

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE  
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA  
COORDENAÇÃO DE PÓS-GRADUAÇÃO EM INFORMÁTICA

**SWS-GIS: Uma arquitetura baseada em serviços para uma federação  
de Spatial Data Infrastructures**

**Fábio Luiz Leite Júnior**

(Mestrando)

**Cláudio de Souza Baptista, PhD**

(Orientador)

Campina Grande – PB  
Maio de 2007

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE  
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA  
COORDENAÇÃO DE PÓS-GRADUAÇÃO EM INFORMÁTICA

**SWS-GIS: Uma arquitetura baseada em serviços para uma federação  
de Spatial Data Infrastructures**

Fábio Luiz Leite Júnior

Dissertação submetida à Coordenação do Curso de Pós-Graduação em ciência da computação da Universidade Federal de Campina Grande, como parte dos requisitos necessários para obtenção do grau de Mestre em Ciência da Computação.

Área de Concentração: Ciência da Computação

Linha de Pesquisa: Sistemas de Informação e Banco de Dados

Orientador: Cláudio de Souza Baptista, PhD

Campina Grande – PB  
Maio de 2007

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

L533s Leite Júnior, Fábio Luíz  
SWS-GIS : uma arquitetura baseada em serviços para uma federação de  
spatial data infrastructures / Fábio Luíz Leite Júnior . – Campina Grande,  
2007.

101f. : il. color.

Referências

Dissertação (Mestrado em Ciência da Computação) – Universidade  
Federal de Campina Grande, Centro de Engenharia Elétrica e Informática.

Orientador: Cláudio de Souza Baptista.

1. Banco de Dados 2. Geoprocessamento 3. Sistemas de Informações  
Geográficas 4. Arquitetura Orientada a Serviço I. Título.

CDU 681.3.07

**“SWS-GIS: UMA ARQUITETURA BASEADA EM SERVIÇOS PARA UMA  
FEDERAÇÃO DE SPATIAL DATA INFRASTRUCTURES”**

**FÁBIO LUIZ LEITE JUNIOR**

**DISSERTAÇÃO APROVADA EM 30.05.2007**

  
**PROF. CLÁUDIO DE SOUZA BAPTISTA, Ph.D**  
Orientador

  
**PROF. ULRICH SCHIEL, Dr.**  
Examinador

  
**PROF<sup>a</sup> VALÉRIA GONÇALVES SOARES, Dr<sup>a</sup>**  
Examinadora

**CAMPINA GRANDE – PB**

"O que as vitórias têm de mau é que não são definitivas. O que as derrotas têm de bom é que também não são definitivas."

(José Saramago)

## Agradecimentos

*Primeiramente à Deus, por permitir o aprendizado constante. À minha família que sempre esteve e estará comigo onde eu estiver, principalmente minha Mãe e meu irmão Tiago e Isabella, minha noiva.*

*A todos os meus amigos. Aos meus amigos de Toritama que sempre me ajudam a enxergar a vida de forma muito especial com toda a sua presteza. Agradeço aos meus amigos irmão que dividi a casa nos anos de graduação e mestrado. Inicialmente, Emerson, Carlos e Lázaro que me proporcionaram anos inesquecíveis onde aprendemos a dividir as diferenças e a nos divertir bastante. No segundo momento aos Vagabundos (Flávio, Lauro, Gustavo, Filipe e Daniel), pelas discussões construtivas a respeito de universo em geral (geladeira, contas, globo, empregadas, contas, feiras, cachaças e muitos outros episódios engraçados e sempre construtivos). Aos meus amigos irmãos que encontrei aqui em Campina Grande, Hugo, Aliandro, Danilo, Vinicius, André e tantos outros colegas de curso. Aos amigos fora da universidade que foram companheiros ao longo desta caminhada, destaco Sandro e Patrícia.*

*Ao meu orientador e amigo Cláudio Baptista.*

*Aos meus colegas do LSI por me acompanharem ao longo destes anos, a Elvis (uma parceria de mais de cinco anos), Fábio Gomes, Damião, ao inoxidável Patrício, Cecir, Rodrigo Vilar, Michael Ângelo, Flávio Olsen, Hugo Feitosa, Yuri, Cláudio Campelo.*

*Aos meus colegas de Projeto 1 e Projeto 2 Lucas, Ricardo e Luciana por me ajudarem na minha pesquisa com os seus excelentes trabalhos.*

*A Aninha, por estar sempre com um sorriso amigo para nos ajudar.*

*A todos os professores que sempre prestativos contribuíram para o meu desenvolvimento intelectual.*

*Finalmente, agradeço a todos os brasileiros e ao governo brasileiro por manterem instituições de pesquisa fabulosas como é o DSC/UFCG, que com grande esforço dos professores mantêm as pesquisas de ponta no Brasil.*

# Sumário

Sumário.....	VI
Lista de Abreviaturas.....	VIII
Lista de Figuras .....	IX
Lista de Tabelas .....	X
Resumo .....	XI
Abstract.....	XII
Capítulo 1 . Introdução .....	13
1.1 Objetivos.....	16
1.2 Estrutura da dissertação.....	17
Capítulo 2 . Fundamentação Teórica.....	18
2.1 Interoperabilidade em SIG Distribuído .....	18
2.2 Spatial Data Infrastructure e descoberta de recursos geoespaciais.....	19
2.2.1 Evolução da descoberta de recursos em SDI.....	22
2.3 Modelos de arquitetura para desenvolvimento de sistemas de informações .....	23
2.3.1 Modelos para composição de GIServices.....	26
2.4. Padrões propostos pelo OpenGeospatial Consortium .....	28
2.4.1 Geographic Markup Language .....	28
2.4.2 Padrões de serviços.....	29
2.4.3 Padrão eBRIM para Modelagem de catálogos de serviços OGC.....	32
2.5 Web semântica e ontologias .....	34
2.5.1 Reconciliação de ontologias .....	37
2.5 Considerações finais .....	39
Capítulo 3 . Trabalhos Relacionados .....	40
3.1 Descoberta de serviços em uma federação de registros (Sivashanmugam et.al)..	40
3.2 Estratégia de mediação baseada em WFS para interoperabilidade em SIG (Boucelma et. al).....	41
3.3 Busca baseada em ontologias para mapas digitais interativos (Hübner et. al) .....	43
3.4 Integrando descrições sintáticas e semânticas na formação de cadeias de serviços geográficos (Lemmens et. al) .....	44
3.5 Descoberta de recursos em uma infra-estrutura de spatial data infrastructure Europeu (Smits e Friis-Christensen) .....	45
3.6 Descoberta de serviços baseada em ontologia em spatial data infrastucture (Lutz) .....	46
3.7 Uma arquitetura para troca e gerenciamento de informações geotécnicas distribuídas (Zimmerman et al) .....	47
3.8 Uma técnica para descoberta orientada a serviços e recuperação de dados espaciais (Ghosh e Paul).....	48
3.9 Uma técnica peer-to-peer para descoberta de GIS services (Xiujun et. al).....	49
3.8 Considerações finais .....	50
Capítulo 4 . A arquitetura do SWS-GIS .....	52
4.1 Requisitos funcionais.....	52
4.2 Requisitos não-funcionais.....	54
4.3 Atores do sistema .....	55
4.4 Arquitetura do SWS-GIS.....	56
4.5 Serviços da arquitetura .....	58
4.5.1 Serviços de dados (WFS e WMS).....	59
4.5.2 Serviço de apresentação de mapas (Basemap service).....	59

4.5.2 Serviço de LBS ( <i>Location Based service</i> ).....	59
4.5.3 Serviço de gazetteer.....	60
4.5.4 Serviço de busca de rotas.....	60
4.5.5 Serviço de workflow.....	61
4.6 O serviço de catálogo .....	61
4.6.1 Modelagem dos recursos nos catálogos do SDI.....	62
4.6.2 Descrição dos módulos do catálogo .....	64
4.6.3 Módulo cliente de acesso ao catálogo .....	66
4.6.4 Módulo de consulta .....	68
4.6.5 Módulo de reescrita .....	69
4.6.6 Módulo de propagação .....	69
4.6.7 Módulo de casamento semântico geográfico (MCSG).....	69
4.7 Protocolo de recuperação da informação em catálogos distribuídos.....	70
4.7.1 Estabelecimento e manutenção da rede de catálogos .....	72
4.8 Considerações finais .....	74
Capítulo 5 . Estudo de caso .....	75
5.1 Descrição do cenário do estudo de caso .....	76
5.2 Organização das comunidades semânticas estabelecidas.....	76
5.3 Consultando por serviços espaciais .....	77
5.3 Processamento da consulta pelo catálogo.....	81
5.4 Considerações finais .....	87
Capítulo 6 . Conclusão .....	88
6.1 Principais contribuições.....	88
6.2 Trabalhos futuros.....	91
Referências Bibliográficas.....	93



## **Lista de Abreviaturas**

API (Application Programming Interface)  
BPEL (Business Process Execution Language)  
CDL (Contract Description Lanaguage)  
CSW (OGC Catalog Services for the Web)  
DAML (DARPA Agent Markup Language)  
DAML - S (DARPA Agent Markup Language Services)  
DHT (Distributed Hash Table)  
HTML (HyperText Markup Language)  
IBGE (Instituto Brasileiro de Geografia e Estatística)  
LBS (Location Based Service)  
OGC (Open Geospatial Consortium)  
OIL (Ontology Inference Layer)  
OWL (Ontology Web Language)  
OWL-S (Ontology Web Language Services)  
OWS (OGC Open Web Services specification)  
SDI (Spatial Data Infrastructure)  
SGBD (Sistema Gerenciador de Banco de Dados)  
SOA (Service Oriented Architecture)  
SOAP (Simple Object Access Protocol)  
SQL (Structured Query Language)  
SVG - Scalable Vector Graphics  
UDDI (Universal Description, Discovery and Integration)  
URL (Uniform Resource Locator)  
URI (Uniform Resource Identifier)  
UML (Unified Modeling Language)  
XML (Extensible Marup Language)  
WFS (Web Feature Service)  
WMS (Web Map Service)  
WSDL (Web Services Definition Language)  
WSMO (Web Service Modeling Ontology)

## Lista de Figuras

Figura 2.1: SDI e o papel dos Geoportais Fonte: [DA05] .....	23
Figura 2.2: web services: Atores, objetos e protocolos (Fonte: [IWS07]) .....	25
Figura 2.3: Visão de alto nível do modelo descrito no padrão ebRIM .....	33
Figura 2.4: Mediador nas consultas distribuídas (Fonte: OGC07) .....	33
Figura 2.5: Mostra uma representação de um conhecimento. [SBH06] .....	35
Figura 2.6: Mostra o exemplo do grafo acima codificado na linguagem RDF/XML	36
Figura 2.7 Ilustra mapeamentos entre as ontologias individuais [Fonte: HPS04] ...	38
Figura 2.8: Ilustra a técnica onde ontologias individuais possuem mapeamentos para a ontologia individual [HPS04] .....	38
Figura 2.9: Ilustrando os mapeamentos entre as ontologias locais de cada cluster e os mapeamentos entre as ontologias locais e individuais [HPS04]. .....	39
Figura 3.1: Mediador para várias bases WFS (Fonte: BEL02) .....	42
Figura 4.1: Diagrama de use cases .....	56
Figura 4.2: Projeto arquitetural .....	57
Figura 4.3: Arquitetura da federação de SDI.....	59
Figura 4.4: Organização dos mapeamentos semânticos entre as comunidades formadas pelos SDI .....	63
Figura 4.5: Arquitetura detalhada do serviço de catálogo. Na figura são apresentados seus módulos, a ferramenta de descrição, o relacionamento com a ontologia do SDI e a participação dos atores do sistema.....	65
Figura 4.6: Tela de entrada para o módulo cliente de consulta ao catálogo.....	67
Figura 4.7: Diagrama de seqüência do processo de consulta ao catálogo.....	72
Figura 5.1: Processo de consulta ao catálogo .....	78
Figura 5.2: Processo de consulta ao catálogo .....	79
Figura 5.3: Processo de consulta ao catálogo .....	80
Figura 5.4: Processo de consulta ao catálogo .....	81
Figura 5.5 a): Exemplo de parte da ontologia da AESA e do Estado .....	83
Figura 5.5 b): Exemplo de parte da ontologia utilizada na SUDEMA.....	84
Figura 5.6: Serviços retornados pelos catálogos distribuídos em outros SDI e a cadeia formada com o serviço de apresentação .....	86
Figura 5.7: Resultado final apresentado no mapa, com os rios do sertão, estradas e rodovias do estado e açudes.....	87

## **Lista de Tabelas**

Tabela 3.1: Trabalhos relacionados .....	51
Tabela 6.1 Diferenças entre as soluções apresentadas e o trabalho desenvolvido ....	91

## Resumo

O volume de dados espaciais disponíveis na Web tem crescido rapidamente. Atualmente, a Web tem sido usada não apenas para busca de documentos, mas também para a provisão e o uso de serviços, conhecidos como *Web services*, que são publicados em um diretório e podem ser automaticamente descobertos por agentes automáticos. Particularmente, no domínio espacial, a possibilidade de acessar estes conjuntos de dados espaciais via *web services* tem motivado pesquisas no campo de *Spatial Data Infrastructure* (SDI). Neste trabalho, apresentamos o SWS-GIS que é um SDI implementado utilizando a arquitetura orientada a serviços. O SWS-GIS permite buscas espaciais em catálogos distribuídos que formam uma federação de serviços espaciais. A federação é mantida através de uma hierarquia semântica definida em cada SDI e implementada nos seus respectivos catálogos.

## **Abstract**

The huge volume of spatial data available in the Web has increasing rapidly. Currently, the Web has been used not only for searching documents but also for service provision, known as Web services. These services are published in a directory, and may be automatically discovered. Particularly, in the spatial domain, the possible use of such services has demanded research on how to distribute and interoperate spatial data. Therefore, Spatial Data Infrastructures (SDI) have been proposed in order to tackle these problems. However the lack of semantics in such SDI is still an open issue. In this dissertation, we propose SWS-GIS, which is a semantic SDI using service-oriented architecture based on Web services. SWS-GIS enables distributed spatial queries over catalogues, which take part in a federation of both spatial data and services. The federation data model is according to a hierarchical ontology, which enables concept mappings.

## Capítulo 1 . Introdução

---

A utilização das informações geográficas tem desempenhado importante papel nas organizações públicas e privadas, principalmente na tomada de decisões estratégicas. Segundo Longley et al. [LGM+05], "*Geoprocessamento é o conjunto de tecnologias que utilizam representações computacionais do espaço geográfico para modelar e analisar fenômenos espaços-temporais*". Ainda em seu texto, Longley define que sistemas de informações geográficas (SIG) são sistemas computacionais com os seguintes objetivos:

- Inserir e integrar, numa base de dados, informações espaciais provenientes de dados cartográficos, dados censitários e cadastro urbano e rural, imagens de satélite, redes e modelos numéricos de terreno;
- Oferecer mecanismos para combinar as várias informações, através de algoritmos de manipulação e análise, bem como para consultar, recuperar, visualizar e plotar o conteúdo da base de dados georreferenciados.

Os Sistemas de informações geográficas (SIG) estão em constante evolução acompanhando as mudanças da tecnologia da informação. Com o advento da Internet, as ferramentas de SIG e, conseqüentemente, as informações espaciais têm ganhado grande poder de difusão. Entretanto, as informações espaciais ainda possuem custos elevados e são difíceis de serem coletadas, embora essas técnicas tenham avançado bastante. Além destes problemas, Rajabifard e Williamson [RW01] destacam em seu trabalho outros motivos da falta de informação espacial para que as organizações possam resolver seus problemas, são eles:

- Diversas organizações precisam de mais dados que elas podem obter;
- Organizações, freqüentemente, necessitam de dados de fora de sua jurisdição ou área operacional;
- Dados coletados por diferentes organizações são freqüentemente incompatíveis.

O principal desafio é a grande quantidade de informações espaciais espalhada em diversos órgãos e mantida de forma inadequada, gerando replicação (redundância de dados e esforço) e problemas de atualização. Por exemplo, no governo do estado da Paraíba vários setores (AES/SECTMA, SUDEMA, DER, PBGAS) possuem

informações espaciais de açudes, rios, estradas, malha de gás, cidades e outras informações. Porém, somente os órgãos que trabalham diretamente com as informações relacionadas atualizam os dados espaciais, enquanto os outros órgãos, que apenas utilizam as informações, fazem cópias e não as mantêm atualizadas.

Destarte, as informações de rios, açudes e hidrografia em geral são mantidas apenas na AESA/SECTMA, as informações da malha de gás estão sempre atualizadas no órgão PBGAS, informações de estradas são mantidas pelo DER-PB e assim por diante. É necessária uma estrutura que mantenha os dados nos respectivos setores, na qual as entidades que necessitam (consumidores) dos dados e serviços possam localizar e usá-los de forma atualizados. Dessa forma, apenas os órgãos responsáveis pela manutenção da informação provida terão permissão de alterar as informações.

Visando resolver tais problemas, várias ferramentas de descoberta, publicação e localização de dados espaciais surgiram. Uma das primeiras infra-estruturas são os Clearinghouses, que estabelecem sítios centralizados cujos metadados de várias fontes podem ser encontrados, provendo serviços de buscas, visualização e transferência de dados espaciais [Fed05, CBR+04]. Várias iniciativas foram formadas em todo mundo, como é o caso de vários órgãos estaduais americanos [Iow07, Ida07, Ill07, Wyo07], o órgão americano *Federal Geographic Data Committee* (FGDC) [Fed05] e vários países da Europa (Reino Unido [Ukg07]) e de outros continentes (National Spatial Information Framework Clearing House Node – África do Sul [NSI07]). Esse tipo de ferramenta tem como único objetivo localizar dados e, portanto, não trata localização de serviços, o que torna os Clearinghouses incapazes de atender às novas tendências das tecnologias orientadas a serviços.

Em seguida, surgiram os Geoportais, que consistem de portais web (sítio web) que organizam e catalogam recursos como serviços, informações, ferramentas de busca, aplicações, provendo um meio para consultar os metadados sobre dados e serviços e ligando diretamente aos provedores dos recursos [LGM+05]. Podemos citar como exemplo, o geoportal criado pelo FGDC americano que é o GOS [GOS07].

É neste contexto que muitas organizações e governos se reuniram, chegando a um acordo que pode beneficiar essas entidades com melhor gerenciamento dos dados e, mais tarde, os serviços geográficos, dando origem aos *Spatial Data Infrastructures* (SDI). Os SDIs são um conjunto de ferramentas, políticas, tecnologias, padrões e pessoas com o objetivo de compartilhar dados espaciais. Vários exemplos de SDI podem ser observados, entre eles, destacam-se o americano *National Spatial Data*

*Infrastructure* mantido pelo *Federal Geographic Data Committee* (FGDC) [Fed05], o INSPIRE [Ins05] que é uma iniciativa europeia, o ANZLIC [Anz05] formado por grupos australianos e neozelandeses, o GSDI [Gsd05] que é uma tentativa de criar uma infra-estrutura global, entre outros.

Concomitantemente, uma crescente demanda por serviços de geoprocessamento vem surgindo à medida que avançam as estruturas de redes mundiais, com essas tendências, uma nova arquitetura vem emergindo e sendo chamada de (*Geographic Information*) *Service Driven Infrastructure* (GI-SDI) [DA05, RW01]. A idéia principal é compartilhar funcionalidades encapsuladas através de serviços (serviços de geoprocessamento – GIServices [Son04]), permitindo que usuários combinem funcionalidades oferecidas por diferentes provedores. Essa nova arquitetura vem sendo definida pela *OpenGeospatial Consortium* através de suas especificações.

Acompanhando essa evolução, algumas tecnologias vêm surgindo e contribuindo para arquitetura orientada a serviços, como é o caso de *web services*. Os *web services* são "*aplicações de software identificadas por uma URI, cujas interfaces e associações devem estar bem definidas, descritas e passíveis de serem descobertas com artefatos em XML. Um web service provê interações diretas com outros agentes de software usando troca de mensagens baseadas em XML através dos protocolos da Internet*" [W3C05]. Estas tecnologias deixaram ainda mais "interoperáveis" os meios de publicação e localização de recursos, pois os protocolos de trocas de mensagens são padronizados. Contudo, ainda é possível identificar alguns problemas semânticos com a tecnologia de *web services*, como o fraco suporte a descrição semântica e classificação de serviços e insuficiência da especificação para um modelo de dados para os catálogos.

