSourceService`, `WFSService`, `OntologyService`

setQualitativeDataCorrelations

```
public void
```

```
setQualitativeDataCorrelations(java.util.Map<java.lang.String, java.lang.Object>qualifiedCorrelations)
```

Cadastra uma tabela contendo a correlação entre os dados qualitativos da fonte de dados e os indivíduos da ontologia no domínio da propriedade cadastrada em `setDomainProperty(SchemaElement)`

Parameters:

Map - tabela de correlação de dados qualitativos

See Also:

[Map](#), `DataSource`, `Domain`, `DataSourceService`, `WFSService`, `OntologyService`

C.12 Classe [SchemaElement](#)

`br.uerj.eng.geomatica.interoperability.model`

Class `SchemaElement`

```
java.lang.Object
```

```
br.uerj.eng.geomatica.interoperability.model.SchemaElement
```

```
public class SchemaElement extends java.lang.Object
```

Representa um elemento de esquema conceitual.

Since:

1.0

Author:

Victor Azevedo (reference Azevedo, V. H. M.)

Constructor Summary	
SchemaElement ()	Cria uma nova instância de um elemento de esquema
SchemaElement (int code, java.lang.String name)	

Cria uma nova instância de um elemento de esquema

Method Summary

boolean	equals (java.lang.Object obj)
java.lang.String	getComment () Retorna um comentário sobre o elemento
java.lang.String	getName () Retorna o nome do elemento de esquema
java.lang.Class	getType () Retorna o tipo do elemento de esquema.
Int	hashCode ()
Void	setComment (java.lang.String comment) Cadastra um comentário sobre o elemento
Void	setName (java.lang.String name) Cadastra o nome do elemento de esquema
Void	setType (java.lang.Class type) Cadastra o tipo do elemento de esquema
java.lang.String	toString ()

Methods inherited from class java.lang.Object

`getClass, notify, notifyAll, wait, wait, wait`

Constructor Detail

SchemaElement

```
public SchemaElement()
```

Cria uma nova instância de um elemento de esquema

SchemaElement

```
public SchemaElement(intcode,  
                    java.lang.String name)
```

Cria uma nova instância de um elemento de esquema

Parameters:

code - código do elemento de esquema
name - nome do elemento de esquema

Method Detail

getName

```
public java.lang.String getName()
```

Retorna o nome do elemento de esquema

Returns:

String nome do elemento de esquema

setName

```
public void setName(java.lang.String name)
```

Cadastra o nome do elemento de esquema

Parameters:

name - nome do elemento de esquema

getType

```
public java.lang.Class getType()
```

Retorna o tipo do elemento de esquema.

Returns:

Class tipo do elemento de esquema (String.class, Integer.class, Multipolygon.class, etc.)

setType

```
public void setType(java.lang.Class type)
```

Cadastra o tipo do elemento de esquema

Parameters:

type - tipo do elemento de esquema (String.class, Integer.class, Multipolygon.class, etc.)

getComment

```
public java.lang.String getComment()
```

Retorna um comentário sobre o elemento

Returns:

String comentário sobre o elemento

setComment

```
public void setComment(java.lang.String comment)
```

Cadastra um comentário sobre o elemento

Parameters:

comment - comentário sobre o elemento

equals

public boolean **equals**(java.lang.Object obj)

Overrides:

equals in class java.lang.Object

hashCode

public int **hashCode**()

Overrides:

hashCode in class java.lang.Object

toString

public java.lang.String **toString**()

Overrides:

toString in class java.lang.Object

C.13 Classe [WFSDataSource](#)

br.uerj.eng.geomatica.interoperability.model

Class WFSDataSource

java.lang.Object

[br.uerj.eng.geomatica.interoperability.model.DataSource](#)

br.uerj.eng.geomatica.interoperability.model.WFSDataSource

public class **WFSDataSource** extends [DataSource](#)

Representa uma fonte de dados do tipo WFS

Since:

1.0

Author:

Victor Azevedo (reference Azevedo, V. H. M.)

Constructor Summary	
WFSDataSource ()	Cria uma nova instância de WFSDataSource
WFSDataSource (int code)	Cria uma nova instância de WFSDataSource

[WFSDataSource](#)(int code, java.lang.String name)
Cria uma nova instância de WFSDataSource

Method Summary

java.lang.String	getTypeName () Retorna o nome do FeatureType do serviço WFS que fornece os dados geográficos em GML
java.lang.String	getURLCapabilities () Retorna a URL da função GetCapabilities() do serviço WFS que fornece os dados geográficos em GML
Void	setTypeNames (java.lang.String typeName) Cadastra o nome do FeatureType do serviço WFS que fornece os dados geográficos em GML
Void	setURLCapabilities (java.lang.String capabilities) Cadastra a URL da função GetCapabilities() do serviço WFS que fornece os dados geográficos em GML

Methods inherited from class [br.uerj.eng.geomatica.interoperability.model.DataSource](#)