No contexto da interoperabilidade semântica em SDI, Kuhn [Kuh05] enumerou três categorias de problemas em pesquisas futuras: (1) suporte a descobrimento e avaliação de dados; (2) suporte à descoberta e avaliação de serviços; (3) suporte à composição de serviços. Quanto à primeira classe de problemas, as pesquisas se mostram bastante avançadas como é o caso de mapeamentos semânticos entre esquemas, melhor detalhado no capítulo 2. Quanto a segunda e terceira classes de problemas, as tecnologias recentes ainda se mostram ineficientes. É notável a necessidade de ferramentas de localização de recursos distribuídos, baseadas em arquitetura orientada a serviços que integrem automaticamente vários SDI.

O propósito deste trabalho é propor uma ferramenta orientada a serviços, na qual o projetista de serviço possa realizar consultas espaciais que são executadas em diversos



catálogos federados, aumentando o poder de difusão e integração dos SDI. Para tanto, os recursos dos catálogos são enriquecidos com descrições semânticas através das ontologias. Diferente dos trabalhos propostos, esta ferramenta utiliza um protocolo de recuperação da informação que se baseia na dimensão espacial das descrições dos serviços contidas nos catálogos dos SDIs e restringe o espaço de busca a medida em que os resultados vão sendo encontrados na rede.

## 1.1 Objetivos

O objetivo do trabalho é apresentar uma ferramenta baseada na arquitetura orientada a serviços, com a qual será possível realizar buscas por recursos geoespaciais em catálogos distribuídos em diferentes domínios compondo uma federação. A ferramenta deve ser composta por vários serviços de geoprocessamento e pode ser estendida de acordo com as necessidades dos usuários. Baseada na arquitetura proposta por Alameh [Alm03], a ferramenta possui os seguintes objetivos específicos:

- **capacidade de localização de serviços utilizando a dimensão espacial das descrições semânticas:** a ferramenta deve ser capaz de localizar os recursos que o usuário busca através de operações espaciais sobre os dados do catálogo, segue alguns exemplos de consultas que o usuário será capaz de realizar:
  - consultar por recursos que estão dentro de um determinado raio geográfico. Por exemplo, buscar no catálogo informações espaciais de todos os reservatórios que estejam num raio de 100 km do município de Campina Grande;
  - consultar por informações que relacionem objetos geográficos. Por exemplo, buscar informações sobre a malha de distribuição de gás no município de Campina Grande. Ou informações sobre a rede de água e esgoto que passa em determinado bairro para uma pavimentação de uma rua. Consulta por informações sobre a situação das estradas do estado da Paraíba;
  - consultar sobre determinadas informações a respeito de uma área determinada. Por exemplo, informações sobre dados ou serviços com informações meteorológicas da Paraíba.
- **capacidade de compartilhar recursos entre vários catálogos:** O usuário pode optar por realizar a busca em vários catálogos, ou seja, caso a cadeia de serviço presente no catálogo local não atenda as necessidades do usuário, o catálogo

deve propagar a consulta por serviços (e dados) que não estão presentes no catálogo local.

- **Resolver problemas relativos à geocodificação (nome geográfico ou código espacial):** ao passar um nome de um lugar no espaço, o sistema deve utilizar um gazetteer para resolver os nomes espaciais dos locais consultados e obter as coordenadas espaciais (*features*) a serem pesquisadas na base de descrições.

## 1.2 Estrutura da dissertação

A organização da dissertação se dá conforme o que segue:

No capítulo 2 contém uma fundamentação teórica que apresenta os diversos conceitos que embasam todo o trabalho. Neste capítulo serão apresentados conceitos de interoperabilidade, geoprocessamento, SIG, web services e ontologias.

No capítulo 3 é realizado um estudo no estado da arte e uma análise das soluções propostas nos vários trabalhos relacionados.

No capítulo 4 será apresentada a solução proposta, os casos de uso e uma discussão sobre a implementação do catálogo, do SDI e da federação dos SDIs.

No capítulo 5 será descrito o estudo de caso utilizando a arquitetura projetada. Será acompanhado passo à passo como se dá o processo desde a requisição do usuário até a execução da cadeia de serviços selecionada.

No capítulo 6 são apresentadas as conclusões, contribuições do trabalho, uma comparação dos trabalhos estudados com o trabalho realizado e são discutidos os trabalhos futuros.

## Capítulo 2 . Fundamentação Teórica

---

Vimos que devido à necessidade crescente das organizações compartilharem suas informações, várias estruturas, estão surgindo para facilitar todo o processo de compartilhamentos de dados espaciais. Muito embora, ao mesmo tempo em que surgem novas estratégias para facilitar o processo de troca de informações, novos desafios aparecem nesta área.

Neste capítulo são apresentados todos os conceitos que fundamentam o trabalho. Na seção 2.1 estão descritos vários conceitos e problemas de interoperabilidade que a comunidade de SIG distribuídos enfrenta. Na seção 2.2 são apresentadas as infra-estruturas de distribuição e localização de serviços em SIG (*spatial data infrastructure, Service oriented architecture, web services*, composição de serviços geográficos), finalizando, nas seções subseqüentes, é apresentada uma explanação sobre Web semântica, ontologias e reconciliação de ontologias.

### 2.1 Interoperabilidade em SIG Distribuído

A necessidade de compartilhar dados entre sistemas é um problema presente há muito tempo nas empresas. Com a Internet, essa necessidade ganhou maiores proporções, pois o objetivo maior é compartilhar dados e recursos de processamento (serviços) entre organizações. Essa necessidade também tem se revelado nas empresas e institutos de pesquisas que compartilham dados geográficos, e com isso abre espaço para a interoperabilidade, que é a habilidade de sistemas ou componentes cooperarem entre si, de forma independente, e trocando informações portáteis [Bis98]

Cada organização tem uma maneira própria de organizar seus dados, e quando decidem compartilhá-los, vários problemas de heterogeneidade ocorrem. Geralmente, os objetos são armazenados nas bases de dados sob forma da tupla < atributos temáticos(T), atributos geográficos(G), ID >, Bishr [Bis98] classificou os três tipos de heterogeneidade como:

- **Heterogeneidade semântica** - Um fato do mundo real pode ter mais de uma descrição nas bases de dados existentes entre várias disciplinas do conhecimento (heterogeneidade cognitiva), um segundo tipo seria o caso em que várias

entidades se referem ao mesmo conceito por meio de nomes diferentes. Um thesaurus poderia ser aplicado para resolver os problemas cognitivos relativos a nomes, porém quanto à divergência de conceitos o problema é mais complicado, principalmente quando não há um conjunto mínimo de definições. Uma possível solução seria capturar o papel dos objetos do mundo real, dentro de um contexto, como um conjunto de regras e cláusulas em uma base de dados e então desenvolver um mecanismo que mapeia os diferentes papéis usando a base de regras.

- **Heterogeneidade esquemática** - Objetos em um banco de dados são considerados como atributos em outro, ou classes de objetos podem ter diferentes agregações ou generalizações, embora descreva o mesmo objeto do mundo real. Uma solução seria prover uma unificação de esquemas (integração de esquemas), ou seja, promover o mapeamento entre os esquemas através de formalismos.
- **Heterogeneidade Sintática** - Cada banco de dados pode ser implementado em diferentes paradigmas, objeto-relacional, orientado a objetos. Outro tipo de heterogeneidade sintática está relacionado à representação geométrica dos objetos (*raster* e *vetorial*). Uma forma de resolver heterogeneidade entre esquemas conceituais seria através de um mapeamento bidirecional entre os modelos lógicos (orientado a objetos e relacional, por exemplo). Uma solução para heterogeneidade entre representação geométrica de objetos seria prover uma sintaxe comum para as representações dos objetos espaciais em todos GIS (por exemplo, GML e outros padrões).

## **2.2 Spatial Data Infrastructure e descoberta de recursos geoespaciais**

*Spatial Data Infrastructure* (SDI) trata de uma infra-estrutura para gerenciar, integrar, usar, publicar/acessar e compartilhar dados geoespaciais. É formado por vários órgãos (setores públicos, privados e terceiro setor) que cooperam mutuamente com o objetivo de estabelecerem padrões, regras e políticas que mantenham a colaboração [FGD05, RW01, DA05, ML05]. Com isso, usuários economizam recursos, tempo e esforços na coleta de dados espaciais, evitando a duplicação de gastos com a geração e manutenção do conteúdo geográfico e integração com outras bases.

Um SDI é formado por vários componentes que interagem entre si de diversas formas, tais componentes podem ser: políticas, canais de comunicação, padrões

técnicos, pessoas (incluindo parcerias) e dados. Estes componentes são classificados em duas categorias: a categoria mais fundamental formada por pessoas e dados e a categoria mais dinâmica que seria os componentes tecnológicos, políticas e padrões. Os SDIs são classificados em diversos níveis de acordo com a sua abrangência, podem ser: corporativos, locais, estaduais, nacionais, regionais, e globais.

Rajabifard [RW01] identifica dois modelos de SDI, o primeiro é baseado em produtos e o segundo baseado em processos. O modelo baseado em produto visa, principalmente, ligar bancos de dados existentes nos vários níveis políticos/administrativos da comunidade. O modelo baseado em processos centra em iniciativas para a formação de um *framework* para gerenciar o acesso aos dados. Em outras palavras, o objetivo maior por trás do projeto de SDI é melhorar os canais de comunicação para que a comunidade possa acessar as informações livremente, ao invés de centrar apenas na ligação das bases de dados.

Bernard e Craglia [BC05] realizaram um estudo sobre as principais tendências dos SDIs e é notado que a evolução tende para serviços, ou seja, cada vez mais a infraestrutura centra esforços para o modelo baseado em processos. É possível observar neste estudo, os esforços dos órgãos de padronização na definição de padrões de serviços de geoprocessamento e das vantagens do modelo orientado a serviços. É chamada de *(GI-) Service Driven Infrastructure* a próxima geração de SDI, onde as principais necessidades estão nas idéias de compartilhamento de funcionalidades, o que permite a derivação de novas funcionalidades a partir de combinações entre si.

Em seguida, apresentamos exemplos de SDI implementados em diversas partes do mundo:

- *NSDI*- O *National Spatial Data Infrastructure* é mantido pelo *Federal Geographic Data Committee* (FGDC)[Fed05] é um comitê formado por várias agências, organizado desde 1990, que promove o uso coordenado, compartilhamento, e disseminação de dados geoespaciais em uma base nacional, no caso, os Estados Unidos. Seus vários sub-comitês são organizados por temas de dados, que trabalham de forma integrada desenvolvendo padrões de metadados para as áreas referentes a cada sub-comitê. O NSDI é um conjunto de tecnologias, políticas, critérios, padrões e pessoas necessárias para promover o compartilhamento de dados espaciais através de todos os níveis de governo, setores privados, terceiro setor e academia. Provê uma estrutura ou base de práticas e relacionamentos entre geradores de informação e usuários que

facilitam o compartilhamento e o uso. São liderados hoje pelo FGDC, mais dois órgãos integram o projeto que é o Geospatial One Stop (GOS) [GOS07], que implementou os elementos básicos do NSDI visando prover um portal de Internet para facilitar a troca de dados em favor da decisão de acolher e encorajar parcerias entre as organizações, e o The National Map [TNM07], que provê uma base integrada de dados geográficos em parceria com produtores de conteúdo em todos os níveis.

- **INSPIRE** - É uma iniciativa para a criação de um *spatial information infrastructure* Europeu que fornece aos seus usuários serviços de informações espaciais integrados. Estes serviços deveriam permitir aos usuários identificar e acessar dados espaciais ou informações geográficas de uma vasta quantidade de fontes, que variam de níveis locais até os níveis globais, de forma interoperável. Os usuários alvos do INSPIRE são políticos, gerentes e tomadores de decisões das organizações [Ins07].
- **ANZLIC** - Projeto que define e coordena um SDI regional compreendendo a Austrália e Nova Zelândia. O ANZLIC advoga o uso de padrões comuns para garantir maior disponibilidade dos dados para os tomadores de decisões, melhorando as informações espaciais e serviços (bem como tudo que depende dos dados espaciais) disponíveis para o governo, empresas e a comunidade em geral. No governo, o ANZLIC está criando uma forte ligação entre decisões políticas e informações necessárias para implementá-las [Ans07]. No projeto ANZLIC existe o Australian Spatial Data Infrastructure (ASDI), o ASDI é definido através de quatro componentes [RW01]:
  - um *framework* institucional, que define políticas e arranjos administrativos para construir, manter, acessar e aplicar os padrões e os dados;
  - padrões técnicos, que define as características técnicas dos conjuntos de dados fundamentais;
  - conjunto de dados fundamentais, que são produzidos dentro do *framework* institucional e cumprem rigorosamente os padrões definidos;
  - uma rede de *clearinghouses*, que é o meio em que os dados fundamentais estão acessíveis para a comunidade, de acordo com as políticas de acessos definidas dentro do *framework* institucional e obedecendo aos padrões técnicos.

- **GSDI** - Em nível global há uma iniciativa chamada Global Spatial Data Infrastructure (GSDI). Nesta iniciativa, organizações regionais estão desempenhando um importante papel no GSDI. O GSDI está, atualmente, num estágio inicial, começando a desenvolver o próprio modelo da organização, políticas e *framework*, bem como um conjunto de diferentes grupos de trabalho para projetar e conduzir as pesquisas em outros importantes componentes do GSDI [Gsd05].

### 2.2.1 Evolução da descoberta de recursos em SDI

O processo de troca de dados padronizados possui um mecanismo de publicação/descoberta extremamente defasado, pois não existe uma forma ou estrutura para consultar os metadados das informações compartilhadas. Para facilitar o acesso aos dados espaciais, surgiram os *clearinghouses*, que estabelecem um sítio centralizado cujos metadados de várias fontes podem ser encontrados, provendo, desta maneira, serviços de buscas, visualização e transferência de dados espaciais [CBR+04, Fed05]. Os *clearinghouses* permitem que provedores de dados tornem suas ofertas conhecidas pelos usuários por meio de descrições de dados e com instruções de como acessar os dados.

Com o avanço das tecnologias orientadas a serviços e a evolução dos SDIs para um modelo centrado em serviços, as tecnologias baseadas em *clearinghouses* se tornaram obsoletas, pois eram completamente centradas na recuperação de dados. Então, surgiu a segunda geração de *clearinghouses*, chamada de Geoportais. Os Geoportais consistem de portais web (sítio web) que organizam e catalogam recursos como serviços, informações, ferramentas de busca e aplicações. Eles provêm um meio para consultar os metadados sobre dados e serviços e ligam diretamente aos provedores dos recursos. [LGM+05]

Os Geoportais [BKA+05, ML05, Tai05] são divididos em dois grupos: geoportais catálogos, que organizam e gerenciam o acesso à informação espacial; geoportal de aplicações, que provê *web services* geográficos dinâmicos, por exemplo, o Mapquest\* que provêm serviços de rotas. Na figura 1 [DA05], é possível identificar o papel dos geoportais e como as tecnologias de web services podem tornar possível o acesso mais direto aos dados do catálogo.

O *Federal Geographic Data Committee* americano numa tentativa de modernizar os serviços de acesso aos dados lançou o GOS [GOS07], que é considerado a evolução dos *clearinghouses* para os Geoportais.

\* [www.mapquest.com](http://www.mapquest.com)

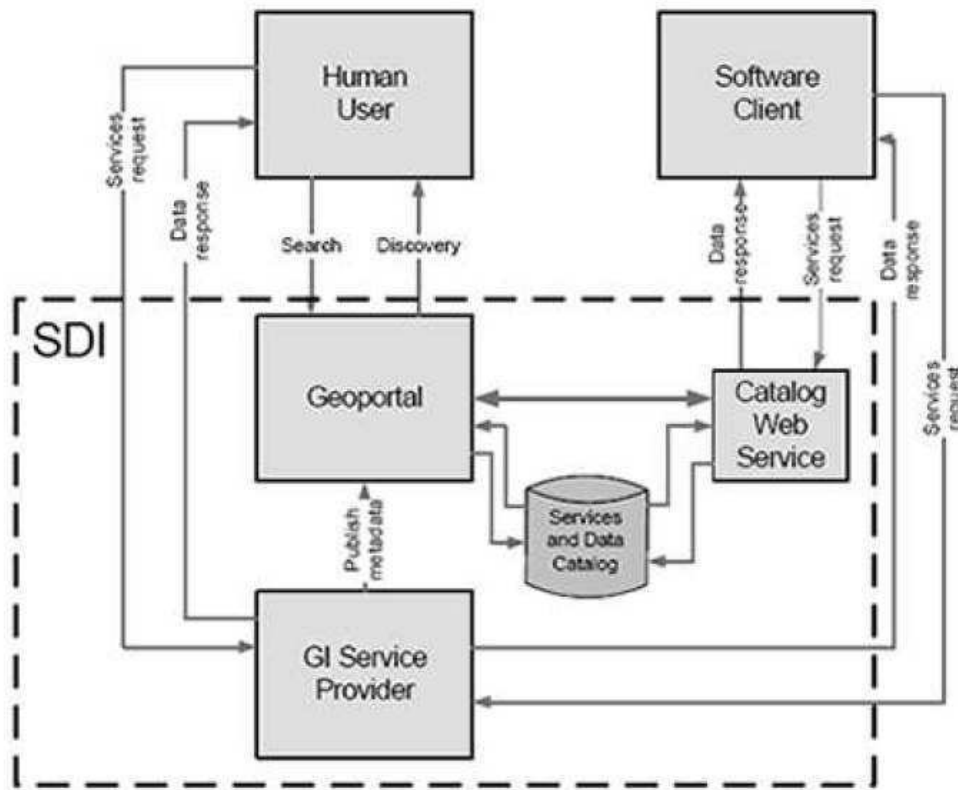


Figura 2.1: SDI e o papel dos Geoportais Fonte: [DA05]

### 2.3 Modelos de arquitetura para desenvolvimento de sistemas de informações

É possível observar que da mesma forma que os sistemas de informações tendem para uma arquitetura orientada a serviços, os SDI seguem as mesmas tendências.

Primeiramente com as **arquiteturas centralizadas**, bastante populares nos anos 80, ambientes como os mainframes apresentavam interface não amigável e o acesso à máquina só poderia ser através de algum terminal com acesso direto. Num segundo momento, a **arquitetura duas camadas** surgiu para fornecer interfaces mais amigáveis com os PCs e aumentar o acesso às aplicações. Em seguida, surgiu a **arquitetura três camadas** com o objetivo de diminuir o número de conexões com o servidor de dados, aumentando a escala, e facilitar a manutenção da lógica de negócio. Com o avanço das conexões de rede, a **arquitetura distribuída em n camadas** surge para aumentar a escala dos sistemas e maior acesso aos dispositivos móveis (quiosques, celulares, PDA



etc.), facilitar a manutenção, pois vários serviços (segurança, *resource pooling*, balanceamento de carga, gerenciamento de transações distribuídas, *impedance mismatch*) já estariam implementados nos *middlewares* [BCP03, Fow02].

Continuando com esta evolução os *web services* surgem como solução alternativa a estas arquiteturas, pois permitem alto grau de reutilização, difusão e integração dos sistemas.

Uma definição para web services que se pode encontrar na World Wide Web consortium (W3C) [W3C07] é "*uma aplicação de software identificada por uma URI, cujas interfaces e associações devem estar bem definidas, descritas e descobertas com artefatos em XML. Um web service provê interações diretas com outros agentes de software usando troca de mensagens baseadas em XML através dos protocolos da Internet*".

Por trás desta tecnologia está o paradigma orientado a serviços. Sua principal característica é que a funcionalidade estando disponível como serviço facilita a composição, onde programas executam outros programas, que podem ou não pertencer à mesma empresa. Desta forma, as organizações podem cooperar mantendo seus dados e lógicas de negócio completamente separados e fracamente acoplados, obtendo maior flexibilidade, robustez e autonomia.

Neste paradigma uma empresa que trabalha com vendas na internet, por exemplo, pode ter disponível vários serviços de transferência automática de fundos para tratar com o pagamento das compras, assim, os serviços podem concorrer para atender a demanda da empresa de venda, negociar para chegar num acordo de utilização, e, uma vez, estabelecido o acordo vão monitorar o funcionamento de ambas as partes (cliente e provedor do serviço) e tratar de segurança.

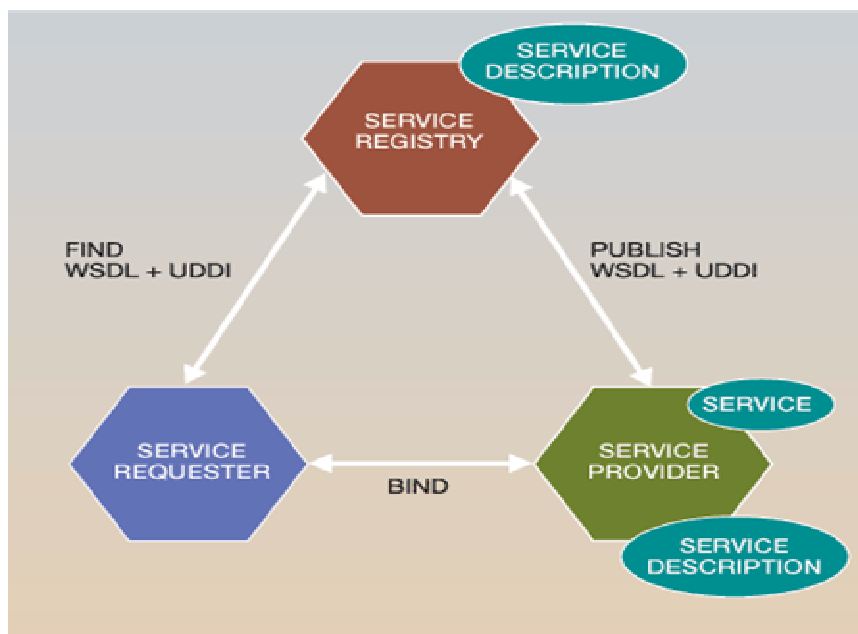


Figura 2.2: web services: Atores, objetos e protocolos (Fonte: [IWS07])

Na Figura 2 é apresentada a arquitetura básica de *web services*. É possível observar um serviço que requisita outro (*service requester*), para isso ele primeiro efetua uma busca no catálogo usando o serviço de nomes e diretório (*Universal Description Discovery and Integration* — UDDI —), chamado na figura de *service registry*. Obtendo o serviço a ser utilizado o próximo passo é a associação (*bind*) entre provedor (*service provider*) e o serviço que o utilizará (*service requester*). Os serviços estão descritos nos diretórios através de *Web Services Description Language* (WSDL), por meio desta linguagem, as descrições podem ser pesquisadas automaticamente nos serviços de registros através de ferramentas. Através dessas descrições, as interfaces dos provedores de serviços (*service provider*) ficam bem definidas e publicadas para que os serviços clientes (*service requester*) possam utilizar os serviços. Outro componente base da arquitetura é o *Simple Object Access Protocol* (SOAP), que fornece um *framework* para protocolo de comunicação entre todas as partes envolvidas na arquitetura. O protocolo SOAP define uma forma de organizar a informação usando XML de maneira estruturada e com um esquema definido para que possam ser trocadas entre as máquinas envolvidas [ACK+03].

Além dessa arquitetura básica, os órgãos de padronização de web services (W3C, OASIS) definem vários outros conjuntos de especificações que regulamentam questões como segurança, coordenação (*Service coordination*), composição de serviços

(BPEL4WS[ACD+04]) e os protocolos verticais (segurança, gerenciamento, qualidade de serviço, etc.). Além disso, toda essa arquitetura precisa de ambientes (*containers*) que implementam parte desses protocolos e mantêm o ciclo de vida dos *web services*. Existem várias especificações para *web services containers*. Alonso [ACK+03] resalta alguns problemas com registros em *web services*:

- Catálogos UDDI não dão suporte à descoberta distribuída através de propagação da consulta por vários registros;
- A especificação é insuficiente na definição de um modelo de dados para catálogos, e deixa a desejar na descrição dos tipos de serviços, apresentando poucos recursos para essa tarefa;
- Devido aos problemas semânticos e de descoberta automática de recursos, existem problemas com a associação (*binding*) dinâmica. Devido a isso, muitas empresas são desencorajadas em usarem UDDI e os serviços são associados estaticamente;
- As aplicações que utilizam *web services* têm se limitado apenas para usar um conjunto limitado de interfaces. Tanto por falta de um padrão ou por falta de uma descrição de serviço de mais alto nível;

### 2.3.1 Modelos para composição de GIServices

Encadeamento de serviços é o processo de combinar ou fazer uma seqüência de serviços (*web services*) complementares encadeados para criar soluções customizadas [Alm03]. A combinação de serviços é um trabalho não trivial, deve-se levar em consideração diversos fatores como tratamento de falhas no encadeamento, tratamento de situações inesperadas, pagamento, monitoramento, entre outros. Além disso, para elaborar um bom encadeamento deve-se ter em mente os tipos de serviços que estão envolvidos, desta forma, é possível aperfeiçoar o desempenho do processamento de resultados. Os GIServices são classificados como:

- **Serviços de dados** (*OGC Web Mapping, Web Coverage, web Feature services*) - oferecem dados customizados para os usuários [OGC07]. Estes serviços são fortemente acoplados a um conjunto específico de dados.
- **Serviços de processamento** - Provêm operações para o processamento genérico ou transformação de dados de forma especificada pelo usuário. Podemos citar serviços de conversão de coordenadas geográficas,

rasterização/vetorização, sobreposição (*overlay*) de mapas, manipulação de imagens, serviços baseados em localização (Roteamento) [Alm03, OGC07].

- **Serviços de registro/Catálogos** - são usados para classificar, registrar, descrever, pesquisar, manter e acessar informação a respeito de *web services* (extensão do catálogo UDDI) [Per03].
- **Serviços de imagens (*Portrayal Services*)** - São serviços que provêm visualização de informações geográficas. São componentes que, dada uma ou mais entradas, produzem saídas “renderizadas” (imagens, mapas). Trabalham em conjunto com os serviços de dados e/ou processamento [Per03].

A idéia chave para o desenvolvimento de cadeias de serviços é criar uma arquitetura que possua maior “escalabilidade” e sem comprometer o desempenho. Vários fatores devem ser analisados, como o grau de transparência da cadeia para o cliente que a utilizará e quantidade de responsabilidade na coordenação transferida ao cliente. Manter os metadados do encadeamento pode ser uma solução importante para uma cadeia onde o grau de transparência é alto. O tratamento de erros e casos especiais na cadeia pode ser tratado com metadados. Seguem os tipos de encadeamento de serviços descritos por Alameh [Alm03]:

- ***Encadeamento de serviços coordenado pelo cliente (Client-Coordinated service chaining)*** - O cliente define e controla a ordem de execução dos serviços individuais na cadeia. Um ponto positivo deste modelo é o poder de decisão que o cliente tem para escolher os serviços a serem utilizados, obtendo maior flexibilidade. A grande desvantagem deste modelo é o grau de transparência baixo, pois o cliente deve se preocupar com as saídas e as entradas dos serviços e fazer todo o tratamento de erros e controle de metadados.
- ***Encadeamento estático de serviços utilizando serviços agregados (Static Chaining using aggregate services)*** - A cadeia de serviços fica totalmente escondida do cliente, tendo este apenas um ponto de comunicação com a cadeia. A vantagem é que o grau de transparência é alta e o cliente não tem que tratar erros na cadeia, nem situações especiais. Porém, essa abordagem pode dificultar a manutenção da robustez da cadeia, pois fica muito complicado a manipulação de metadados no tratamento de erros, exceções, pagamento etc. No encadeamento coordenado pelo cliente, os problemas são tratados rapidamente, já que o cliente tem ponto de contato com todos os serviços utilizados.

- ***Encadeamento de serviços gerenciado pelo serviço de Workflow com serviços mediadores (Workflow-managed service chaining with mediating services)*** - esta idéia visa combinar a simplicidade de serviços agregados estaticamente com a flexibilidade e controle inerentes em encadeamento de serviços coordenados pelo cliente. O cliente pode selecionar uma cadeia predefinida ou selecionar uma cadeia elaborada pelo *mediating service*. *Mediating services* constroem dinamicamente e gerenciam cadeias de *web services*. Baseados nos requisitos dos usuários determinam fontes de dados e serviços, recuperam e procuram os dados, compilam e retornam resultados para o cliente. Esta idéia é oriunda de banco de dados distribuídos quando o mediador recebe a consulta e a divide em subconsultas despachando-as para as fontes de dados devidas. *Mediating services* consultam os catálogos e podem lançar mão de ferramentas de busca para localizar os serviços.

## **2.4. Padrões propostos pelo OpenGeospatial Consortium**

### **2.4.1 Geographic Markup Language**

*Geographic Markup Language (GML)* [CDL+04], é uma linguagem de marcação para definir feições geográficas com o objetivo de facilitar a troca de dados entre sistemas e/ou domínios, otimizar o armazenamento de dados geográficos. Resumindo os objetivos de GML temos os seguintes tópicos:

- prover uma estrutura aberta para a definição de esquemas de aplicações e objetos geoespaciais;
- permitir perfis que suportem subconjuntos das capacidades declarativas de GML;
- dar suporte à descrição de esquemas de aplicações geoespaciais para domínios específicos e comunidades de informação;
- permitir a criação e manutenção de esquemas e dados geoespaciais distribuídos e relacionados;
- facilitar o compartilhamento de dados geográficos entre organizações;
- servir como ferramenta para a interoperabilidade de Sistemas de Informação Geográfica (SIG) de uma maneira incremental.

Nesta terceira versão, o padrão GML visa suprir as limitações das versões passadas, como:

- Além da representação espacial de objetos em duas dimensões, representar feições mais complexas como, formas tridimensionais, feições com topologia em duas dimensões, feições com atributos temporais, feições dinâmicas, coberturas e observações.
- prover um maior suporte às propriedades das feições e aos objetos complexos;
- representar sistemas de referências espaço-temporais, unidades de medidas e padrões de informação;
- representar estilos default para feições e cobertura de visualização;
- maior conformidade com o padrão ISO 19100.

No trecho de código 2.1, é apresentado um exemplo de um rio da Paraíba codificado em GML:

```
<gml:featureMember>
  <ParaibaRios fid="xxx">
    <nome>Rio Soledade</nome>
    <geometria>
      <gml:LineString
        srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
        <gml:coordinates>-36.35456,-7.07528 -36.36055,-
          7.08547 -36.36781,-7.09561 -36.36768,-7.10622 -
          36.37196,-7.11669 -36.37224,-7.1267 -36.37243,-
          7.13689 -36.37243,-7.14704 -36.37023,-7.15738 -
          36.37083,-7.16743 -36.3729,-7.17763 -36.37562,-
          7.18783 -36.38252,-7.19792 -36.39255,-7.2048 -
          36.39237,-7.2149 -36.39412,-7.22505 -36.39067,-
          7.23506 -36.38534,-7.24508 -36.39537,-7.24837 -
          36.39197,-7.25862 -36.38402,-7.26873 -36.37482,-
          7.27875 -36.38057,-7.2888 -36.39066,-7.29876 -
          36.39254,-7.30886 -36.39564,-7.32604 -36.39485,-
          7.33623 -36.39388,-7.34642 -36.38381,-7.35222 -
          36.39131,-7.36231 -36.38129,-7.36843 -36.3798,-
          7.37856 -36.38,-7.38864 -36.37717,-7.39841
        </gml:coordinates>
      </gml:LineString>
    </geometria>
  </ParaibaRios>
</gml:featureMember>
```

**Trecho de código 2.1: Exemplo de uma feição codificada em GML**

## 2.4.2 Padrões de serviços

O OGC é um consórcio formado por vários membros (aproximadamente 300) que determinam padrões de interoperabilidade para geoprocessamento. Através do OGC são propostos, melhorados e publicados vários serviços e modelos de arquiteturas para aplicações envolvendo dados espaciais. Seguem os principais serviços propostos pelo OGC:

- **Web Mapping service (WMS)** - Padroniza um serviço de mapas. Clientes solicitam mapas através das camadas disponíveis e outros parâmetros como tamanho do mapa, área de cobertura (coordenadas), sistema de referência utilizado. A especificação define três interfaces básicas: *getCapabilities*, retorna metadados do serviço, ou seja, descrição sobre os dados contidos no serviço, e alguns parâmetros aceitáveis nas requisições; *getMap*, retorna uma imagem de um mapa cujos parâmetros espaciais e dimensionais foram definidos; *getFeatureInfo* (opcional), retorna informação a respeito de determinados objetos espaciais do mapa.
- **Web Feature service (WFS)** - Trata com a manipulação direta de objetos (*features*) geográficos em um servidor. Sua forma de interação é através de GML, ao solicitar um objeto (feições) geográfico o servidor WFS retorna uma representação em GML do objeto solicitado. Além disso, o serviço trata com inserção, remoção, atualização e descoberta de objetos geográficos. Suas interfaces básicas de leitura (*read-only*) são: *getCapabilities*, que descreve características gerais das informações do servidor; *describeFeatureType*, que descreve a estrutura dos tipos de objetos providos pelo servidor; *getFeature*, que retorna a representação em GML de objetos disponíveis ou selecionados pelo cliente; *transaction*, interface utilizada para a operações de modificação dos objetos no servidor.
- **Web Coverage service (WCS)** – Este serviço provê um suporte à recuperação de dados sob forma de coberturas. Coberturas são representações digitais de fenômenos que não possuem espaço bem definido. Por exemplo, representações digitais de informações de clima, de tempo, etc. Diferente do WFS, que retorna ao usuário representações digitais discretas do espaço-tempo (feições), o WCS retorna para os usuários fenômenos mais complexos que podem variar no espaço-tempo (coberturas). O WCS provê três operações: *getCapabilities*, que retorna um documento XML que descreve informações a respeito do próprio serviço; *describeCoverage*, que retorna para o cliente a descrição completa de uma ou mais coberturas existente no servidor; *getCoverage*, que retorna uma cobertura, esta operação, normalmente solicitada depois do *getCapabilities* e depois do *describeCoverage*.

- **Web Gazetteer service** – este serviço retorna as feições geográficas levando em consideração as associações semânticas entre elas, as associações são construídas através de identificadores dos objetos, por exemplo, nomes dos locais. Cada instância de um serviço de *gazetteer* tem um vocabulário associado, dessa forma pode explorar todas as capacidades hierárquicas, como conceito de termo mais geral, termo mais específico, termo relacionado. Sua interface é semelhante ao WFS.
- **Web coordinate transformation service** - Serviço que provê uma interface para transformação de feições geográficas codificadas em um sistema de referência para outro. Podemos identificar algumas operações como: *getCapabilities*, que retorna propriedades das informações contidas e formas de interações; *transform*, requisição para disparar o processo de transformação devem ser informados os sistemas de coordenadas origem e o desejado; *isTransformable*, que retorna se o serviço consegue transformar a partir de um sistema de coordenadas para outro, trabalha tanto com feições quanto com coberturas;
- **Catalog service** - Especifica uma interface para associar, pesquisar, descrever, publicar e acessar catálogos digitais de metadados para dados e serviços espaciais. *Catalogue services* suportam uma linguagem de consulta (*Common catalog query language*) e um modelo(esquemas de metadados) de descrição de recursos, baseado na norma ISO19115-Geographic information metadata. De forma padronizada, vários catálogos podem atender a mesma consulta. As principais interfaces são: *getCapabilities*, que retorna informações sobre o servidor; *query*, operação que realiza a consulta sobre os metadados; *present*, retorna os metadados de uma ou mais referências; *transaction*, para iniciar um conjunto de operações para modificar os itens do catálogo; e *describeRecordType*, descreve o tipo da referência passada.
- **Web Registry Service** - A idéia deste serviço é localizar catálogos (*Catalog Services*) distribuídos, fornecendo um serviço de localização e registro de metadados dos servidores disponíveis. Esta especificação ainda está sob debate na OGC. Suas principais interfaces são: *getCapabilities*, que retorna informações do servidor; *getDescriptor*, que retorna os servidores registrados, sendo possível passar um filtro para essa consulta; *registryService*, responsável por registrar um servidor no WRS.



Todas essas especificações surgiram antes da popularização de *web services*, antes os serviços eram definidos para o protocolo HTTP, porém com a sedimentação das tecnologias *web services* o OGC acompanhou o mercado e publicou novas revisões de especificações para todos os serviços tendo como base as tecnologias de *web services*.

### **2.4.3 Padrão ebRIM para Modelagem de catálogos de serviços OGC**

O OGC definiu vários padrões de serviços como vimos na seção 2.4.2, porém a especificação do *catalog service* é abstrata, estabelecendo um *framework* onde os usuários não possuem um esquema de dados para descrever os recursos do catálogo. A especificação *OGC Catalogue Services - ebRIM (ISO/TS 15000-3) profile of CSW* [Mar04] surge para modelagem de catálogos que obedecem a especificação de *catalog services*.

A especificação define um esquema conceitual para a informação gerenciada pelo *catalog service*, este modelo constitui um esquema público que admite uma ou mais associações (*binds*) entre dados usados para representar o conteúdo do catálogo.

O serviço de *Catalog Service* desempenha um papel importante na localização de serviços e recursos geográficos, pois, diferente dos *web services* que não são padronizados pela OGC, os serviços de geoprocessamento possuem um forte acoplamento com o tipo de dado que é fornecido. Esta característica adiciona ainda mais complexidade no processo de recuperação de serviços do catálogo. A especificação permite aos catálogos funcionarem de forma isolada ou numa federação, desta forma, vários catálogos cooperam para aumentar o escopo das consultas por serviços e/ou conjuntos de dados. O processo de propagação das consultas através de catálogos federados funciona de forma transparente para o cliente e este comportamento é previsto pelo *application profile*.

O principal propósito da especificação ebRIM para *catalog services* é representar os recursos catalogados e seus relacionamentos, provendo um esquema lógico de busca e navegação no conteúdo dos catálogos. Na figura 2.3, um modelo em alto nível é descrito, onde é apresentado apenas classes na especificação e não são definidos métodos.

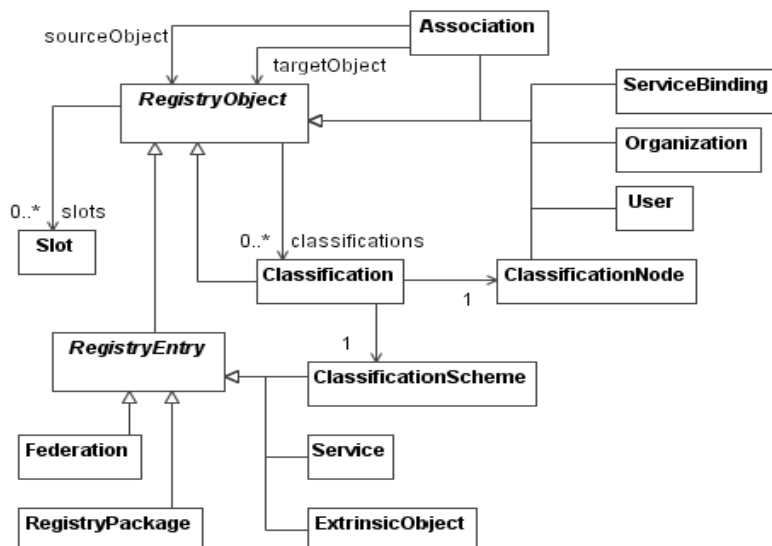


Figura 2.3: Visão de alto nível do modelo descrito no padrão ebRIM (Fonte: OGC07)

O padrão ebRIM define um sistema de busca distribuída simples, baseada em mediador, cujo mecanismo dá suporte a consultas síncronas ou assíncronas que são propagadas pela federação de catálogos. Dentro da federação, um catálogo desempenha o papel de mediador despachando as consultas e acumulando os resultados, observe a figura 2.4. O usuário pode optar por realizar a consulta apenas no catálogo local ou de forma distribuída configurando apenas os parâmetros (*hopCount* para especificar o escopo da propagação e o *requestID*, que é obrigatório para a identificação das mensagens distribuídas).