[equals](#), [getCode](#), [getCorrelationTable](#), [getDomain](#), [getMetadata](#), [getName](#), [hashCode](#), [setCode](#), [setCorrelationTable](#), [setDomain](#), [setMetadata](#), [setName](#)

Methods inherited from class [java.lang.Object](#)

[getClass](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Constructor Detail

WFSDataSource

```
public WFSDataSource()
```

Cria uma nova instância de WFSDataSource

WFSDataSource

```
public WFSDataSource(int code, java.lang.String name)
```

Cria uma nova instância de WFSDataSource

Parameters:

code - código da fonte de dados
name - nome da fonte de dados

WFSDataSource

```
public WFSDataSource(int code)
```

Cria uma nova instância de WFSDataSource

Parameters:

code - código da fonte de dados

Method Detail

getTypeName

```
public java.lang.String getTypeName()
```

Retorna o nome do FeatureType do serviço WFS que fornece os dados geográficos em GML

Returns:

String nome do WFS FeatureType

setName

```
public void setName(java.lang.String typeName)
```

Cadastra o nome do FeatureType do serviço WFS que fornece os dados geográficos em GML

Parameters:

typeName - nome do WFS FeatureType

getURLCapabilities

```
public java.lang.String getURLCapabilities()
```

Retorna a URL da função GetCapabilities() do serviço WFS que fornece os dados geográficos em GML

Returns:

String URL da função GetCapabilities()

setURLCapabilities

```
public void setURLCapabilities(java.lang.String capabilities)
```

Cadastra a URL da função GetCapabilities() do serviço WFS que fornece os dados geográficos em GML

Parameters:

capabilities - URL da função GetCapabilities()

C.14 Pacote [Service](#)

Package br.uerj.eng.geomatica.interoperability.service

Este pacote contém os serviços disponíveis na biblioteca para interagir com as fontes de dados, os domínios, as ontologias, os serviços WFS, etc.

See:

[Description](#)

Interface Summary	
DataSourceService	Interface que define as operações de uma fonte de dados geográfica.

Class Summary	
ConfigurationService	Serviço de configuração da biblioteca.
ExportService	Exporta as informações de uma camada de informação geográfica para um arquivo (shp,owl,etc.).
OntologyService	Este Serviço interage com as ontologias que representam os domínio de interesse em formato owl.
WFSService	Serviço que interage com as fontes de dados geográfica, disponíveis através de um serviço Web Feature Service (WFS) do consórcio Open Geospatial(OGC).

Package `br.uerj.eng.geomatica.interoperability.service` Description

Este pacote contém os serviços disponíveis na biblioteca para interagir com as fontes de dados, os domínios, as ontologias, os serviços WFS, etc. É responsável por realizar a integração dos dados.

Since:

1.0

Author:

Victor Azevedo (reference Azevedo, V. H. M.)

C.15 Interface [DataSourceService](#)

`br.uerj.eng.geomatica.interoperability.service`

Interface `DataSourceService`

All Known Implementing Classes:

[WFSService](#)

```
public interface DataSourceService
```

Interface que define as operações de uma fonte de dados geográfica.

Since:

1.0

Author:

Victor Azevedo (reference Azevedo, V. H. M.)

Field Summary	
Static java.lang.String	CACHE_SCHEMA Arquivo que armazena o cache dos esquemas (XMLSchema) das fontes de dados.

Method Summary	
java.util.Collection< SchemaElement >	getSchema (DataSource ds) Recupera o esquema da fonte de dados geográfica.
boolean	isGeometry (SchemaElement schema) Verifica se o elemento do esquema representa uma feição geográfica (Geometria).

Field Detail

CACHE_SCHEMA

static final java.lang.String **CACHE_SCHEMA**

Arquivo que armazena o cache dos esquemas (XMLSchema) das fontes de dados.

See Also:

[Constant Field Values](#)

Method Detail

getSchema

java.util.Collection<[SchemaElement](#)> **getSchema**([DataSource](#) ds)
throws [DataSourceException](#)

Recupera o esquema da fonte de dados geográfica.

Parameters:

ds - fonte de dados geográfica.

Returns:

Collection esquema da fonte de dados geográfica

Throws:

[DataSourceException](#)

isGeometry

boolean **isGeometry**([SchemaElement](#) schema)
throws [DataSourceException](#)

Verifica se o elemento do esquema representa uma feição geográfica (Geometria).

Parameters:

schema - esquema da fonte de dados

Returns:

true se o elemento do esquema for uma geometria

Throws:

[DataSourceException](#)

C.16 Classe [ConfigurationService](#)

br.uerj.eng.geomatica.interoperability.service

Class ConfigurationService

java.lang.Object

br.uerj.eng.geomatica.interoperability.service.ConfigurationService

All Implemented Interfaces:

java.beans.ExceptionListener

```
public class ConfigurationService extends java.lang.Object
```

Serviço de configuração da biblioteca. Armazena as informações dos domínios de interesse, fontes de dados geográficas e camadas de informação espacial.

Since:

1.0

Author:

Victor Azevedo (reference Azevedo, V. H. M.)