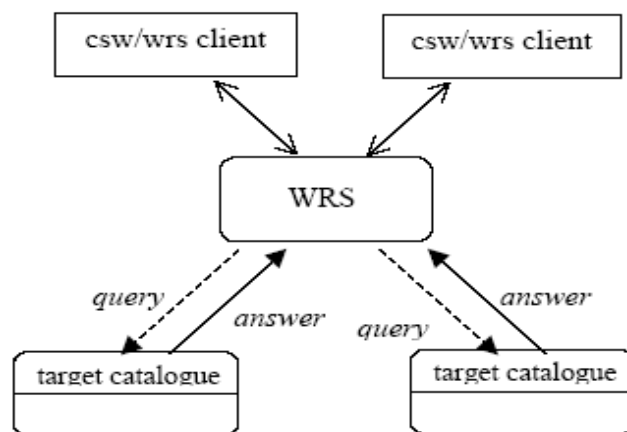


Figura 2.4: Mediador nas consultas distribuídas (Fonte: OGC07)

Podemos observar na literatura um exemplo de uso deste padrão em Zhao [ZCL+04], neste trabalho, é proposto um modelo de integração entre o *Metadata Catalog Service* (MCS), que é o serviço de armazenamento de metadados e localização

de recursos em grids computacionais [FKT01, VBR05], com o modelo proposto pela especificação WRS ebRIM. Desta forma, agrega-se semântica às consultas por dados geoespaciais em ambientes distribuídos baseados em grid computacionais. Ainda em seu trabalho, o autor ressalta que os dois modelos se complementam, pois a especificação WRS ebRIM define uma representação dos dados em um nível conceitual enquanto MCS define uma implementação para armazenamento e acesso aos metadados. A idéia é fazer mapeamentos semânticos entre os dois modelos através de ontologias (OWL Web Ontology Language [Owl05]) e realizar inferência automática sobre os dois modelos. Portanto, os usuários submetem consultas no nível semântico de geoprocessamento, através das ontologias a consulta é mapeada para que sejam realizadas no *grid* computacional através do MCS.

## **2.5 Web semântica e ontologias**

Depois de participar da concepção da internet atual, Tim Berners-Lee publicou um conjunto de artigos [SBH06, BHL01] que apresentaram a Web semântica. Em seus artigos, o autor afirma que no futuro teremos uma rede mundial, nos moldes da internet atual, porém com todo o seu conteúdo processável pelas máquinas, ou seja, os computadores (via agentes de software) iriam interpretar o conteúdo da Web e conseguir operacionalizar tarefas que, até então, são impossíveis de serem realizadas, devido à estrutura da Web atual. Podemos citar como exemplo, várias operações como o processo de negociar o melhor preço num leilão, montar um pacote de viagens baseado no perfil do usuário, disparar um processo de compra quando faltarem mantimentos na casa dos usuários, enfim várias tarefas que hoje as ferramentas de software, podem realizar apenas sob o controle de humanos. Desta maneira, podemos definir Web semântica como um *framework* comum que possibilita a partilha e o reuso de dados entre aplicações, empresas e comunidades. Esta sendo concebida a partir de um esforço colaborativo liderado pela W3C, que define padrões para Web, pesquisa novas tecnologias e trabalha junto ao meio industrial para evoluir tais conceitos.

Como sabemos, atualmente, as informações da Web estão dispostas de forma que apenas humanos possam ler e entender o conteúdo. A grande contribuição da Web semântica é melhorar a Web adicionando meios para que softwares possam interpretar o conteúdo desta. Para tanto, é preciso que as informações da rede estejam estruturadas de maneira significativa, além disso, é necessário que se defina um conjunto de regras de inferência para o processamento automático destas informações. O primeiro passo, para

que se possa definir um conjunto de “conceitos estruturados”, seria um acordo entre vários órgãos de uma determinada área do conhecimento para definir um vocabulário [SBH06]. Tais iniciativas já estão sendo tomadas, por exemplo, no Reino Unido com o Integrated Public Sector Vocabulary ([www.esd.org.uk/standards/ipsv](http://www.esd.org.uk/standards/ipsv)) e outras iniciativas européias. Um segundo passo, seria formalizar e estruturar esse vocabulário para que os agentes de software possam interpretar e inferir conhecimento a partir dos conceitos lá contidos.

É com esta finalidade que surgiram as ontologias no mundo computacional. Podemos definir ontologias como sendo uma especificação formal de uma “conceitualização” compartilhada por um determinado grupo [GG95]. Onde “conceitualização” pode ser entendida como um conjunto de conceitos (vocabulário) que define o conhecimento em determinada área do conhecimento.

Atualmente, existem vários recursos para formalizar as ontologias, dentre as quais podemos citar o Resource Description Framework (RDF) [RDF07], que é um padrão recomendado pela W3C ([www.w3c.org](http://www.w3c.org)), órgão que define vários padrões para a internet. Desta forma, RDF define uma linguagem e um conjunto de regras para representar o conhecimento através de triplas (sujeito, predicado e objeto) que formam uma rede de conhecimento baseada em grafo, cujos vértices são identificados através de URIs (Universal Resource Identifiers). Na figura 2.5 segue um exemplo de um grafo representando algumas associações gráficas em grafos de RDF, em seguida, na figura 2.6 é apresentado o mesmo conhecimento utilizando a linguagem RDF/XML.

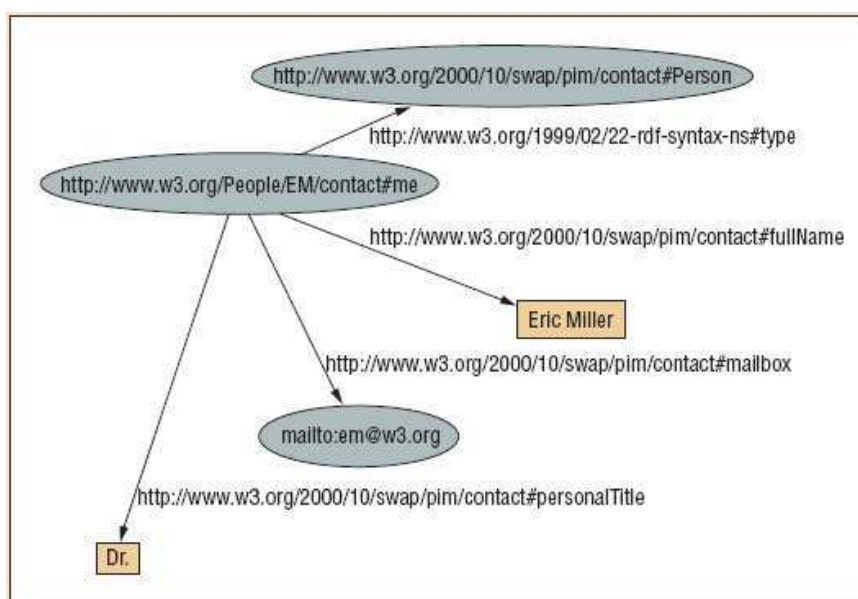


Figura 2.5: Mostra uma representação de um conhecimento. [SBH06]

```

<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">
  <contact:Person rdf:about="http://www.w3.org/People/EM/contact#me">
    <contact:fullName>Eric Miller</contact:fullName>
    <contact:mailbox rdf:resource="mailto:em@w3.org"/>
    <contact:personalTitle>Dr.</contact:personalTitle>
  </contact:Person>
</rdf:RDF>

```

Figura 2.6: Mostra o exemplo do grafo acima codificado na linguagem RDF/XML

Além de RDF, a W3C definiu outras linguagens para formalizar as ontologias. Uma das mais evidentes no momento é a OWL [Owl05], que surgiu a partir da evolução de duas outras linguagens que formalizavam ontologias (DAML + Oil), assim, é uma linguagem mais completa que RDF, pois é possível com OWL formalizar conhecimento com maior poder de expressividade e definir regras de inferência. Na figura 7 segue um exemplo da linguagem OWL. Atualmente, OWL possui três versões:

- OWL Lite - provê um conjunto mínimo básico da linguagem, permitindo, assim, um poder de expressividade menor;
- OWL DL – abrange um conjunto maior de construções, permitindo, um poder maior de expressividade em relação a OWL Lite. Esta versão da linguagem possui a característica de ser determinística, ou seja, as máquinas de inferência conseguem executar uma seqüência de passos finitos sobre esta linguagem.
- OWL FULL – abrange um conjunto completo de todas as construções da linguagem, possibilitando o maior poder de expressividade dentre as três versões, porém tem a desvantagem de não ser “decidível”, ou seja, não existe a garantia que alguma máquina de inferência que esteja processando conhecimento a partir da linguagem, execute uma seqüência de passos finitos.

```

<owl:Class rdf:ID="Açude">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Corpodagua"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Rio">
  <rdfs:subClassOf rdf:resource="# Corpodagua "/>
</owl:Class>

```

**Trecho de código 7: Formaliza uma herança entre a classe Corpodagua e as subclasses rio e açude**

Além destas ferramentas, a W3C definiu a OWL-S, que é uma linguagem derivada de OWL e que serve como uma ontologia para descrever o domínio de *web*

*services*. Com OWL-S é possível adicionar descrições semânticas aos serviços, de forma que agentes de softwares possam encontrar tais serviços automaticamente. A linguagem OWL-S é uma ontologia de alto nível composta por sub-ontologias que definem o *profile* (especifica o que cada serviço faz, sendo responsável pela parte da publicação do serviço em catálogos, o que facilita para que o serviço seja encontrado por agentes), *grounding* (especifica os detalhes de como os agentes podem acessar o serviço, inclui detalhes de protocolo de comunicação, formatos de mensagens, endereços de acessos, etc.) e o *process* (especifica como acessar o serviço, detalhando o conteúdo de cada requisição, entradas e saídas dos serviços, bem como pré-condições e pós-condições). Ainda é possível agregar a estas ontologias, mais outras que podem definir o conceito de tempo, parâmetros de qualidade de serviço, e outros recursos mais.

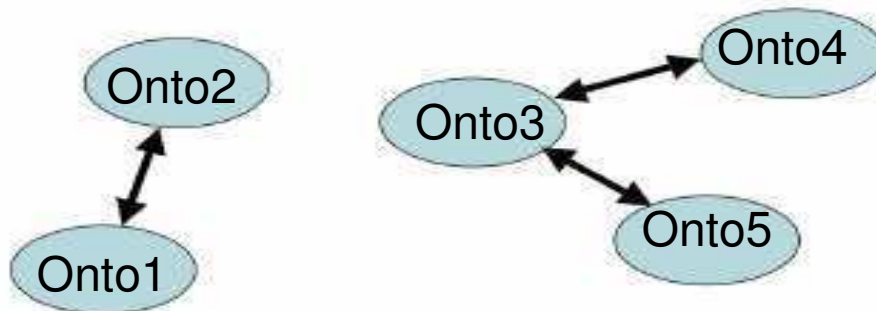
### **2.5.1 Reconciliação de ontologias**

Há um crescente uso das ontologias para formalizar conhecimentos em diversos grupos (acadêmicos, industriais e outros) ao redor do mundo. Porém, quando estes grupos tentam se comunicar (através de aplicações ou sistemas que trocam informações, ou alguma outra forma de troca de dados) surge um grande problema, que é a diferença entre os conceitos.

Para tratar este eminente problema, várias pesquisas [HPS04, BFM+03, CSC04, VS06] se dedicaram para tratar a reconciliação de ontologias. Este processo, geralmente, é gerenciado por uma pessoa humana e divide-se em três partes: (1) Fusão (*Merging*), que é o processo de gerar uma nova ontologia a partir da unificação de várias outras; (2) Alinhamento (*Aligning*), que é um processo usado quando se deseja manter duas ou mais ontologias coerentes e consistentes, porém separadas, este processo geralmente resulta num arquivo de mapeamento entre as ontologias em questão; (3) Integração, que é um processo semelhante ao processo de fusão, porém difere na ontologia final obtida, pois esta deve conter apenas os principais conceitos, enquanto no processo de fusão, a ontologia resultante contém todos os conceitos das ontologias fundidas.

Atualmente, existem várias arquiteturas para implementar alinhamento de ontologias, a mais simples, é o processo de mapear ontologias entre si, por exemplo, gerar um arquivo de mapeamento que contenha as correspondências entre os conceitos de uma ontologia A para um ontologia B. A vantagem desta técnica é a simplicidade e a flexibilidade de se manter as ontologias separadas. A grande desvantagem é que esta

técnica não prover maior “escalabilidade” quando se precisa tratar com um número elevado de ontologias, desta forma, seria necessário gerar  $O(n^2)$  mapeamentos para  $n$  ontologias. Na figura 8 é ilustrada esta técnica, cada elipse representa uma ontologia.



**Figura 2.7** Ilustra mapeamentos entre as ontologias individuais [Fonte: HPS04]

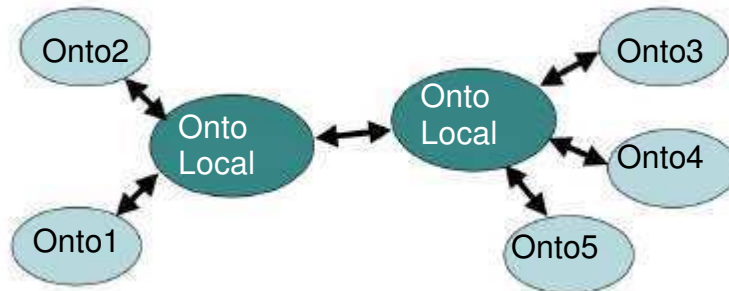
Com o objetivo de gerar um número menor de mapeamentos, surge um segundo processo. Para implementar esta técnica, é necessário gerar uma ontologia global a partir de todas as outras ontologias individuais e, em seguida, definir mapeamentos de todas as ontologias envolvidas para a ontologia global. A grande vantagem desta abordagem é a diminuição do número de mapeamentos que são necessários gerar, pois são evitados mapeamentos diretos entre todas as ontologias envolvidas, tornando-se necessário apenas um número  $O(n)$  mapeamentos para  $n$  ontologias individuais. Entretanto, desenvolver uma ontologia global que atenda aos requisitos de todas as ontologias envolvidas é um processo muito difícil e possui um alto custo, em certos casos, essa operação chega a ser inviável. Outra desvantagem é a perda da flexibilidade, pois uma vez que se muda a ontologia global ou uma ontologia individual, esta deve se refletir em todos os mapeamentos da rede. A figura 9 ilustra um exemplo desta estratégia.



**Figura 2.8:** Ilustra a técnica onde ontologias individuais possuem mapeamentos para a ontologia individual [HPS04]

Para minimizar as desvantagens das técnicas abordadas nos parágrafos anteriores, surge uma terceira técnica híbrida. Esta nova estratégia consiste em se elaborar agrupamentos (*clusters*) semânticos de ontologias locais, onde cada ontologia individual possui mapeamentos para a ontologia global local do seu respectivo

agrupamento, e, caso necessário, os agrupamentos podem ter mapeamentos entre as ontologias globais locais, permitindo a troca de informações entre os agrupamentos. Desta forma, esta técnica diminui o alto custo de se gerar uma ontologia global para todas as ontologias individuais e mantém um número bastante factível de mapeamentos entre as ontologias. Além disso, mantém certo grau de flexibilidade, pois mudanças ficam mais regionalizadas nos agrupamentos. A figura 10 descreve graficamente a estratégia híbrida descrita.



**Figura 2.9: Ilustrando os mapeamentos entre as ontologias locais de cada cluster e os mapeamentos entre as ontologias locais e individuais [HPS04].**

## 2.5 Considerações finais

Vimos, neste capítulo, a fundamentação teórica para esta dissertação. Inicialmente, apresentamos um discurso introdutório sobre os problemas de interoperabilidade em sistemas de informações geográficas distribuídos. Logo em seguida, foi apresentada a evolução dos SIG distribuídos e ferramentas de descoberta de dados espaciais, primeiramente com os clearinghouses, geoportais e, por fim, apresentamos a idéia de *Spatial Data Infrastructure*. Neste capítulo, vimos, também, as diversas tecnologias de *web services* e composição de serviços espaciais. Ainda neste capítulo, apresentamos os padrões de interoperabilidade e serviços da OGC. Por fim, exploramos vários conceitos relativos à Web semântica. Definimos o conceito de ontologia e exemplificamos com as ferramentas de formalização mais utilizadas no momento, que são RDF e OWL. Finalmente, discutimos as várias técnicas de reconciliação de ontologias e um exemplo prático da sua utilização.

No capítulo que segue, é exposta uma análise sobre os vários trabalhos relacionados à dissertação.



## Capítulo 3 . Trabalhos Relacionados

---

Neste capítulo, são abordados alguns trabalhos relacionados ao SWS-GIS. Ressaltamos que nosso foco primário não é localização de dados, mas a localização de serviços espaciais. Para cada trabalho abordado, apresentamos uma descrição das suas características, que confrontam as idéias utilizadas no desenvolvimento destes trabalhos com as idéias utilizadas no desenvolvimento do SWS-GIS. No final apresentamos uma conclusão do levantamento realizado e resumimos numa tabela, as características observadas de cada trabalho.

### 3.1 Descoberta de serviços em uma federação de registros (Sivashanmugam et.al)

Conforme visto no capítulo 2, catálogos UDDI não dão suporte necessário para descoberta de recursos distribuídos, Sivashanmugam et. al [SVS04] sugere uma rede *peer-to-peer* de catálogos UDDI públicos e semi-privados, que permite acesso transparente a qualquer registro da federação. São usadas ontologias para classificar registros e localizar recursos. As empresas e organizações têm necessidade de distribuir seus recursos (serviços e dados) por diversos motivos: localização geográfica, natureza dos serviços, funcionalidade de negócio e especificações técnicas. Certamente os serviços devem ser agrupados em conjuntos de registros de forma hierárquica ou não. O mais importante é que deve existir uma maneira de recuperar os recursos distribuídos em vários registros de forma transparente para o usuário.

Numa federação de registros todos cooperam entre si, porém não perdem a autonomia. O objetivo principal de uma federação é formar uma comunidade que serve tanto para um domínio de negócio quanto para formar um "mercado" onde os serviços podem cooperar e concorrer entre si. Os membros da federação podem ser de naturezas (modelo de dados e API) diferentes e podem participar de uma ou mais federações. Podemos citar alguns benefícios desta técnica:

- Gerenciar recursos distribuídos sem concentrar todas as informações em um único lugar;
- Acessar e integrar várias informações sem acumular todas num repositório único manualmente. Ajudar a obter transparência entre os modelos de dados;

- Prover uma técnica “escalável” para acrescentar mais recursos ao sistema;
- A união agrega mais valores aos serviços (como um único ponto de autenticação/autorização, interfaces que especificam consultas e regras de integração de informação).

Os dados devem estar dispostos na federação de registros, desta forma, a replicação de dados em todos os registros não é necessária. Distribuir os dados usando as técnicas de “particionamento” de bancos de dados distribuídos (vertical e horizontal) também deve ser feito para obter melhores resultados. Os registros apresentam dois tipos de heterogeneidades: estrutural, que são diferenças entre modelos de dados; semântica, que são diferenças semânticas nos modelos de dados. Quanto à autonomia existem dois tipos básicos: autonomia de projeto, que inclui a seleção de diferentes tipos de acessos, diferentes algoritmos para publicação e descoberta; autonomia de execução, onde um registro deve ser capaz de dar suporte a publicação, e descoberta de *web services* independente de outros registros na federação.

### **3.2 Estratégia de mediação baseada em WFS para interoperabilidade em SIG (Boucelma et. al)**

Em Boucelma et. al [BEL02], a proposta é integrar dados e linguagem de consulta de fontes heterogêneas distribuídas em uma única visão, seguindo a arquitetura não-hierárquica de bancos de dados distribuídos, que é baseada em mediador e *wrappers* para acessar os dados distribuídos e fazer mapeamentos de consultas. Baseiam-se no padrão *global-as-view* (GAV) [CGM+94], onde os tradutores derivados - *derived wrappers* - permitem tanto atender consultas sobre os dados distribuídos, quanto consultar dados locais resultantes de operações sobre vários dados.

O sistema usa regras semânticas para fazer a integração de esquemas no mediador. As *Inter-schema Correspondence Assertions* (ICA) são regras baseadas em *XQuery* [Xqu05] que expressam os relacionamentos inter-esquemas. Operações são definidas baseadas nos esquemas mediados e nos tradutores derivados *derived wrappers*. A arquitetura do sistema proposto é baseada em um mediador, que é formado por um analisador de consultas, um “otimizador” e um módulo de execução. O mediador é responsável por analisar as consultas e aplicar as regras (ICA) para fazer as transformações devidas, realizar otimizações, e dividir a consulta em partes que serão despachadas para as respectivas fontes pelo módulo de execução. O formato de troca de informações entre os vários WFS e o mediador é realizada através de GML. O módulo

de otimização explora meta-informações dos esquemas das fontes, através da operação *getCapabilities* e define um plano de execução mais eficiente. Ao longo da execução da consulta, alguns operadores podem não ser atendidos pelas fontes WFS, para isso o sistema mantém um mecanismo para resolver as consultas que realiza os seguintes passos:

- Sub-consultas que correspondem diretamente às fontes de dados são diretamente submetidas para os seus correspondentes servidores WFS;
- Sub-consultas que possuem operadores que não são atendidas diretamente, serão resolvidas através de estruturas (*program wrappers*) que simulam fontes WFS com os resultados das operações;
- O *program manager* constrói um *program wrapper* para atender a operação requisitada, uma fonte de dados virtual é criada e populada com os resultados temporários da operação;
- Um *derived wrapper* é criado e posto na fonte de dados virtual recém criada para responder pelos resultados da operação, tornando a comunicação transparente para o mediador.

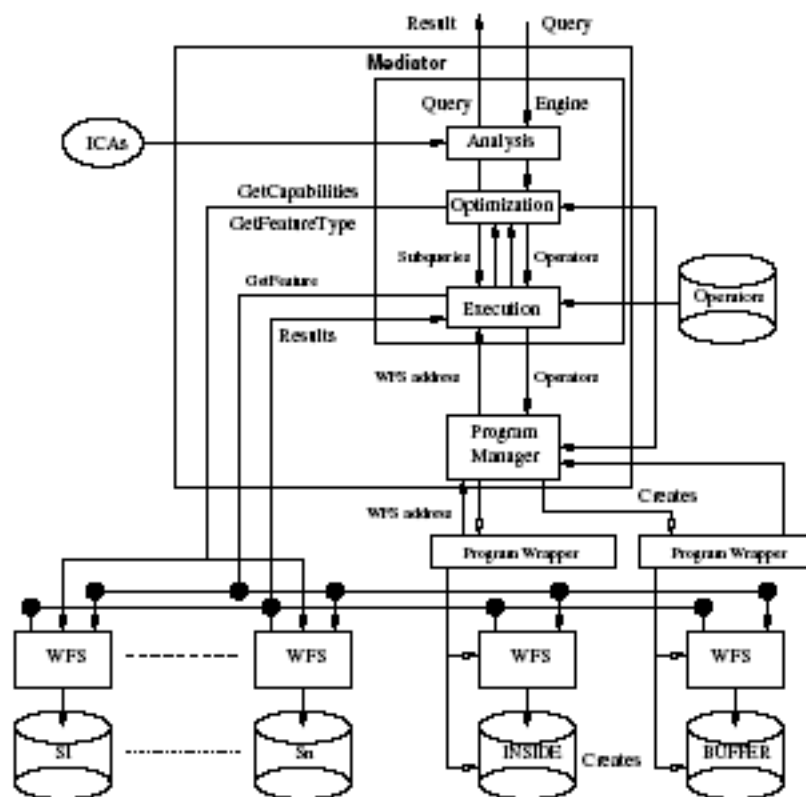


Figura 3.1: Mediador para várias bases WFS (Fonte: BEL02)

Na figura 3.1 apresentamos a arquitetura descrita. As principais vantagens desta arquitetura é o uso de linguagens padrões (XQuery, GML) que facilita a manutenção, trata de forma transparente fontes geradas (resultado de operações) ou puras (servidores WFS). Um esquema flexível devido ao uso de *wrappers*, possibilitando sempre o acréscimo de mais fontes de dados ao sistema. Uma desvantagem é que o sistema é baseado numa arquitetura bem centralizada, tornando o mediador um ponto de falha. Outro problema é que a cada nova fonte adicionada ao sistema, devem-se gerar as regras ICA para o esquema mediador, esse problema surge devido ao modelo *global-as-view* utilizado. Além disso, o trabalho não contempla a descoberta de novos recursos, deixando a cargo do administrador do sistema a inclusão de novas fontes de dados.

### **3.3 Busca baseada em ontologias para mapas digitais interativos (Hübner et. al)**

Hübner et. al [HSV+04] propõe unificar fontes de dados dispostas em vários lugares da Europa provendo um ponto único de interação com o cliente, através da ferramenta de definição dos mapas. O sistema trabalha com localização de recursos através de busca em metadados, os recursos são descritos usando uma linguagem e uma ferramenta (o Buster) realiza o tratamento semântico.

A arquitetura do sistema compreende serviços de armazenamento de dados, um catálogo, visualização de mapas (WMS) e WFS. O catálogo é responsável por recuperar informação através do casamento sintático de palavras-chave e/ou das coordenadas espaciais. Integrando o catálogo ao Buster, o serviço pode reter termos semânticos e relações espaciais em ontologias formalizadas e aplicar processos de inferência nas formalizações. O usuário especifica os termos da busca, que pode ter maior relevância semântica (sobre os termos definidos) ou maior relevância espacial (baseada no conceito de vizinhança). Desta forma, o sistema irá retornar os resultados baseados nas entradas do usuário e gerará um mapa interativo com os dados das diversas bases de dados, desde que opte por integrar o resultado da consulta com o WMS. O catálogo do sistema pode propagar suas consultas para outros catálogos OGC.

Todo processo de definição das ontologias e geração de mapas escolhendo as bases é centrado no usuário, o que torna o sistema mais complicado de usar e acarretando vários problemas discutidos no capítulo 2, pois o cliente deve ter profundo conhecimento dos serviços envolvidos. O ponto positivo da solução é que ele realmente faz buscas espaciais em catálogos distribuídos levando em consideração os dados espaciais dos registros e conceitos de proximidade, embora não utilize a dimensão

espacial das descrições dos serviços para melhorar a busca dos resultados. Além da característica espacial da busca o sistema realmente trata semanticamente as palavras-chave submetidas nas buscas, levando em consideração hierarquia e similaridades de conceitos.

### **3.4 Integrando descrições sintáticas e semânticas na formação de cadeias de serviços geográficos (Lemmens et. al)**

A formação de cadeias de serviços espaciais é uma tarefa não trivial devido à natureza peculiar dos dados espaciais, dentre as quais podemos citar a variedade de versões existentes, massivos *data sets*, dados que requerem mapas como interfaces obrigatórias e a própria natureza espacial dos dados.

Neste trabalho, Lemmens et. al [LWR+06] apresenta uma metodologia para descoberta, composição abstrata (identificação da funcionalidade das cadeias de serviços com a ajuda de parâmetros conceituais), composição concreta (gerenciamento do controle e do fluxo de dados nas cadeias formadas) e execução de cadeias de serviços espaciais. A técnica básica é utilizar descrições semânticas e sintáticas dos serviços espaciais para a formação das cadeias. Além disso, as cadeias formadas podem ser reutilizadas desde que sejam previamente anotadas, agilizando o processo de formação de cadeias.

A metodologia proposta para a criação das cadeias de serviço visa envolver diretamente o projetista de aplicações (*service designer*) no processo de formação de cadeias. Desta forma, com ajuda de uma ferramenta, o projetista de aplicações consegue determinar as funcionalidades dos serviços que estão envolvidos no processo de formação de cadeias. É baseada nas descrições sintáticas e semânticas que a ferramenta valida, auxilia e conclui o processo de formação das cadeias (*matchmaking*). Ao final do processo é gerado um arquivo com o resultado da cadeia abstrata obtida que, por sua vez, é “traduzido” para um arquivo BPEL responsável pela formação concreta da cadeia e execução.

Embora este processo leve em consideração as descrições semânticas dos serviços espaciais, a metodologia proposta não leva em consideração a natureza espacial das descrições dos serviços. Ou seja, não trata a dimensão espacial no processo de formação das cadeias de serviços. O catálogo espacial deve levar em consideração a abrangência dos serviços espaciais para que se obtenham informações do local em que se deseja visualizar no mapa. Ainda mais, a estratégia proposta não trabalha de forma

cooperativa com outros catálogos, ou seja, utiliza apenas um único catálogo para a localização de serviços e formação de cadeias. A proposta ainda considera uma única ontologia para descrever os serviços o que sujeita o trabalho a vários problemas como: a manutenção desta ontologia no catálogo; e a aceitação desta ontologia pela comunidade inteira dos SDIs.

### **3.5 Descoberta de recursos em uma infra-estrutura de spatial data infrastructure Europeu (Smits e Friis-Christensen)**

No cenário atual, onde os SDI estão proliferando e formando infra-estruturas que possibilitam um intercâmbio mais eficiente das informações espaciais, surge um interessante desafio no desenvolvimento de SDI, que é compartilhar recursos através de várias comunidades, onde cada uma possui seus idiomas, culturas e carga histórica diferentes. O trabalho de Smits e Friis-Christensen [SF07] investiga se é possível usar uma ontologia comum para a descoberta de recursos em um SDI multilíngüe.

A técnica utilizada é baseada em técnicas de recuperação da informação. Desta forma, o ponto chave é a implementação de uma ontologia multilíngüe que incorpora um vocabulário multilíngüe e uma coleção de descritores de recursos. A informação contida nas descrições é provida utilizando uma linguagem conhecida, e um *data set* (ou serviço) pode ter mais de uma descrição em várias linguagens. Desta forma, os catálogos espaciais usam a ontologia multilíngüe tanto para descrever seus recursos quanto no processo de recuperação da informação.

No trabalho não é tratado a formação de cadeias de serviços, apenas detalha a recuperação de recursos num SDI utilizando as ontologias descritas. Embora utilize uma ontologia para todo o SDI, não são tratadas consultas distribuídas em diversos catálogos no SDI ou em SDI diferentes, o que promoveria uma abrangência ainda maior do trabalho. Apesar de recuperar recursos através de descrições semânticas utilizando ontologias multilíngües, os autores não usam a dimensão espacial dos recursos descritos nos catálogos (ou catálogo) do SDI. O problema tratado pelos autores pode ser resolvido com um gazetteer, que poderia usar uma ontologia multilíngüe e poderia ser consultado pelo catálogo no processo de descrição e no processo de recuperação dos recursos existentes.

### 3.6 Descoberta de serviços baseada em ontologia em spatial data infrastucture (Lutz)

Atualmente, a grande maioria destas ferramentas de localização de serviços é baseada em buscas por palavras-chaves o que as tornam ineficientes considerando a precisão (*precision*) e o *recall* dos resultados retornados.

Para diminuir esses problemas o trabalho de Lutz [Lut05] apresenta uma metodologia para representar semanticamente as operações realizadas por *GIS services*. São usadas assinaturas semânticas que contêm conceitos representados em *Description Logic* para representar os inputs e os outputs das operações dos serviços e uma especificação das pré e pós condições em Lógica de primeira ordem. Desta forma, o processo de descoberta de serviços (*matchmaking*) é realizado em duas fases: a primeira, a assinatura semântica, é utilizada para filtrar a grande quantidade de serviços armazenada, usando a operação *subssumes* para retornar apenas os serviços que, semanticamente, estão nos níveis de detalhamentos inferiores; o segundo passo compara as pré e as pós condições dos serviços restantes com a consulta realizada.

A metodologia apresentada combina as técnicas de descrição de serviços e descoberta semi-automática para a composição de um catálogo de serviços de geoprocessamento. No processo de descrição, o usuário que está descrevendo o serviço (*advertisement*) deve escolher, de acordo com a ontologia única, o conceito que representa a operação que está sendo cadastrada, no passo seguinte, deve-se usar a ontologia de domínio para detalhar as entradas e as saídas desta operação. Durante o processo de *matchmaking* o usuário realiza as consultas por serviços espaciais no catálogo, em seguida o catálogo transforma a consulta em uma descrição de serviço e executa o *matchmaking* (filtragem e processamento das pré e pós condições) com o objetivo de casar a requisição com os serviços existentes.

Esta metodologia leva apenas em consideração as entradas e as saídas dos serviços em questão, todavia, a estratégia utilizada não leva em consideração a dimensão espacial dos serviços. Apesar de assumir que o SDI possui catálogos de serviços, o trabalho não leva em consideração a possibilidade de integrar SDIs através dos catálogos, o que implicaria uma discussão sobre a ontologia única que é assumida no trabalho. Utilizar uma única ontologia de domínio possui seus problemas como manutenção (mudanças na modelagem que ocorrem ao longo do tempo no SDI), imposição de um modelo a uma comunidade, entre outros problemas.

### 3.7 Uma arquitetura para troca e gerenciamento de informações geotécnicas distribuídas (Zimmerman et al)

Zimmerman et al [ZWW06]. apresenta o projeto Geotechnical Information Management and Exchange (GIME) que tem como objetivo melhorar a distribuição (e compartilhamento) dos dados espaciais entre bancos de dados heterogêneos sob diferentes controles administrativos. O projeto possui algumas características como: autonomia de cada organização sobre seus dados; acesso padronizado através de *web services*, com o objetivo de esconder detalhes de acesso aos dados distribuídos; processamento de consultas eficiente e cooperativo.

A arquitetura do GIME compreende dois tipos de arquivos: *read-only (RO)* e *read-write (RW)*. Os arquivos RW são compatíveis com três tipos de serviços: *web services* compatíveis com o armazenamento (FSWS), consultar e recuperar (*Query exchange web service; QEWS*) e visualizar (VWS) informações geotécnicas. Arquivos RO são compatíveis, apenas, com o serviço de QEWS. É utilizando estes serviços que um administrador de um banco de dados local publica os dados a serem compartilhados.

O processamento de consultas espaciais através da arquitetura é baseado numa estratégia eficiente de “roteamento” das consultas através dos sites. O fundamento básico da estratégia projetada é o repasse das consultas entre os bancos de dados envolvidos. Por exemplo, se um site (ou banco de dados) não possui a informação requerida, o algoritmo é responsável por encaminhar a consulta para outros sites da rede. Esta estratégia é sujeita a um problema comum em sistemas distribuídos que é a quantidade excessiva de mensagens na rede causando uma sobrecarga da estrutura. Este problema é chamado de *flooding*. Para evitar tal problema, a equipe desenvolveu uma estratégia de indexação distribuída dos dados dos bancos de dados. A indexação é baseada e nas estruturas de R-tree e Quadtree, desta forma, é inserido o MBR de cada arquivo de conjunto de dados numa R-tree ou Quadtree global. Portanto, durante o processamento de uma consulta, cada arquivo faz interseção com a área de consulta (retângulo de consulta) com o MBR armazenado. A consulta é repassada apenas para os candidatos cujos MBRs sobreponham a área de consulta, reduzindo o tráfego de mensagens na rede.

A estrutura de índices é mantida por arquivos que armazenam vários *data sets* que se enquadram (especialmente) numa determinada área, ou seja, cada arquivo armazena os bancos de dados com informações espaciais que descrevem seus respectivos MBRs. A atualização destes índices só acontecerá caso um dos quatro



arquivos que definem a área de abrangência de cada arquivo forem alterados ou removidos, que é um evento raro de acontecer. Desta forma, a atualização dos índices espaciais não é muito freqüente, o que viabiliza a manutenção destes índices em detrimento da grande diminuição do tráfego de mensagens na rede.

Apesar da eficiente técnica de recuperação de dados espaciais em bancos de dados distribuídos, o trabalho visa, apenas, a recuperação de dados distribuídos. Não é foco do trabalho a recuperação de serviços espaciais genéricos, nem a formação e localização de cadeias de serviços, que é algo mais abrangente. O trabalho não trata semântica dos dados da rede, ou seja, assume que todos os dados da arquitetura seguem o mesmo padrão.

### **3.8 Uma técnica para descoberta orientada a serviços e recuperação de dados espaciais (Ghosh e Paul)**

Embora muitos padrões de serviços espaciais tenham sido definidos pela OGC, ainda existe uma insuficiência de soluções no tocante ao tratamento semântico dos serviços espaciais. Para isto, o trabalho de Ghosh e Paul [PG06] visa integrar diversos repositórios espaciais através da arquitetura SOA, utilizando descrições semânticas de serviços.

A estratégia elaborada pelos autores tem o objetivo de melhorar a descoberta e a recuperação de GIS services em catálogos espaciais. A idéia é que os usuários busquem serviços no catálogo espacial central, recuperem os serviços desejados e utilizem os serviços através de um módulo, que resolve as diferenças semânticas entre as fontes de dados. A estratégia implementada neste trabalho, é semelhante à idéia de GAV (*Global-as-view*), ou seja, os usuários possuem a visão de esquema global (catálogo central), onde tal esquema, que é representado por uma única ontologia formada a partir de um vocabulário comum entre as fontes de dados, descreve os serviços e é utilizado no processo de recuperação da informação.

O módulo de consulta do protótipo, chamado de Query Processor, recebe as consultas dos usuários, divide a consulta inicial em sub-consultas para executar cada uma na respectiva fonte de dados (serviços WFS ou WMS). Em seguida, os serviços invocados retornam os resultados das requisições para o módulo de consulta, finalizando o processo. A comunicação entre os serviços WFS e WMS é realizada através de wrappers que traduzem as consultas e aplicam os filtros (no caso de WFS) e tratam as diferenças sintáticas entre as fontes de dados.

A solução proposta por Gosh e Paul é interessante, pois leva em consideração uma ontologia para descrever os serviços nos catálogos. Porém, a estratégia aborda apenas a localização de serviços considerando apenas um catálogo centralizado o que limita bastante o poder de difusão da solução. Outro ponto a se observar é que embora se utilize ontologia para descrever os recursos do catálogo, a solução não leva em consideração a dimensão espacial das descrições de WFS e WMS contidas no catálogo. Numa estratégia baseada na técnica Global as view, é evidente observar que existe uma limitação na flexibilidade que os wrappers (tradutores) impõem sobre a solução, pois a cada novo serviço que surge é preciso de um novo tradutor para tratar a comunicação entre o módulo de consulta e as fontes de dados.

### **3.9 Uma técnica peer-to-peer para descoberta de GIS services (Xiujun et. al)**

Um dos principais desafios na descoberta de recursos espaciais em SDI é tratar a disposição das informações e dos serviços na rede. Ou seja, é localizar os recursos descritos nos catálogos de forma eficiente. Para tanto, é necessário tratar com as diferenças semânticas entre os catálogos, bem como elaborar uma estratégia de indexação da informação distribuída eficiente. Xiujun et. al [XGK+06] propõe uma técnica baseada em P2P para descoberta de serviços espaciais distribuídos.

Inicialmente, cada máquina da rede extrai as meta-informações dos serviços de dados padronizados pela OGC (WFS, WMS e WCS). Assim, informações sobre parâmetros de qualidade de serviços e a área de abrangência de cada serviço (*bounding box*) é utilizado para a elaborar as descrições dos serviços de cada catálogo. Em seguida, cada catálogo monta a estrutura que irá indexar todas as descrições obtidas. Em cada catálogo, o processo de indexação utiliza *R+ tree*, que usa o *bounding box* das descrições para montar a estrutura de dados em árvore. Para montar a *R+ tree* distribuída, é preciso dividir o espaço em partições, onde cada partição terá uma sub-árvore que indexa todos os recursos que estão dentro da determinada área. Os nodos folha das sub-árvores são armazenados em cada catálogo, assim, cada um possui um identificador e o *bounding box* da área que é responsável (é importante ressaltar que cada catálogo, possui uma estrutura de indexação baseada em *R+ tree* para indexar seus recursos próprios).

Desta forma, quando um usuário realiza uma consulta para um nodo da rede, este verifica na sua estrutura local se pode atender a requisição, caso não possua

serviços que contemple a consulta, o nodo é responsável por propagar a consulta para os outros nodos, utilizando-se da estrutura R+ *tree* distribuída.

A solução proposta pelos autores é eficiente na indexação e descoberta de recursos espaciais em catálogos P2P. A grande contribuição está na construção de uma estrutura de indexação espacial que leva em consideração a dimensão espacial das descrições dos serviços. Assim, a estratégia otimiza o espaço de busca a cada catálogo percorrido no processo de propagação das consultas. Porém, o trabalho não contempla o tratamento semântico das consultas e conceitos utilizados nas descrições e consultas, embora nas descrições contenham parâmetros de qualidade de serviços, não é mencionado como os catálogos formalizam estas definições. O que nos leva a crer que seja obrigado a utilizar um conjunto de conceitos sintáticos e semânticos em toda a rede P2P de catálogos distribuídos. Outro ponto não contemplado no trabalho é o processo de execução dos serviços encontrados. Não fica claro como são resolvidas as questões de heterogeneidades semânticas entre as fontes de dados e nem como são utilizadas pelo cliente.