Field Summary	
Static java.lang.String	DATASOURCES_FILE_NAME Nome do arquivo xml das fontes de dados.
Static java.lang.String	DOMAINS_FILE_NAME Nome do arquivo xml de domínios.
Static java.lang.String	LAYERS_FILE_NAME Nome do arquivo xml das camadas geográficas .

Constructor Summary	
ConfigurationService ()	

Method Summary	
Void	exceptionThrown (java.lang.Exception ex)
DataSource	getDataSource (int code) Recupera uma fonte de dados pelo seu código de identificação.
java.util.Collection < DataSource >	getDataSources (Domain domain) Recupera todas as fontes de dados de um domínio específico.
Domain	getDomain (int code) Recupera um domínio pelo seu código de identificação.
java.util.Collection < DataSource >	restoreDataSources () Recupera as fontes de dados armazenadas.
java.util.Collection < Domain >	restoreDomains () Recupera os domínios de interesse armazenados.
java.util.Collection < GeographicInformationLayer >	restoreGeographicLayers () Recupera as camadas de informações geográficas.
Void	save () Salva o status atual da configuração, armazenando todos os elementos do modelo.
Void	setDataSources (java.util.Collection < DataSource > col) Atualiza as informações sobre as fontes de dados geográficas na configuração.
Void	setDomains (java.util.Collection < Domain > col) Atualiza as informações sobre as domínios de interesse na configuração.
Void	startup () Inicializa os objetos de configuração restaurando todos os elementos armazenados.
Void	storeDataSources (java.util.Collection col) Armazena uma coleção de fontes de dados para a integração.

Void	storeDomains (java.util.Collection< Domain > col) Armazena uma coleção de domínios de interesse representados em linguagem owl, checando a integridade de cada domínio.
Void	storeGeographicLayer (java.util.Collection< GeographicInformationLayer > col) Armazena uma coleção de camadas de informações geográficas, no âmbito da integração entre fontes de dados heterogêneas, utilizando uma ontologia como esquema de referência.

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Field Detail

DOMAINS_FILE_NAME

```
public static final java.lang.String DOMAINS_FILE_NAME
```

Nome do arquivo xml de domínios.

See Also:

[Constant Field Values](#)

DATASOURCES_FILE_NAME

```
public static final java.lang.String DATASOURCES_FILE_NAME
```

Nome do arquivo xml das fontes de dados.

See Also:

[Constant Field Values](#)

LAYERS_FILE_NAME

```
public static final java.lang.String LAYERS_FILE_NAME
```

Nome do arquivo xml das camadas geográficas .

See Also:

[Constant Field Values](#)

Constructor Detail

ConfigurationService

```
public ConfigurationService()
```

Method Detail

restoreDataSources

```
public java.util.Collection<DataSource> restoreDataSources()
```

Recupera as fontes de dados armazenadas.

Returns:

Collection coleção de fontes de dados

See Also:

[Collection](#), [DataSource](#)

restoreDomains

```
public java.util.Collection<Domain> restoreDomains()
```

Recupera os domínios de interesse armazenados.

Returns:

Collection coleção de domínios de interesse

See Also:

[Collection](#), [Domain](#)

save

```
public void save()
```

Salva o status atual da configuração, armazenando todos os elementos do modelo.

See Also:

[Domain](#), [DataSource](#), [GeographicInformationLayer](#)

startup

```
public void startup()
```

Inicializa os objetos de configuração restaurando todos os elementos armazenados.

See Also:

[Domain](#), [DataSource](#), [GeographicInformationLayer](#)

storeDataSources

```
public void storeDataSources(java.util.Collection col)
```

Armazena uma coleção de fontes de dados para a integração.

Parameters:

col - fonte de dados geográfica;

See Also:

[Collection](#), [DataSource](#)

storeDomains

```
public void storeDomains(java.util.Collection<Domain> col)
```

Armazena uma coleção de domínios de interesse representados em linguagem owl, checando a integridade de cada domínio.

See Also:

[Collection](#), [Domain](#)

setDataSources

```
public void setDataSources(java.util.Collection<DataSource> col)
```

Atualiza as informações sobre as fontes de dados geográficas na configuração.

Parameters:

col - coleção de fontes de dados

See Also:

[Collection](#), [DataSource](#)

setDomains

```
public void setDomains(java.util.Collection<Domain> col)
```

Atualiza as informações sobre as domínios de interesse na configuração.

Parameters:

col - coleção de domínios de interesse

getDomain

```
public Domain getDomain(intcode)
```

Recupera um domínio pelo seu código de identificação.

Parameters:

code - código de identificação do domínio

Returns:

Domain domínio de interesse

getDataSource

```
public DataSource getDataSource(intcode)
```

Recupera uma fonte de dados pelo seu código de identificação.

Parameters:

code - código de identificação da fonte de dados

Returns:

DataSource fonte de dados geográfica

getDataSources

```
public java.util.Collection<DataSource> getDataSources(Domain domain)
```

Recupera todas as fontes de dados de um domínio específico.