### **3.8 Considerações finais**

A descoberta e a composição de serviços (GIS services) proporcionaram um grande avanço para as arquiteturas baseadas em SDI. A descoberta de serviços é um dos grandes desafios que precisam ser resolvidos para que essas infra-estruturas atinjam seus objetivos. São previstos pela arquitetura de SDI, serviços de catálogos para descobrir serviços e dados apropriados para uma tarefa específica.

Após a análise de vários trabalhos na área de localização de recursos espaciais em catálogos distribuídos, é possível verificar que são poucos os trabalhos que abordam o problema de localização de serviços de geoprocessamento distribuídos. Da mesma forma, podemos constatar que muitos dos trabalhos analisados apresentam soluções que ainda não exploraram, de forma satisfatória, todos os potenciais que estas tecnologias provêm, nem atendem completamente aos requisitos dos SDIs orientados a serviços que serão listados na tabela 3.1. Conforme visto, nenhuma dessas abordagens é adequada para os requisitos de buscas em serviços em um ambiente SDI, dando margem à necessidade de soluções mais centradas em modelagens espaciais para descoberta de recursos em catálogos. Além destas deficiências, a maioria das ferramentas de localização de recursos, que estão sendo desenvolvidas para os SDI, apresentados no capítulo 2, não atende aos requisitos das tendências dos SDI. Pois, tais soluções são

baseadas em portais Web e a interação com o usuário carece de um navegador, tornando o trabalho de seleção de recursos quase impossível de ser automatizado. Certamente, os primeiros passos foram dados os trabalhos listados, utilização de ontologias para modelar recursos em catálogos distribuídos supre a carência semântica das especificações de catálogos descritas na seção 2. A Tabela 3.1 apresentamos uma visão resumindo os trabalhos até então discutidos:

<b>Trabalhos analisados</b>	<b>Características</b>				
	<i>Utiliza federação de catálogos espaciais</i>	<i>Utiliza buscas espaciais em catálogos</i>	<i>Padrão OGC</i>	<i>Forma de interação da cadeia de serviços (Client, Static, Wrokflow)</i>	<i>Trata processamento de consultas distribuídas</i>
<i>Lemmens et. al</i>	Não	Não	Não	Client	Não
<i>Hubner et. al</i>	Sim**	Sim	Parcial	Client	Não
<i>Boucelma et. al</i>	Não	Não	Sim	Static	Sim
<i>Sivashanmugam et.al</i>	Não	Não	Não	Client	Não
<i>Friis-Christensen</i>	Não	Não	Sim	Não trata	Não
<i>Lutz</i>	Não	Não	Não	Não trata	Não
<i>Zimmerman et al</i>	Não	Sim	Não	Não trata	Sim
<i>Ghosh e Paul</i>	Não	Sim	Sim	Não	não
<i>Xiujun, M et al</i>	Sim	Sim	Não	Não trata	Parcial *

**Tabela 3.1: Trabalhos relacionados**

\*Trata a parte da recuperação, porém não trata a parte semântica das consultas

\*\* Não considera a dimensão espacial quando da propagação das consultas para os outros catálogos

## Capítulo 4 . A arquitetura do SWS-GIS

---

Este capítulo discorre sobre a arquitetura proposta pelo SWS-GIS e como é promovida a integração entre os SDIs através dos catálogos. É explanado também, neste capítulo, como está implementada a modelagem dos recursos nos catálogos e como acontecem as consultas locais e a propagação destas pela rede de catálogos distribuídos. A arquitetura proposta no trabalho é composta pelo catálogo e os serviços que são providos pelo SDI que estão publicados no seu respectivo catálogo. Inicialmente, apresentaremos o resultado do levantamento de requisitos realizado, em seguida, os detalhes de estrutura.

### 4.1 Requisitos funcionais

O principal objetivo deste trabalho é construir uma ferramenta que integre diversos SDIs através de seus catálogos. Tal ferramenta deve levar em consideração a dimensão espacial dos serviços que nela estão descritos. Os principais requisitos funcionais levantados foram:

#### *Registro de serviços espaciais (GIS services)*

Os catálogos dos SDIs devem ser capazes de prover uma ferramenta para que os provedores de serviços espaciais possam publicar seus serviços. Cada catálogo deve descrever os serviços de acordo com o domínio adotado no SDI. O provedor do serviço deve acessar um sistema Web, onde fornece as principais informações dos seus serviços como: URI, descrição das entradas dos serviços, descrição das saídas, área de abrangência do serviço (*bounding box*), tipo de serviço (serviços de processamento, de dados) entre outras.

#### *Gerenciamento da modelagem dos recursos do catálogo*

As ontologias de domínio devem ser utilizadas para compartilhar as informações entre os catálogos. Portanto, para que as consultas possam ser propagadas entre os catálogos, cada catálogo deve modelar os seus recursos de acordo com uma ontologia que deve estar inserida numa hierarquia de domínios, conforme foram levantados os requisitos do

sistema. Estabelecemos uma hierarquia que descreve os recursos em diversos níveis, por exemplo, nível de órgão, secretária ou organização, nível estadual, nacional e assim por diante. Desta forma, cada catálogo deve conter, além de uma ontologia que modela os seus recursos, arquivos de mapeamentos, que ligam os conceitos de cada entidade para o nível, imediatamente, mais elevado na hierarquia.

#### *Descoberta de serviços*

No processo de formação de uma aplicação, o *service designer* requisita os serviços desejados ao catálogo através de uma interface bem definida e que segue o padrão OGC CSW. Ou seja, o catálogo deve prover o método *query* que recebe um OGC Filter (que é um padrão de linguagem de consultas espaciais adotado pela OGC) contendo a requisição dos serviços necessários. Para tanto, o programa cliente (utilizado pelo *service designer*) deve estar ciente do modelo utilizado no catálogo (que é o do SDI em que se encontra). Ao receber a consulta, o catálogo deve verificar na base de descrições local se os serviços publicados que possui atendem à requisição do cliente.

#### *Descoberta distribuída de composições e de serviços*

Quando o usuário efetuar uma requisição, o catálogo deve ser capaz de, primeiramente, verificar se um único serviço é capaz de atender a requisição. Caso o repositório de descrições local não contenha tal serviço, o catálogo deve ser capaz de tentar montar uma cadeia (composição) de serviços que satisfaça a requisição ou, pelo menos, que satisfaça parcialmente a mesma.

No caso em que o catálogo não consiga montar uma cadeia de serviços que satisfaça completamente a requisição do cliente, ele deve consultar os outros catálogos da rede em que se encontra em busca, apenas, dos serviços que faltam para atender completamente a consulta e não existem no repositório de descrições local.

#### *Execução das composições e serviços*

Ao final do processo de formação de aplicações, o usuário pode receber serviços ou uma cadeia (composição) de serviços. O próximo passo é executar o serviço ou cadeia. Para essa tarefa, o sistema deve prover um serviço que executa e gerencia as cadeias de

serviços selecionadas pelo usuário. Esse serviço deve tratar erros entre serviços, exceções e atrasos. O modelo de formação de cadeias de serviços deve ser o *Workflow-based service*, como foi descrito na seção 2.

#### *Resolução dos nomes de lugares através de um serviço de gazetteer*

O usuário pode optar por consultar o catálogo passando o nome da localidade como parâmetro. Para resolver o nome do lugar passado pelo usuário, o catálogo deve consultar um serviço de *gazetteer* para resolver o nome. Desta forma, o catálogo deve obter a feição do lugar passado pelo usuário e utiliza-la para que se possa consultar a base de descrições.

## **4.2 Requisitos não-funcionais**

Os requisitos não funcionais levantados foram:

#### *Arquitetura orientada a serviço (SOA)*

Na seção 2 vimos que *web services* são aplicações identificadas por uma URI, cujas interfaces e associações devem estar bem definidas, descritas e descobertas com artefatos em XML. Portanto, o paradigma orientado a serviços torna-se requisito indispensável na implementação deste sistema, pois garante maior flexibilidade entre as organizações, autonomia e integração. Como o sistema trata de organizações diferentes (no caso, órgãos do governo) nem sempre teremos o mesmo sistema, assim a tecnologia *web services* surge como a forma ideal para integrar sistemas tão diferentes, já que garantem a separação das aplicações (“encapsulamento” através dos serviços) e integração através de um padrão de troca de dados formatados.

#### *Tempo hábil de resposta*

O catálogo deve manter um tempo tolerável para retornar os resultados para o usuário. Todavia, como se trata de uma aplicação meio, ou seja, a aplicação não é voltada para o usuário final, o tempo deve ser mais tolerável que o tempo de resposta de uma aplicação comum.

## *Uso de Padrões*

É um requisito importante para manter a compatibilidade com toda a arquitetura proposta pela OGC, desta forma, seguimos o padrão OWS, que define todo o conjunto de *web services* geográficos da OGC. Nos clientes que acessam os serviços nós primamos pelo padrão *Model-View-Controller* (MVC) de desenvolvimento de aplicações.

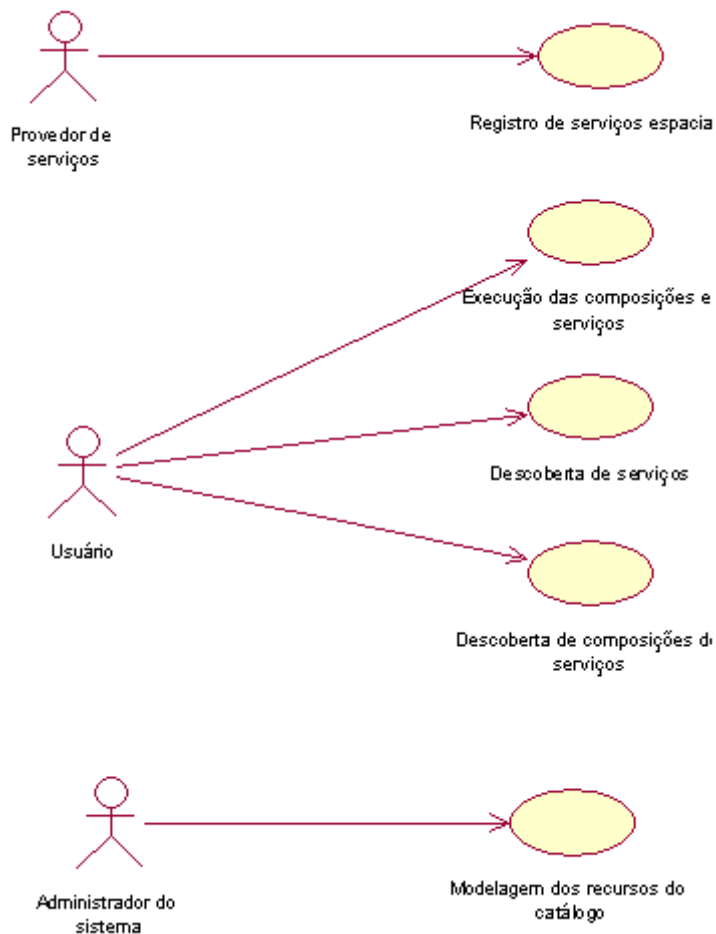
### **4.3 Atores do sistema**

Após a identificação dos requisitos da ferramenta e das suas características, realizou-se um levantamento dos atores e participantes do sistema, ou seja, a identificação de todos os componentes e atores que interagem diretamente com o mesmo. De acordo com os seus papéis, os atores identificados foram:

- Administrador do sistema: este ator usa o sistema para acrescentar, alterar ou remover alguma ontologia de domínio, é responsável pela inserção inicial do catálogo na rede que interliga os SDI;
- Provedor de serviços: registra as informações e os serviços que ele oferece;
- Usuário ou programa cliente: Desempenha o papel do cliente final. A pessoa ou componente de software que o service designer entrega a aplicação final resultante das consultas ao catálogo.

Na figura 4.1 segue um diagrama de *use cases* que ilustra a interação com os requisitos definidos no sistema.





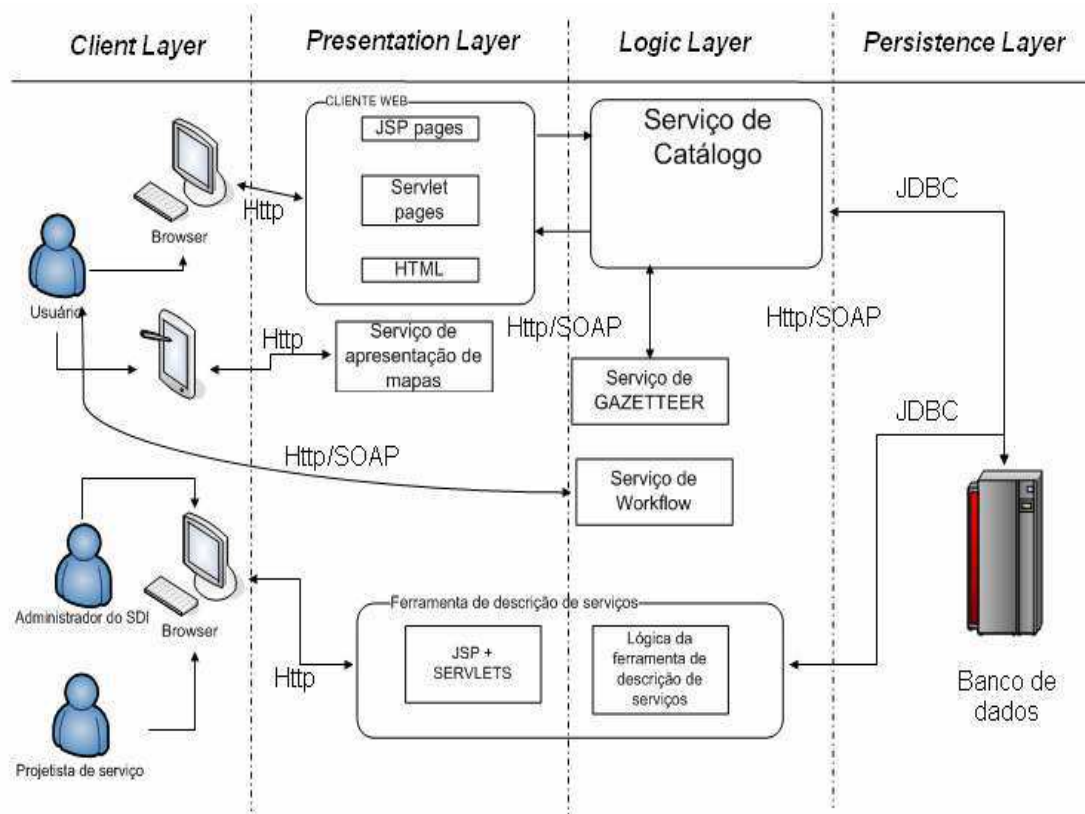
**Figura 4.1: Diagrama de use cases**

#### 4.4 Arquitetura do SWS-GIS

A arquitetura do SWS-GIS é baseada em serviços e utiliza o padrão “*find-bind-execute*”, no qual o usuário localiza serviços e cadeias e as executa com o objetivo de obter as informações do serviço espaciais. Como a arquitetura está implementada utilizando *web services*, é possível ter vários tipos de cliente (outros programas, inclusive). Os serviços podem ser descobertos e são independentes de plataforma (clientes escritos em várias linguagens (Java, ou .NET) podem acessar os serviços da arquitetura) e os serviços podem ser compostos para formar uma solução mais completa para o usuário.

Na figura 4.2 segue o projeto arquitetural do sistema desenvolvido. Esta figura ilustra a interação entre os serviços, usuários e programas clientes, no nível de protocolo de comunicação entre os módulos. Observe que o usuário interage com o sistema de três formas: a primeira, quando está buscando serviços ou composições; no segundo momento, quando envia uma cadeia de serviços para que o serviço de *workflow* execute

a cadeia; e a terceira forma ocorre quando o usuário recebe o resultado do processamento final sob forma de um mapa (saída do serviço de apresentação de mapas) ou a saída textual. É importante notar como o administrador do sistema e o provedor de serviço.



**Figura 4.2: Projeto arquitetural**

Ainda na figura 4.2, observamos o módulo de descrição interagindo com o provedor de serviço (através da ferramenta de descrição de serviços) e com o banco de dados, onde são armazenadas as descrições dos serviços. É possível observar também, a interação entre o serviço de catálogo com o banco de descrições, para recuperar as cadeias de serviços, com a aplicação Web, que serve como cliente para o catálogo e com o serviço de Gazetteer, onde são recuperadas as feições geográficas dos lugares pesquisados pelo cliente.

Em um SDI orientado a serviços, o SWS-GIS permite buscas espaciais pelos serviços do SDI e de outros catálogos que pertencem a outros SDI que formam uma federação de serviços espaciais. Podemos citar as seguintes características desta arquitetura:

- **Capacidade de localização de serviços através de consultas espaciais:** É possível localizar serviços utilizando a dimensão espacial das descrições de

serviços. Desta maneira, operadores espaciais como buffer, contains, inside, overlaps, disjoint e etc. podem ser usados para formar as consultas.

- **Capacidade de localização de recursos em catálogos federados:** através das ontologias de cada catálogo é possível modelar os recursos dos catálogos. Contudo, cada catálogo deve possuir arquivos de mapeamento entre a ontologia do SDI a que ele pertence e a ontologia diretamente mais global. Uma vez que o catálogo decide propagar a consulta para os outros catálogos, ele utiliza estes mapeamentos para a transformação dos conceitos utilizados na consulta a ser propagada para conceitos pertencentes à ontologia diretamente mais global. Desta maneira, os catálogos que recebem a consulta propagada podem encontrar a ontologia utilizada na consulta e mapear seus conceitos para a ontologia de seus respectivos SDIs;
- **Protocolo de otimização das consultas distribuídas:** o protocolo de consulta estabelecido nos catálogos dos SDI utiliza a dimensão espacial das descrições de serviços armazenadas em cada catálogo. Desta forma, o protocolo sempre aperfeiçoa a consulta inicial a cada propagação, descartando da consulta as informações que já foram encontradas e buscando sempre as informações que ainda não foram obtidas.
- **Execução flexível das cadeias de serviços:** Nesta arquitetura, o módulo de execução de serviço é separado do catálogo, assim fica a critério do cliente optar por um serviço de execução de cadeias de serviço que mais se adapte as suas necessidades.
- **Transparência de domínios:** Para o cliente da cadeia de serviços formada, não interessa saber a qual SDI os serviços que ele utiliza pertencem.

#### 4.5 Serviços da arquitetura

Os SDIs cooperam através da federação de catálogos. Desta forma, é possível, para qualquer cliente de qualquer SDI, obter recursos (serviços) providos pelos SDIs da federação. Na arquitetura desenvolvida, o cliente pode consultar o serviço de catálogo, que, por sua vez, propaga a consulta para os outros SDIs e o catálogo é responsável por acumular os resultados obtidos, compor as cadeias e retorná-las para o cliente. A figura 4.3 ilustra a arquitetura da federação.

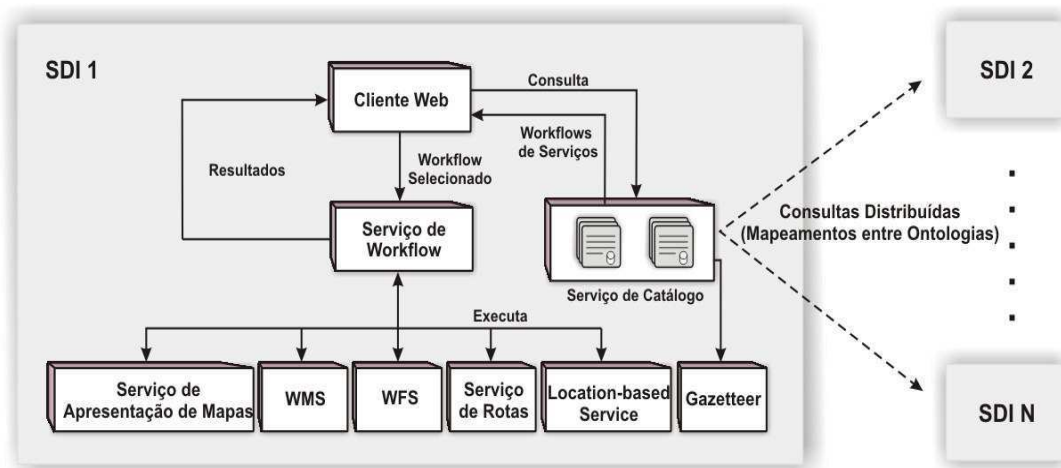


Figura 4.3: Arquitetura da federação de SDI

Na figura 4.3, cada SDI contém um conjunto de serviços espaciais que são providos por organizações públicas ou privadas. O catálogo representa um OGC CWS (*catalog web service*), que possui uma base de descrições dos diversos serviços que podem conter no SDI. Nesta seção, apresentamos uma descrição detalhada de todos os serviços que podem conter na arquitetura.

#### 4.5.1 Serviços de dados (WFS e WMS)

O WFS e o WMS representam *OGC Web Feature Service* e *Web Map Service*, respectivamente [OGC06]. Tais serviços, estão implementados usando as tecnologias de *web services*. Atualmente, foram implementadas as seguintes operações para o WMS: *getMap* e *getFeatureInfo*. Já o WFS, como foi descrito na seção 2, a especificação permite que o usuário consulte e atualize os dados espaciais de forma “interoperável”, as operações implementadas para este serviço foram: *getCapabilities*, *describeFeatureType* e *getFeature*.

#### 4.5.2 Serviço de apresentação de mapas (Basemap service)

Um serviço de apresentação de mapas foi desenvolvido para se integrar na arquitetura desenvolvida. Este serviço tem a função de apresentar dados espaciais de forma gráfica para o usuário final. Os mapas são obtidos a partir da execução do serviço de *workflow* que retorna o resultado final dos serviços espaciais executados (detalhado na seção 4.5.5).

#### 4.5.2 Serviço de LBS (*Location Based service*)

Este serviço é para aplicações móveis cientes de contexto que foi desenvolvido num outro trabalho de dissertação [Alm06]. Neste serviço, os usuários podem se

registrar e recebem alertas de interesse durante o seu deslocamento (viagens, caminho para o trabalho, etc.). O serviço de LBS é responsável por manter e gerenciar a informação de contexto do usuário. Quando um cliente está registrado e provê seu perfil ao sistema, o cliente pode receber informações do contexto. As principais operações disponíveis são:

- `registerUser()`: que registra um usuário em um *web service*, seu perfil e suas informações sobre o contexto;
- `updatePosition()`: que permite atualizações sobre a posição geográfica dos usuários no sistema e analisa o contexto em busca de ações que são relevantes para o contexto do usuário. Cada vez que a localização do usuário é atualizada o contexto é analisado;
- `registerAppointment()` que registra uma indicação do usuário.

#### 4.5.3 Serviço de gazetteer

Um *gazetteer* espaço-temporal é usado para ajudar aos usuários a encontrar o *bounding box* ou as coordenadas espaciais a partir dos nomes dos lugares, e vice-versa. A forma mais simples de implementar um *gazetteer* é utilizando, apenas, o par <nome do lugar, característica espacial> de cada feição. Neste caso, o gazetteer não considera o tipo da feição de cada elemento. Os tipos das feições podem ser tanto implícitas quanto explícitas. No primeiro caso, é assumido que os usuários não sabem o tipo do lugar, por exemplo, o fato de saber que a Inglaterra é um país. No segundo caso, o tipo da feição vem junto com o nome do lugar, como, por exemplo, *North Sea* [PBM02]. Este serviço é acessado através de uma consulta pelos quatro atributos: nome da feição, tipo da feição, característica espacial da feição (coordenadas) e atributo temporal da feição. Por exemplo, um usuário pode estar interessado onde é um determinado lugar, ou o tipo de uma feição espacial que se encontra na superfície terrestre, inclusive, por sua evolução espaço-temporal (por exemplo, o estado de Goiás que sofreu uma mudança na sua feição espacial ao longo do tempo com a criação do estado do Tocantins).

#### 4.5.4 Serviço de busca de rotas

Este serviço é responsável por prover rotas entre dois ou mais lugares. O cliente passa como parâmetro da requisição (usando a operação *getRoute*) dois pares de coordenadas (latitude e longitude). O cliente pode também informar algumas preferências que são levadas em consideração pelo serviço, como estradas e condições do tráfego. Este serviço é parte de um outro trabalho de dissertação [Sil07].

#### 4.5.5 Serviço de workflow

O serviço de *workflow* é responsável por executar as cadeias de serviços que são formadas. O serviço de catálogo retorna um conjunto de cadeias de serviços (composições de serviços, ou *workflows*) que atendem à requisição do usuário. O cliente então seleciona a cadeia mais adequada dentre as retornadas. Em seguida, o serviço de catálogo repassa a referência da cadeia selecionada para o serviço de *workflow*, para que este execute a cadeia. O serviço de *workflow* executa a cadeia e monitora a execução, basicamente, possíveis erros, exceções, falhas e atrasos que venham a ocorrer durante as chamadas dos serviços. Ademais, para cada serviço envolvido na cadeia, o serviço de *workflow* verifica se as entradas e as saídas dos serviços são compatíveis.

Outro papel do serviço de *workflow* é organizar as saídas dos serviços WFS e WMS, de duas maneiras:

1. caso o usuário opte por ver o resultado final do processamento num serviço de apresentação de mapas, o serviço de *workflow* transforma a saída WFS, que são dados no formato GML, para o formato vetorial SVG, tornando assim possível a visualização dos dados resultantes do WFS serem apresentados no serviço de apresentação de mapas;
2. o serviço de *workflow* realiza a transformação semântica entre os serviços WFS retornados. Geralmente, os serviços WFS que pertencem a SDIs diferentes podem ser semanticamente heterogêneos. Para resolver este problema, o serviço de *workflow* utiliza a hierarquia semântica (que será explicada na seção 4.6.1) definida em cada SDI para traduzir os dados descritos por uma ontologia para serem descritos por outra. Para tanto, o serviço de *workflow* navega na hierarquia de ontologias em que os serviços WFS pertencem, buscando sempre a ontologia de mais alto nível de abstração em comum que descreve todos os serviços WFS, identificada a ontologia global comum a todos os serviços, o serviço de *workflow* transforma os dados de todos os serviços WFS envolvidos no resultado para uma única sintaxe.

#### 4.6 O serviço de catálogo

O serviço de catálogo é o núcleo da arquitetura desenvolvida. Ele é responsável pela integração entre os SDIs, seja em nível local, estadual, nacional ou internacional. Cada SDI mantém um catálogo espacial dos serviços providos pelo SDI local. Quando um catálogo não encontra os serviços solicitados em sua base de descrições local, ele

propaga a consulta (em busca dos recursos que não foram encontrados localmente) para os outros catálogos através do protocolo de comunicação desenvolvido.

No processo de consulta, o serviço de catálogo acumula os resultados (locais ou distribuídos) das consultas que contêm cadeias de serviços, retorna os resultados para o *service designer* e envia a cadeia de serviços selecionada para o serviço de *workflow* executar. Objetivando o cumprimento do padrão OGC, o serviço de catálogo implementa a especificação OGC CWS que contem os métodos: *getCapabilities*, *getRecords*, *getRecordById*, *describeRecord*, *getDomain*, *presentResults* e métodos de transação.

A próxima seção discute como os recursos nos catálogos estão modelados, em seguida é detalhada como ocorre a fase de descrição de serviços, os módulos do serviço de catálogo, incluindo de consulta, o módulo de reescrita de consultas, o módulo responsável por propagar as consultas e o módulo de casamento geográfico, uma seção explicando como é a interoperabilidade semântica entre os catálogos. Finalmente, apresentamos uma seção descrevendo o protocolo de recuperação da informação estabelecido entre os catálogos.

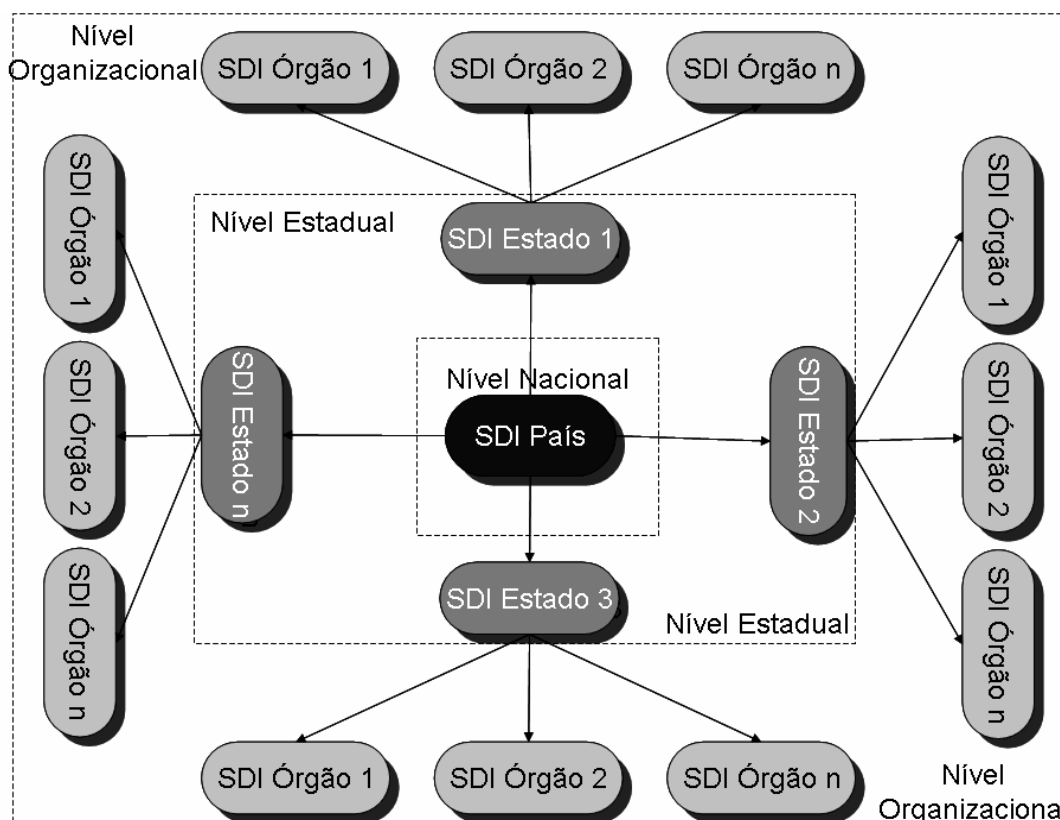
#### **4.6.1 Modelagem dos recursos nos catálogos do SDI**

A utilização de ontologias para que as organizações possam modelar e intercambiar conhecimento já está bem consolidado no cenário atual. Contudo, por várias razões, as pessoas e as organizações tendem a usar diferentes ontologias. A heterogeneidade semântica é um grande desafio que deve ser levado em consideração quando tratamos de integrar vários catálogos com suas diferentes ontologias. Portanto, no sistema foi adotada a estratégia de mapear os conceitos e propriedades entre as ontologias de cada SDI.

Cada SDI define sua própria ontologia que é usada no catálogo. Isto significa que existe um vocabulário comum que cada SDI representa os conceitos da ontologia de domínio utilizada para descrever os serviços. Contudo, é assumida a hipótese que deve existir um conjunto de conceitos que pode ser compartilhado entre os diversos SDIs, através de comunidades formadas por um *cluster* semântico. Tal hipótese foi assumida quando da fase de levantamento de requisitos, foi identificado esse comportamento no ambiente que focamos o problema, que são os órgãos públicos do Estado da Paraíba.

Desta maneira, cada comunidade possui sua própria ontologia de domínios que pode ser compartilhada, e outras comunidades podem compartilhar seus recursos

através dos mapeamentos entre as ontologias. Todas as comunidades reconciliam as ontologias através de vários possíveis softwares como o Protege [Pro07], alignapi [Ali07], OLA algorithm [OLA07], OWL ontology aligner [OOA07], dentre outras. Dentre tantas ferramentas, decidimos utilizar a ferramenta alignapi para gerar um alinhamento inicial e, em seguida, foi utilizado o Protégé para identificar pontos de diferenças semânticas entre os SDIs [HPS04]. Na figura 4.4 segue um exemplo de como podem ficar organizadas as comunidades semânticas. Esta hierarquia semântica é definida de acordo com metadados existentes nas próprias ontologias que indicam quais são as ontologias de mais alto nível para cada uma.



**Figura 4.4: Organização dos mapeamentos semânticos entre as comunidades formadas pelos SDI**

Considere os círculos mais externos como o modelo dos SDIs dos órgãos dentro dos Estados, os círculos intermediários como modelos entre os SDIs a nível estadual e o último à nível nacional. Na figura, é mostrada a possibilidade de haver mapeamentos diretos entre os níveis estaduais. Desta forma, se um recurso precisa ser localizado e este permanece em outro SDI (no órgão AESA procurar por rios de Pernambuco, por exemplo) o catálogo deve levar em consideração os mapeamentos semânticos entre os SDIs envolvidos, seja nível estadual seja em nível de organização.



## 4.6.2 Descrição dos módulos do catálogo

### *Repositório de descrições*

Todo catálogo da federação descreve os serviços de acordo com a ontologia do seu respectivo SDI, tal processo gera um arquivo de descrição do serviço. Estes arquivos estão implementados em OWL-S, ontologia utilizada para modelar *web services* (conforme visto do capítulo 2). Complementando as descrições dos serviços, as ontologias de domínio delimitam a semântica das entradas, saídas, atributos espaciais presentes nas descrições de serviços. Os arquivos são mantidos em um banco de dados espacial, onde são conservados junto com os dados espaciais que delimitam a área de abrangência (*bounding box*) dos serviços espaciais. O catálogo manipula as ontologias através do *framework* Jena [Jen07] em vários momentos: (a) para recuperar os conceitos que serão utilizados no processo da descrição dos serviços, (b) para utilizar os conceitos no momento da formação da consulta espacial, (c) para traduzir as consultas espaciais que serão propagadas para os outros catálogos e (d) para transformar os dados espaciais dos serviços de WFS que estão descritos utilizando ontologias diferentes.

A recuperação da informação é baseada nas descrições semânticas dos serviços de cada SDI. O sistema provê uma ferramenta, para ser utilizada pelo provedor de serviços ou o administrador, que gera as descrições dos serviços através das informações que são fornecidas e das ontologias do SDI. Desta forma, adicionar as descrições semânticas no catálogo segue os seguintes passos:

1. O provedor de serviço informa, primeiramente, a URL que contém o documento WSDL relacionado ao serviço. O sistema usa a API OWL-S para gerar o OWL-S [SiB04] a partir do documento WSDL.
2. Uma vez que o documento OWL-S é gerado, o sistema vai para o segundo passo do processo de descrição, no qual o provedor deve descrever (utilizando os conceitos da ontologia de domínio do SDI) as saídas e as entradas do serviço em questão. Por exemplo, ao descrever um serviço que possui dados sobre rios, açudes e poços, o provedor deve indicar na ontologia todos os conceitos de açude, poço e rios a que o serviço se refere. Além destas informações, devem ser providas o *bounding box* do serviço, a versão, o tipo de serviço (WFS, WMS, WCS, LBS, routing service ou serviço de transformação de coordenadas) e o sistema de coordenadas que se trabalha (se for um serviço de processamento,

como o serviço de transformação de coordenada, deve informar as coordenadas que recebem como entradas e as coordenadas que se obtêm na saída).

Na figura 4.5 é ilustrada a arquitetura do catálogo. Destacamos o processo de descrição de serviços na arquitetura. O provedor de serviços (*service providers*) acessa a ferramenta de descrição (*Service Description Tool*), a ferramenta usa a ontologia do SDI (*SDI Ontology*) para gerar as descrições dos serviços (*Semantic Web services descriptions OWL-S*) e armazenar no repositório de descrições (*Service Description Repository*).

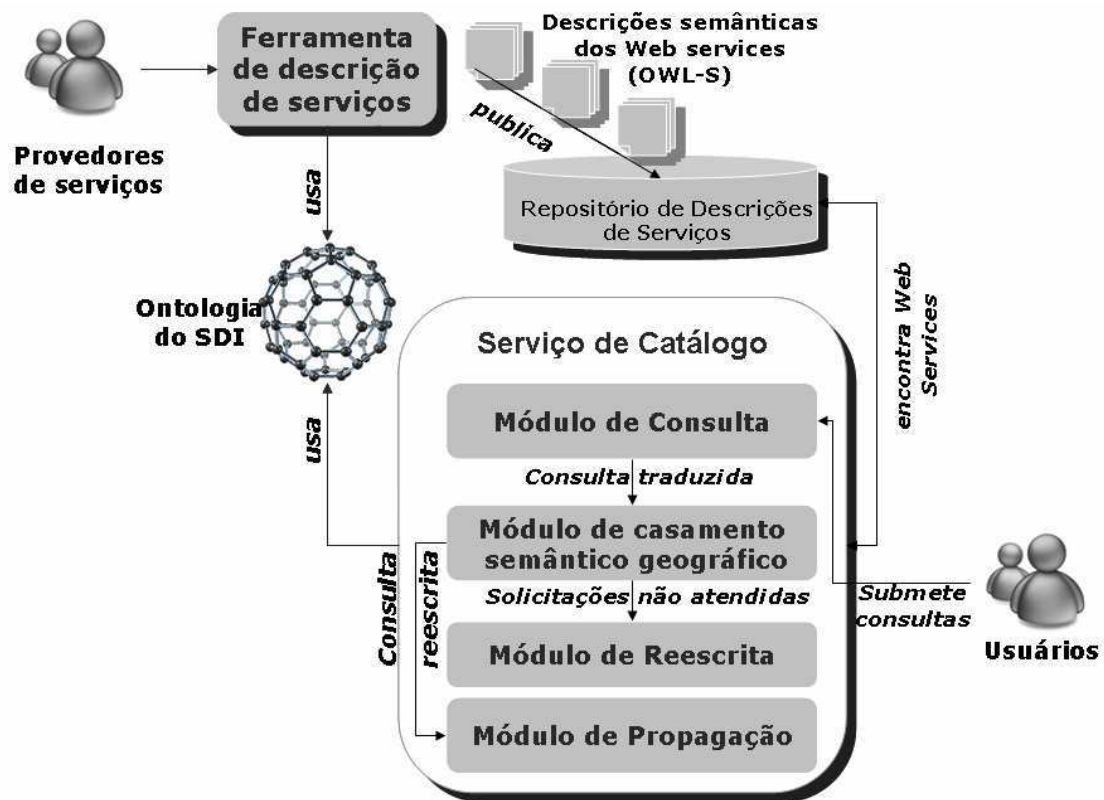


Figura 4.5: Arquitetura detalhada do serviço de catálogo. Na figura são apresentados seus módulos, a ferramenta de descrição, o relacionamento com a ontologia do SDI e a participação dos atores do sistema.

No trecho de código 4.1, segue um exemplo de um *service profile* (parte da descrição de um serviço usando OWL-S) gerado pela ferramenta de descrição. É importante notar que os parâmetros de entrada e saída (em negrito) apontam para a ontologia de domínio.

```
<profile:Profile rdf:about="#ResolvePlaceNameProfile">
  <profile:hasInput>
    <process:Input rdf:ID="placeName">
      <process:parameterType
rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">
http://buchada.dsc.ufcg.edu.br/~lsi/ontology/SDIConcepts.owl#placeName
```

```

</process:parameterType>
  <rdfs:label>placeName</rdfs:label>
</process:Input>
</profile:hasInput>
<service:presentedBy rdf:resource="#ResolvePlaceNameService"/>
</profile:hasOutput>
<process:Output rdf:ID="ResolvePlaceNameResult">
  <process:parameterType
rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">
http://buchada.dsc.ufcg.edu.br/~lsi/ontology/SDIConcepts.owl#place
  </process:parameterType>
  <rdfs:label>ResolvePlaceNameResult</rdfs:label>
</process:Output>
</profile:hasOutput>
</profile:Profile>

```

#### Trecho de código 4.1

No fim do processo de descrição, é gerado o arquivo OWL-S completo, que é armazenado no banco de dados. O módulo de consulta recupera esses arquivos no processo de descoberta de serviços.