Parameters:

domain - domínio de interesse

Returns:

Collection coleção de fontes de dados geográficas

restoreGeographicLayers

```
public java.util.Collection<GeographicInformationLayer>  
restoreGeographicLayers()
```

Recupera as camadas de informações geográficas.

Returns:

Collection coleção de camadas de informações geográficas

See Also:

[Collection](#), [GeographicInformationLayer](#)

storeGeographicLayer

```
public void  
storeGeographicLayer(java.util.Collection<GeographicInformationLayer> col)
```

Armazena uma coleção de camadas de informações geográficas, no âmbito da integração entre fontes de dados heterogêneas, utilizando uma ontologia como esquema de referência.

Parameters:

col - camadas de informações geográficas

See Also:

[Collection](#), [GeographicInformationLayer](#)

exceptionThrown

```
public void exceptionThrown(java.lang.Exception ex)
```

Specified by:

exceptionThrown in interface `java.beans.ExceptionListener`

C.17 Classe [ExportService](#)

`br.uerj.eng.geomatica.interoperability.service`

Class ExportService

```
java.lang.Object  
br.uerj.eng.geomatica.interoperability.service.ExportService
```

```
public class ExportService extends java.lang.Object
```

Exporta as informações de uma camada de informação geográfica para um arquivo (shp,owl,etc.).

Since:

1.0

Author:

Victor Azevedo (reference Azevedo, V. H. M.)

Constructor Summary

[ExportService](#)()

Method Summary

static void	exportToOWL (GeographicInformationLayer layer) Exporta as informações de uma camada geográfica para uma representação em ontologias owl do domínio que a camada representa.
static void	exportToShapefile (GeographicInformationLayer map) Exporta as informações de uma camada geográfica para um arquivo shapefile.

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Detail

ExportService

```
public ExportService()
```

Method Detail

exportToOWL

```
public static void exportToOWL(GeographicInformationLayer layer)
```

Exporta as informações de uma camada geográfica para uma representação em ontologias owl do domínio que a camada representa.

Parameters:

layer - camada de informação geográfica

exportToShapefile

```
public static void exportToShapefile(GeographicInformationLayer map)
```

Exporta as informações de uma camada geográfica para um arquivo shapefile.

Parameters:

layer - camada de informação geográfica

C.18 Classe [OntologyService](#)

br.uerj.eng.geomatica.interoperability.service

Class **OntologyService**

java.lang.Object

br.uerj.eng.geomatica.interoperability.service.OntologyService

```
public class OntologyService extends java.lang.Object
```

Este Serviço interage com as ontologias que representam os domínio de interesse em formato owl.

Since:

1.0

Author:

Victor Azevedo (reference Azevedo, V. H. M.)

Field Summary	
static float	DEFAULT_MIN_SIMILARITY_METRIC Valor mínimo aceitável da métrica de similaridade entre dois termos.
static AbstractStringMetric	Jaro Algoritmo de similaridade métrica de Jaro da biblioteca SimMetrics
static AbstractStringMetric	JaroWinkler Algoritmo de similaridade métrica de Needleman-Wunch da biblioteca SimMetrics
static AbstractStringMetric	Levenshtein Algoritmo de similaridade métrica de Smith-Waterman da biblioteca SimMetrics
static AbstractStringMetric	MongeElkan Algoritmo de similaridade métrica de Monge-Elkan da biblioteca SimMetrics
static AbstractStringMetric	NeedlemanWunch Algoritmo de similaridade métrica de Needleman-Wunch da biblioteca SimMetrics
static AbstractStringMetric	QGramsDistance Algoritmo de similaridade métrica de Q-Gram da biblioteca SimMetrics
static AbstractStringMetric	SmithWaterman

	Algoritmo de similaridade métrica de Smith-Waterman da biblioteca SimMetrics
static AbstractStringMetric	SmithWatermanGotoh Algoritmo de similaridade métrica de Smith-Waterman-Gotoh da biblioteca SimMetrics

Constructor Summary	
OntologyService ()	

Method Summary	
static java.lang.String	getFeatureClassName (OWLModel owlModel) Recupera o nome da classe que representa a feição geográfica na ontologia do domínio.
static java.util.Set < SchemaElement >	getFeatureProperties (Domain domain) Recupera as propriedades não geométricas da feição geográfica do domínio.
static java.util.Set < SchemaElement >	getGeometryFeatureProperties (Domain domain) Recupera as propriedades geométricas da feição geográfica do domínio.
static java.util.Set < SchemaElement >	getOWLProperties (Domain domain) Recupera as propriedades da ontologia associada ao domínio de interesse.
static boolean	isSimilar (java.lang.String term1, java.lang.String term2) Compara a similaridade entre dois termos utilizando a métrica default.
static boolean	isSimilar (java.lang.String term1, java.lang.String term2, AbstractStringMetric stringMetric, java.lang.Float minSimilarityMetric) Compara a similaridade entre dois termos utilizando um algoritmo específico.
static boolean	isSimilar (java.lang.String term1, java.lang.String term2, java.lang.Float minSimilarityMetric) Compara a similaridade entre dois termos.

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Field Detail

DEFAULT_MIN_SIMILARITY_METRIC

```
public static float DEFAULT_MIN_SIMILARITY_METRIC
```

Valor mínimo aceitável da métrica de similaridade entre dois termos.

Levenshtein

```
public static AbstractStringMetric Levenshtein
```

Algoritmo de similaridade métrica de [Smith-Waterman](#) da biblioteca [SimMetrics](#)

NeedlemanWunch

```
public static AbstractStringMetric NeedlemanWunch
```

Algoritmo de similaridade métrica de [Needleman-Wunch](#) da biblioteca [SimMetrics](#)

SmithWaterman

```
public static AbstractStringMetric SmithWaterman
```

Algoritmo de similaridade métrica de [Smith-Waterman](#) da biblioteca [SimMetrics](#)

SmithWatermanGotoh

```
public static AbstractStringMetric SmithWatermanGotoh
```

Algoritmo de similaridade métrica de [Smith-Waterman-Gotoh](#) da biblioteca [SimMetrics](#)

MongeElkan

```
public static AbstractStringMetric MongeElkan
```

Algoritmo de similaridade métrica de [Monge-Elkan](#) da biblioteca [SimMetrics](#)

Jaro

```
public static AbstractStringMetric Jaro
```

Algoritmo de similaridade métrica de [Jaro](#) da biblioteca [SimMetrics](#)

JaroWinkler

```
public static AbstractStringMetric JaroWinkler
```

Algoritmo de similaridade métrica de [Needleman-Wunch](#) da biblioteca [SimMetrics](#)

QGramsDistance

```
public static AbstractStringMetric QGramsDistance
```

Algoritmo de similaridade métrica de [Q-Gram](#) da biblioteca [SimMetrics](#)

Constructor Detail

OntologyService

```
public OntologyService()
```

Method Detail

getOWLProperties

```
public static java.util.Set<SchemaElement> getOWLProperties(Domain domain)
```

Recupera as propriedades da ontologia associada ao domínio de interesse.

Parameters:

domain - domínio de interesse

Returns:

Set conjunto de propriedades da ontologias que representa o domínio

getFeatureClassName

```
public static java.lang.String  
getFeatureClassName(edu.stanford.smi.protege.owl.model.OWLModel owlModel)
```

Recupera o nome da classe que representa a feição geográfica na ontologia do domínio.

Parameters:

owlModel - modelo OWL da ontologia

Returns:

String nome da classe

getGeometryFeatureProperties

```
public static java.util.Set<SchemaElement>  
getGeometryFeatureProperties(Domain domain)
```

Recupera as propriedades geométricas da feição geográfica do domínio. A feição geográfica é representada na ontologia como uma subclasse de gml:FeaturePropertyType.

Parameters:

domain - domínio de interesse

Returns:

Set propriedades geométricas da feição geográfica do domínio

getFeatureProperties

```
public static java.util.Set<SchemaElement> getFeatureProperties(Domain  
domain)
```

Recupera as propriedades não geométricas da feição geográfica do domínio. A feição geográfica é representada na ontologia como uma subclasse de gml:FeaturePropertyType.

Parameters:

domain - domínio de interesse

Returns:

Set propriedades não geométricas da feição geográfica do domínio

isSimilar

```
public static boolean isSimilar(java.lang.String term1,  
                                java.lang.String term2)
```

Compara a similaridade entre dois termos utilizando a métrica default.
Utiliza o algoritmo de [Jaro-Winkler](#) da biblioteca [SimMetrics](#)

Parameters:

term1 - termo para comparação
term2 - termo para comparação

Returns:

boolean true se a comparação retornar uma métrica acima do valor mínimo definido.

isSimilar

```
public static boolean isSimilar(java.lang.String term1,  
                                java.lang.String term2,  
                                java.lang.Float minSimilarityMetric)
```

Compara a similaridade entre dois termos.
Utiliza o algoritmo de [Jaro-Winkler](#) da biblioteca [SimMetrics](#)

Parameters:

term1 - termo para comparação
term2 - termo para comparação
minSimilarityMetric - valor mínimo permitido para que os dois termos sejam considerados similares

Returns:

boolean true se a comparação retornar uma métrica acima do valor mínimo definido.

isSimilar

```
public static boolean isSimilar(java.lang.String term1,  
                                java.lang.String term2,  
                                AbstractStringMetric stringMetric,  
                                java.lang.Float minSimilarityMetric)
```

Compara a similaridade entre dois termos utilizando um algoritmo específico.
Utiliza a biblioteca [SimMetrics](#)

Parameters:

term1 - termo para comparação
term2 - termo para comparação
stringMetric - algoritmo de métrica de similaridade utilizado
minSimilarityMetric - valor mínimo permitido para que os dois termos sejam considerados similares

Returns:

boolean true se a comparação retornar uma métrica acima do valor mínimo definido.

C.19 Classe [WFSService](#)

br.uerj.eng.geomatica.interoperability.service

Class WFSService

java.lang.Object

br.uerj.eng.geomatica.interoperability.service.WFSService

All Implemented Interfaces:

[DataSourceService](#), java.beans.ExceptionListener

```
public class WFSService extends java.lang.Object implements  
DataSourceService
```

Serviço que interage com as fontes de dados geográfica, disponíveis através de um serviço Web Feature Service (WFS) do consórcio Open Geospatial(OGC).

Since:

1.0

Author:

Victor Azevedo (reference Azevedo, V. H. M.)

Field Summary

Fields inherited from interface

br.uerj.eng.geomatica.interoperability.service.DataSourceService

[CACHE_SCHEMA](#)

Constructor Summary

[WFSService](#)()

Method Summary

void	exceptionThrown (java.lang.Exception ex)
java.util.Collection < SchemaElement >	getSchema (DataSource ds) Recupera o esquema da fonte de dados, disponível no formato XMLSchema através da função describeFeatureType do serviço WFS.
boolean	isGeometry (SchemaElement schema) Verifica se o elemento do esquema representa uma feição geográfica do GML (GML Geometry).

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Detail

WFSService

```
public WFSService()
```

Method Detail

getSchema

```
public java.util.Collection<SchemaElement> getSchema(DataSource ds)  
throws DataSourceException
```

Recupera o esquema da fonte de dados, disponível no formato XMLSchema através da função `describeFeatureType` do serviço WFS.

Specified by:

[getSchema](#) in interface [DataSourceService](#)

Parameters:

`ds` - fonte de dados WFS

Returns:

Collection esquema da fonte de dados WFS

Throws:

[DataSourceException](#)

isGeometry

```
public boolean isGeometry(SchemaElement schema)
```

Verifica se o elemento do esquema representa uma feição geográfica do GML (GML Geometry).

Specified by:

[isGeometry](#) in interface [DataSourceService](#)

Parameters:

`schema` - esquema da fonte de dados

Returns:

`true` se o elemento do esquema for do tipo `gml:Geometry`

exceptionThrown

```
public void exceptionThrown(java.lang.Exception ex)
```

Specified by:

`exceptionThrown` in interface `java.beans.ExceptionListener`

C.20 Pacote [unitTest](#)

Package **br.uerj.eng.geomatica.interoperability.unitTest**

Contém os testes unitários (junit tests) que validam as classes da biblioteca.

See:

[Description](#)

Class Summary	
SimilarityStringTest	Testa a similaridade entre termos.

Package **br.uerj.eng.geomatica.interoperability.unitTest** Description

Contém os testes unitários (junit tests) que validam as classes da biblioteca.

Since:

1.0

Author:

Victor Azevedo (reference Azevedo, V. H. M.)

C.21 Classe [SimilarityStringTest](#)

br.uerj.eng.geomatica.interoperability.unitTest

Class **SimilarityStringTest**

```
java.lang.Object
    junit.framework.Assert
        junit.framework.TestCase
            br.uerj.eng.geomatica.interoperability.unitTest.SimilarityStringTest
```

All Implemented Interfaces:

junit.framework.Test

```
public class SimilarityStringTest extends junit.framework.TestCase
```

Testa a similaridade entre termos. Utiliza a biblioteca [SimMetrics](#)

Since:

1.0

Author:

Victor Azevedo (reference Azevedo, V. H. M.)

Constructor Summary	
SimilarityStringTest ()	

Constructor Detail

SimilarityStringTest

```
public SimilarityStringTest()
```

Method Detail

testLevenshteinMetric

```
public void testLevenshteinMetric()
```

Testa o algoritimo de [Levenshtein](#) da biblioteca [SimMetrics](#)

testNeedlemanWunchMetric

```
public void testNeedlemanWunchMetric()
```

Testa o algoritimo de [Needleman-Wunch](#) da biblioteca [SimMetrics](#)

testSmithWatermanMetric

```
public void testSmithWatermanMetric()
```

Testa o algoritimo de [Smith-Waterman](#) da biblioteca [SimMetrics](#)

testSmithWatermanGotohMetric

```
public void testSmithWatermanGotohMetric()
```

Testa o algoritimo de [Smith-Waterman-Gotoh](#) da biblioteca [SimMetrics](#)

testMongeElkanMetric

```
public void testMongeElkanMetric()
```

Testa o algoritimo de [Monge-Elkan](#) da biblioteca [SimMetrics](#)

testJaroMetric

```
public void testJaroMetric()
```

Testa o algoritimo de [Jaro](#) da biblioteca [SimMetrics](#)

testJaroWinklerMetric

```
public void testJaroWinklerMetric()
```

Testa o algoritimo de [Jaro-Winkler](#) da biblioteca [SimMetrics](#)

testQGramsDistanceMetric

```
public void testQGramsDistanceMetric()
```

Testa o algoritimo [Q-Gram](#) da biblioteca [SimMetrics](#)

Referências

- 1 FONSECA, F. T.; EGENHOFER, M. J. *Sistemas de Informação Geográficos Baseados em Ontologias*, Informática Pública, Belo Horizonte, v.1, n.2, dez. 1999. Disponível em: http://www.ip.pbh.gov.br/ANO1_N2_PDF/ip0102fonseca.pdf. Acesso em: 21 de junho de 2007.
- 2 OPEN GEOSPATIAL CONSORTIUM (OGC). Disponível em: <http://www.opengeospatial.org/>. Acesso em: 21 de junho de 2007.
- 3 LIMA, P.; CÂMARA, G.; QUEIROZ, G. *GeoBR: Intercâmbio Sintático e Semântico de Dados Espaciais*. In: BRAZILIAN SYMPOSIUM ON GEOINFORMATICS, IV, 2002, Caxambu (MG). Anais do Geoinfo 2002, Caxambu: 2002, p. 139-146. Disponível em: <http://www.geoinfo.info/geoinfo2002/papers/Lima.pdf>. Acesso em: 21 de junho de 2007.
- 4 CÂMARA, G et al. *Centros de Dados Geográficos*. In: _____. Anatomia de Sistemas de Informações Geográficas. São José dos Campos: INPE, 1996. INPE-11454-RPE/770. p. 167-173. Disponível em: <http://www.dpi.inpe.br/geopro/livros/anatomia.pdf>. Acesso em: 21 de junho de 2007.
- 5 CASANOVA, M. A. et al. *Integração e interoperabilidade entre fontes de dados geográficos*. In: CASANOVA, M. A. et al. Bancos de Dados Geográficos. Curitiba: Editora MundoGeo, 2005. p. 305- 340. Disponível em: <http://www.dpi.inpe.br/livros/bdados/>. Acesso em: 21 de junho de 2007.
- 6 WORLD WIDE WEB CONSORTIUM (W3C). *Extensive Markup Language (XML)*. Disponível em: <http://www.w3.org/XML/>. Acesso em: 21 de junho de 2007.
- 7 INTERNATIONAL ORGANIZATION FOR STANDARDIZATION/ TC 211 – GEOGRAPHIC INFORMATION/GEOMATICS. *Geographic information – Geography Markup Language (GML)*, 2003. ISO/TC 211/WG 4/PT 19136. Disponível em: <http://www.isotc211.org/opendoc/211n1392/211n1392.pdf>. Acesso em: 21 de junho de 2007.
- 8 DAVIS Jr., C. A. et al. *O Open Geospatial Consortium*. In: CASANOVA, M. A. et al. Bancos de Dados Geográficos. Curitiba: Editora MundoGeo, 2005. p. 367- 383. Disponível em: <http://www.dpi.inpe.br/livros/bdados/>. Acesso em: 21 de junho de 2007..
- 9 UNIVERSITY OF SHEFFIELD. *Simmetrics*. , Department Of Computer Science, Natural Language Processing Group. Disponível em: <http://www.dcs.shef.ac.uk/~sam/simmetrics.html>. Acesso em: 21 de junho de 2007.
- 10 WORLD WIDE WEB CONSORTIUM (W3C). *XML Schema*. Disponível em: <http://www.w3.org/XML/Schema> . Acesso em: 21 de junho de 2007.
- 11 WORLD WIDE WEB CONSORTIUM (W3C). *Web Services Activity*. Disponível em: <http://www.w3.org/2002/ws/> . Acesso em: 21 de junho de 2007.
- 12 WORLD WIDE WEB CONSORTIUM (W3C). *Semantic Web*. Disponível em: <http://www.w3.org/2001/sw/> . Acesso em: 21 de junho de 2007.

13 NOY, N. F.; McGUINNESS, D. L. *Ontology Development 101: A Guide to Creating Your First Ontology*. Stanford, CA: Stanford University, [2001], 94305. Disponível em: http://protege.stanford.edu/publications/ontology_development/ontology101.pdf . Acesso em: 22 de junho de 2007.

14 WORLD WIDE WEB CONSORTIUM (W3C). *OWL Web Ontology Language Reference*, 2004. Disponível em: <http://www.w3.org/TR/owl-ref/> . Acesso em: 22 de junho de 2007.

15 KLIEN, E.; LUTZ, M ; KUHN, W. *Ontology-Based Discovery of Geographic Information Services – An Application in Disaster Management*. In: Longley, P. A. Computers, Environment and Urban Systems, 30 ed. ISSN: 0198-9715 Elsevier, 2006. p. 102-123. Disponível em: http://ifgi.uni-muenster.de/~lutzm/Klien_et_al_Ontology-based_Discovery_of_GIServices.pdf. Acesso em: 22 de junho de 2007.

16 EMBRAPA SOLOS. Zoneamento Agroecológico para Culturas Oleaginosas na Região Amazônica, com ênfase na Fronteira Brasileira, FINEP, 2004.

17 PRINCETON UNIVERSITY. *WordNET: A lexical database for the English language*. Disponível em: <http://wordnet.princeton.edu/>. Acesso em: 22 de junho de 2007.

18 ALONSO, G. et al. *Web Services*. In: ____ Web Services: Concepts, Architectures and Applications. Springer Verlag, 2004. ISBN 3540440089. p. 123-149. Disponível em: <http://www.inf.ethz.ch/personal/alonso/WebServicesBook>. Acesso em: 22 de junho de 2007.

19 KRESSE, W. *Standardization of Geographic Information*. In: ISPRS Congress, XX, 2004, Istanbul, Turkey. p. 249. Disponível em: <http://www.isprs.org/istanbul2004/comm2/papers/132.pdf>. Acesso em: 22 de junho de 2007.

20 GRUBER, T. R. *Toward Principles for the Design of Ontologies Used for Knowledge Sharing*, International Journal Human-Computer, Studies 43, 2003, p. 907-928. Disponível em: <http://tomgruber.org/writing/onto-design.pdf>. Acesso em: 22 de junho de 2007.

21 HORRIDGE, M. et. al. *A Practical Guide To Building OWL Ontologies Using The Protégé-OWL Plugin and CO-ODE Tools*, ed. 1. The University Of Manchester, 2004. Disponível em: <http://www.co-ode.org/resources/tutorials/ProtegeOWLTutorial.pdf>. Acesso em: 22 de junho de 2007.

22 PROTÉGÉ. Disponível em: <http://protege.stanford.edu/>. Acesso em: 22 de junho de 2007.

23 ONTOLINGUA. Disponível em: <http://www.ksl.stanford.edu/software/ontolingua/>. Acesso em: 22 de junho de 2007.

24 OPEN GEOSPATIAL CONSORTIUM (OGC). *Compliant Products*. Disponível em: <http://www.opengeospatial.org/resource/products/compliant>. Acesso em: 22 de junho de 2007.

25 OPEN GEOSPATIAL CONSORTIUM (OGC). *Web Feature Service Implementation Specification*. Open Geospatial Consortium Inc, 2005. OGC 04-094. Versão: 1.1.0. 131 p. Disponível em: <http://www.opengeospatial.org/standards/wfs>. Acesso em: 22 de junho de 2007.

26 THE DARPA AGENT MARKUP LANGUAGE (DAML). *DAML Ontology Library*. Disponível em: <http://www.daml.org/ontologies/>. Acesso em: 22 de junho de 2007.

- 27 OPEN GEOSPATIAL CONSORTIUM (OGC). *Web Map Service Implementation Specification*. Open Geospatial Consortium Inc, 2004. OGC 04-024. Versão: 1.3. 85 p. Disponível em: <http://www.opengeospatial.org/standards/wms>. Acesso em: 22 de junho de 2007.
- 28 INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA (IBGE). *Documentação Técnica: Projeto de Sistematização das Informações sobre Recursos Naturais*. Disponível em: http://www.ibge.gov.br/servidor_arquivos_geo/. Acesso em: 22 de junho de 2007.
- 29 GEOSERVER. v. 1.5.1. Disponível em: <http://geoserver.org/>. Acesso em: 22 de junho de 2007.
- 30 RAMALHO FILHO, A.; BEEK, K. J. *Sistema de Avaliação da Aptidão Agrícola das Terras*. Rio de Janeiro, EMBRAPA, 1995. p.65.
- 31 FILETO, R. et. al. Uma arquitetura para sistemas de informações sobre solos voltada para o zoneamento agrícola. In: Congresso da Sociedade Brasileira de Informática Agropecuária (SBIAGRO), 2005 Londrina, PR. p. 8. Disponível em: http://www.inf.ufsc.br/~fileto/Agrissolos/publics/Agrissolos_SBIAgro2005.pdf. Acesso em: 23 de junho de 2007.
- 32 EMPRESA BRASILEIRA DE PESQUISA AGROPECUÁRIA (EMBRAPA). *Sistema Brasileiro de Classificação de Solos - SIBCS*. Disponível em: <http://www.cnps.embrapa.br/sibcs>. Acesso em: 23 de junho de 2007.
- 33 SANTOS, H. G. et al. *Procedimentos Normativos de Levantamentos Pedológicos*. Rio de Janeiro: EMBRAPA, Centro Nacional de Pesquisa de Solos, 1995.
- 34 OLIVEIRA, J. B., JACOMINE, P. K. T., CAMARGO, M. N. *Classes Gerais de Solos do Brasil: guia auxiliar para seu reconhecimento*. Jaboticabal: FUNEP, 1992.
- 35 RECTOR, A. et. al. *OWL Pizzas: Practical Experiences of Teaching OWL-DL: Common Errors and Common Patterns*. In: International Conference on Knowledge Engineering and Knowledge Management (EKAW), 14, 2004, Whittlebury Hall, Northamptonshire, UK. p. 19. Disponível em: <http://www.co-ode.org/resources/papers/ekaw2004.pdf>. Acesso em: 23 de junho de 2007.
- 36 GEOTOOLS. *The Open Source Java GIS Toolkit*. v. 2.3.1. Disponível em: <http://geotools.codehaus.org/>. Acesso em: 23 de junho de 2007.
- 37 PROTÉGÉ. *Protégé-OWL API*. v. 3.2.1. Disponível em: <http://protege.stanford.edu/plugins/owl/api/>. Acesso em: 23 de junho de 2007.