#### 4.6.3 Módulo cliente de acesso ao catálogo

Durante a fase de construção da nova aplicação (através da composição dos serviços), o usuário consulta o catálogo para encontrar os serviços de geoprocessamento que possuem os dados (ou algum processamento) que lhe interessam. Primeiramente, o usuário acessa o catálogo através de um programa cliente que chama os métodos do *web service* de catálogo, através desta são passadas as consultas em OGC Filter. Durante este processo, o usuário especifica alguns detalhes da busca, tais como: nome das feições que deseja obter dados, tipo do serviço (WMS, WFS, serviço de rotas, etc.) que deseja obter, pode também passar as coordenadas da área de interesse e as operações espaciais. Esta aplicação cliente foi desenvolvida por uma equipe da disciplina de projeto e foi utilizada a tecnologia J2EE para desenvolvimento de aplicações Web e seguindo o padrão *Model-View-Controller* (MVC), onde temos as seguintes camadas:

- (a) A camada *view*: representada pelo *browser*, com os scripts que executam nele;
- (b) a camada *controller*: contem os JSP que definem o formulário de consulta e apresentam um mapa para que o usuário possa selecionar o *bounding box* de consulta., servets, a tecnologia google api e algumas classes de negócio que definem o fluxo do usuário no sistema;
- (c) a camada *model*: Esta camada submete as consultas para o catálogo através das chamadas via os *stubs* clientes do *web service* do catálogo.

Na figura 4.6 é ilustrado este primeiro processo de consulta pelo usuário.

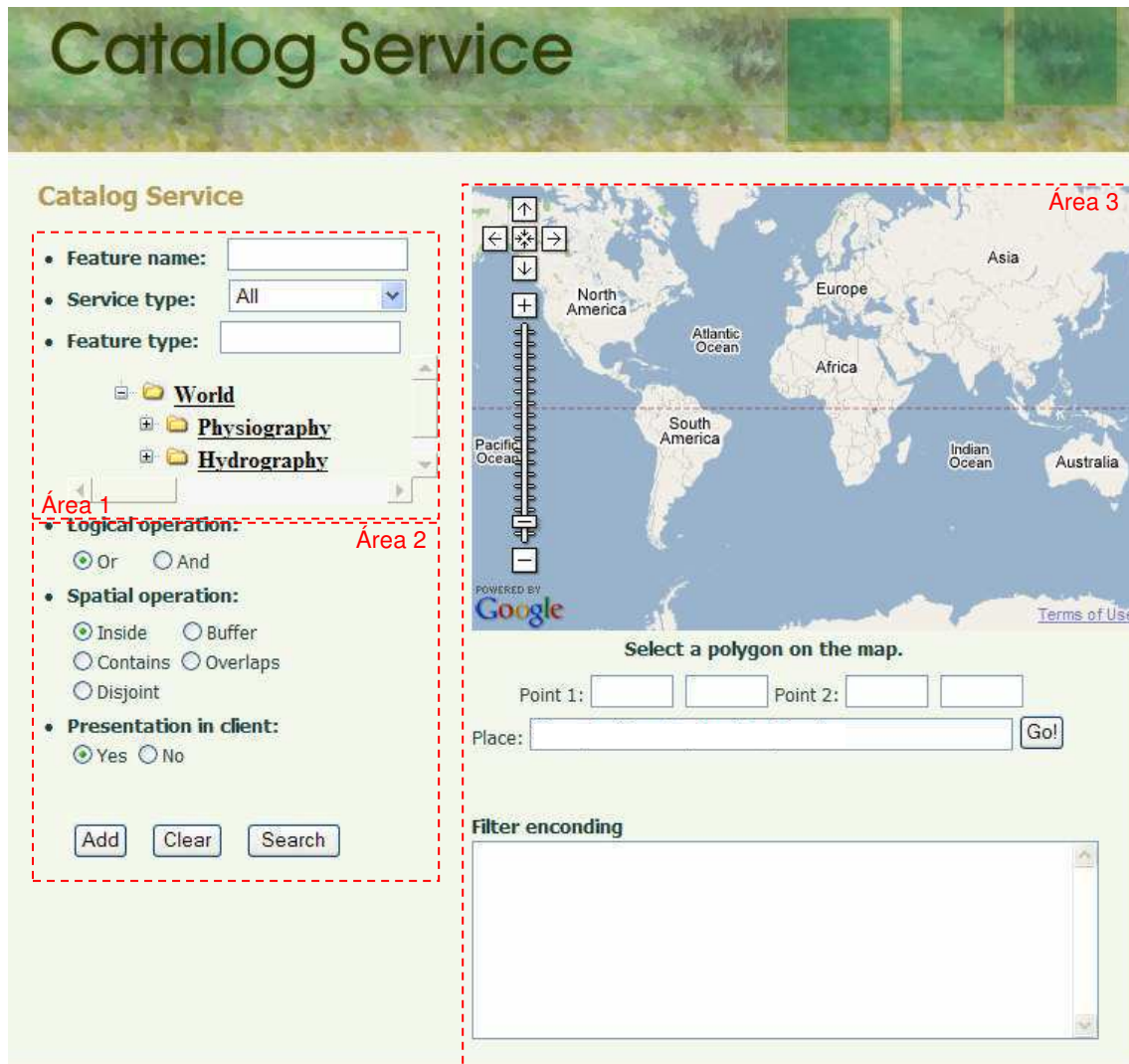


Figura 4.6: Tela de entrada para o módulo cliente de consulta ao catálogo.

Como é possível notar, a figura 4.6 apresenta um formulário Web dividido em 4 áreas. A primeira área define o campo chamado *Feature Name*, que serve para que o usuário informe a feição geográfica que se deseja obter dados, caso este campo não seja utilizado, o usuário pode utilizar a área 3 do formulário, que através dela, o mesmo pode especificar um *bounding box* de pesquisa. Ainda, na área 1, temos os campos *Service Type*, que indica o tipo de serviço que se deseja que retorne (WFS, WMS, *route service*) e o campo *Feature type*, que define o tipo da feição que o usuário deseja buscar os dados. O campo *Feature type* é especializado de acordo com a ontologia do SDI, desta forma, temos a feição *Hydrography*, que representa feições da hidrográfica e, por sua vez, possui três outras feições (River – rio; Dam – açude, Pole - poço) que especializam o escopo de *Hydrography*. Ainda neste campo, existe uma especialização das feições relativas à divisão política mundial (world – mundo, country – país, state – estado, city

– cidade, etc). Na área 2 temos os campos de operação espacial (buffer, inside, overlaps, contains, disjoint), o campo que solicita a apresentação dos dados num mapa (*Presentation in client*) e os botões para adicionar novas restrições de consultas (*add*), efetuar busca (*search*) e limpar as restrições de consultas (*clear*). Vale salientar que implementamos algumas operações espaciais, mas todas as outras estão disponíveis para o uso no catálogo como *web service*. Ainda, na área 2, estão as opções de adicionar o tipo das restrições de consultas se vai ser um “Or” ou um “And”. Todos esses campos são transformados em filtro OGC para consultar o catálogo espacial, que é exibido sempre na caixa de texto da área 3 da figura.

#### **4.6.4 Módulo de consulta**

Uma vez que a consulta foi submetida, o primeiro módulo do catálogo a receber a requisição é o módulo de consulta. Sua função é, primeiramente, consultar o um serviço de gazetteer para resolver os nomes dos lugares passados e obter a feição geográfica que será utilizada para consultar o catálogo (no caso da consulta pelo nome do lugar). Em seguida, este módulo traduz a consulta para ser submetida à base de descrições (na figura 4.5 o Módulo de casamento semântico geográfico), para recuperar as informações de acordo com as operações espaciais e as outras informações passadas. O processo de tradução das consultas de OGC Filter utiliza a hierarquia semântica das ontologias dos catálogos (explicada na seção 4.5) e compreende os seguintes passos:

1. O módulo verifica a ontologia da consulta realizada;
2. O módulo compara a ontologia utilizada na consulta com a ontologia do catálogo, caso sejam iguais ocorre o processo de tradução simples que é apenas mudar de OGC Filter para a linguagem da base de dados.
  - a. Caso as ontologias sejam diferentes, o módulo compara a ontologia da consulta que está diretamente um nível mais alto (mais abrangente) com a ontologia que está diretamente um nível mais alto (mais abrangente) que a ontologia do catálogo local;
3. Se as ontologias comparadas forem iguais, ocorre o processo de tradução dos conceitos utilizados para a ontologia que a consulta e o catálogo possuem em comum. Caso sejam diferentes o passo 2a deve ser repetido recursivamente até que se ache uma ontologia em comum entre os catálogos.

Por exemplo, uma agência da Paraíba solicitar rios (ou outro tipo de dado) do estado de Pernambuco. Nesta ocasião, ao catálogo que possui os dados de Pernambuco receber a

consulta, ele deve comparar com a ontologia local, verá que não é a mesma, então irá compara com a ontologia do estado, verá que não são do mesmo estado, em seguida, aumentando ainda mais o grau de abrangência, verá que possuem ontologias similares no terceiro nível, que é o nível de país. Assim, o módulo de consulta faz a tradução da consulta para o nível de secretaria para o nível ontológico estadual.

#### **4.6.5 Módulo de reescrita**

Este módulo é responsável por traduzir as consultas que serão submetidas para os outros catálogos, durante a fase de propagação de consultas. Durante a fase de tradução, o módulo recebe a consulta sobre os serviços que faltam à base local para atender a requisição por completo. Este módulo traduz essa consulta para o nível semântico diretamente superior (vide hierarquia semântica descrita na seção 4.5) ao do SDI local e repassa para os outros catálogos. Este processo será mais bem detalhado na seção 4.6.7.

#### **4.6.6 Módulo de propagação**

Este módulo recebe as consultas do módulo de reescrita que devem ser propagadas para os outros catálogos. Durante a fase de propagação, este módulo agrega os resultados retornados pelos outros catálogos consultados e repassa para o módulo de casamento semântico geográfico formar as cadeias com os novos resultados.

Este módulo também é responsável por manter a rede de catálogos, através do algoritmo descrito na seção 4.7.1

#### **4.6.7 Módulo de casamento semântico geográfico (MCSG)**

Este módulo recebe as consultas traduzidas do módulo de consulta do módulo de propagação para fazer o casamento semântico entre as informações solicitadas e as informações contidas no repositório de descrições de serviços. Para atender às solicitações, este módulo faz consultas espaciais no repositório de descrições usando as áreas de coberturas dos serviços cadastrados e as áreas de coberturas solicitadas. Ou seja, operações espaciais (*buffer*, *inside*, *overlaps*, *intersects*, *disjoint*, entre outras) sobre o *bounding box* das feições envolvidas.

Além da busca espacial, o MCSG filtra os serviços de acordo com outros tipos de parâmetros. Portanto, as requisições do usuário são comparadas com os serviços existentes a fim de verificar se os requisitos de qualidade de serviços encontrados atendem aos requisitos solicitados.

Além disso, no processo de formação das cadeias, este módulo leva em consideração as versões dos serviços retornados e o sistema de coordenada geográfica de cada serviço envolvido na cadeia. No processo de formação das cadeias de serviços, os serviços de dados retornados podem possuir diferentes sistemas de coordenadas, para resolver esse problema, o módulo identifica qual é o sistema de coordenada com maior número de serviços dentre os retornados e os ordena. Em seguida, ele adota o sistema de coordenadas mais numeroso na cadeia e para os outros serviços que possuem o sistema de coordenadas diferente ele tenta buscar serviços de transformação de coordenadas para uniformizar o sistema de coordenadas dos serviços de dados. Se não for encontrado um serviço de coordenada local, é disparada uma consulta para os outros catálogos em busca deste serviço de coordenada. Este passo é muito importante, principalmente, quando a cadeia possui um serviço de processamento e vários de dados como entrada deste serviço, por exemplo, um mapa com rios, rodovias e açudes, onde cada feição pode estar em um serviço diferente e o serviço de apresentação de mapas recebe como entrada as saídas dos serviços de dados que detentores desta informação.

Desta maneira, o módulo procura atender as consultas dos usuários de acordo com o repositório de descrições local, seja através de composições de serviços, seja através de serviços atômicos. Porém, se não conseguir serviços que atendam completamente às requisições, este módulo repassa as consultas para o módulo de propagação, que será mais bem detalhado na próxima seção.

#### **4.7 Protocolo de recuperação da informação em catálogos distribuídos**

No trecho de código 4.2 é apresentado o algoritmo do protocolo de recuperação. Partimos, inicialmente, do ponto em que o módulo de consulta propaga as consultas reescritas usando a ontologia global do SDI para os catálogos que ele consegue propagar. A princípio, o catálogo que receber a consulta propagada faz o mapeamento semântico entre a ontologia global e a local (passos 1 e 2 do algoritmo) (algoritmo de tradução explicado na seção 4.6.1). Através dos arquivos de mapeamentos, resultantes do processo de reconciliação entre as ontologias, os catálogos compartilham seus conceitos usados no SDI e interpretam as consultas propagadas pela rede [HPS04]. No passo 3 é verificado se o conjunto de descrições locais atendem a requisição do usuário, caso a condição tenha sido satisfeita ocorre uma filtragem dos serviços de acordo com os parâmetros de qualidade, ordenação pelo tipo de coordenadas e versões. Caso o repositório local não satisfaça a requisição do usuário (passo 6), é preparada a consulta

pelos recursos que ainda faltam para atender a requisição, verifica-se o valor da variável *hops* para checar se já atingiu o limite, caso a variável tenha alcançado o limite máximo o serviço não propaga a consulta e retorna um conjunto vazio para o catálogo que solicitou a requisição inicial, caso a variável não tenha atingido o limite a consulta é propagada. Enquanto, o processo de propagação acontece, o serviço atual “espera” pelo retorno das requisições, o tempo de espera é contado por uma variável no módulo de propagação. Finalmente, o catálogo coleta os serviços obtidos (passo 9), refina o resultado do processamento de acordo com as variáveis de qualidade de serviço, sistema de coordenadas e versões (passo 10) e por fim retorna o resultado para o cliente ou para algum catálogo que iniciou a requisição inicial.

```

Algorithm search (query, hops) {
1  if (localOntology != query.ontology)
2    query = mapping (localOntology, query);
    // semantic matching among the SDI catalogues
3  if (query is fully satisfied)
4    resultSet = refineServicesQoS(QoSParametersRequested,
resultSet);
5  return resultSet; // as service references
  else {
6    resultSet = partialResult();
7    newQuery = mapping (globalOntology, (query - resultSet));
8    if (hops threshold is not achieved) {
9      resultSet = resultSet  $\cup$  otherSDIs.search (newQuery, hops);
      // If a timeout is achieved throws an exception
10   resultSet = refineServicesQoS(QoSParametersRequested,
resultSet);
      return resultSet;
    } else
      // throws an exception;
  }
}

```

#### Algoritmo 4.1

A figura 4.6 exibe um diagrama de seqüência que mostra as interações entre os módulos, serviços e os autores no processamento de uma consulta.



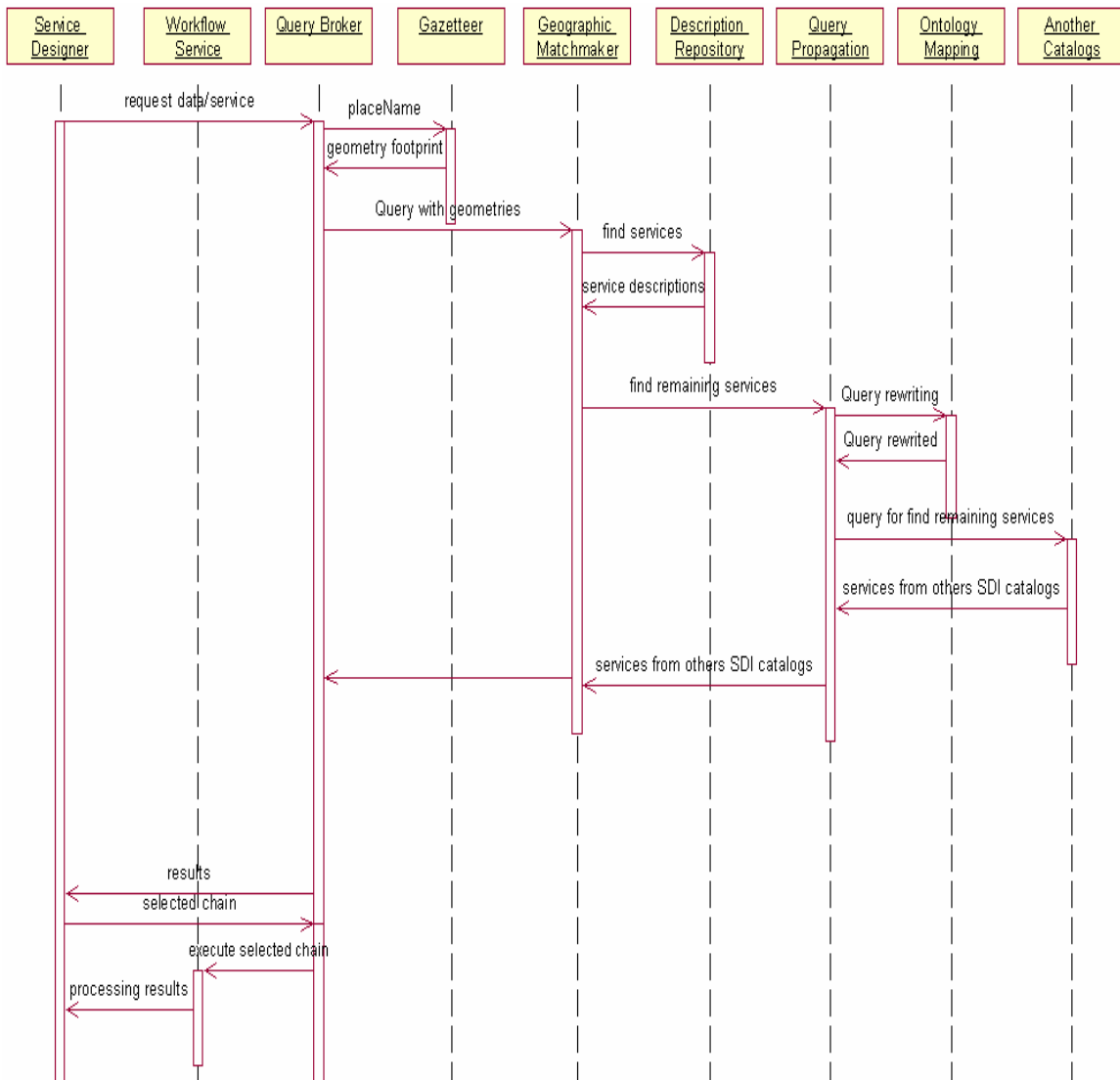


Figura 4.7: Diagrama de seqüência do processo de consulta ao catálogo

#### 4.7.1 Estabelecimento e manutenção da rede de catálogos

A tarefa de estabelecer e manter a rede de catálogos é um processo que depende, inicialmente, do administrador de cada SDI e de cada catálogo. Cada catálogo possui uma referência para uma DHT, que é uma estrutura de dados descentralizada, que gerencia um conjunto de chaves identificadoras nos nodos participantes e encaminham eficientemente as mensagens para um único responsável pelo identificador [ZHS+04] (atualmente utilizamos o serviço de DHT provido pelo OpenDHT [Opn07]). A estrutura de armazenamento dos endereços dos catálogos nas tabelas DHT visa restringir uma quantidade impraticável de mensagens na rede. A composição da tabela DHT é formada por tuplas  $\langle targetLevel, URL\_do\_catálogo \rangle$  onde:

- *targetLevel* - especifica o nível do SDI (estadual, nacional, internacional), bem como o lugar a que pertence o SDI. É através deste parâmetro que os catálogos delimitam a categoria de SDI que irão propagar as consultas.
- *URL\_do\_catálogo* - é apenas o endereço para acessar o *web service* do catálogo. Se o SDI tiver mais de um catálogo então a chave deve ser repetida para todos os catálogos do SDI.

Para o catálogo aderir a rede, basta que o administrador do SDI cadastre a chave *targetLevel* (com o identificador do local do SDI e o nível do SDI) e o endereço do catálogo como uma “tupla” de entrada na tabela DHT, utilizando o método *put* do serviço do *openDHT*. Para ter acesso à tabela DHT ele deve entrar em contato com algum SDI já existente e obter o endereço de acesso do serviço, cada catálogo já provê um módulo que trata com essa parte de acesso à DHT.

Definido e configurado o ambiente da rede de catálogos, o SDI passa a fazer parte da federação, tornando visível aos outros catálogos, os serviços nele publicados. A partir de então, cada catálogo define sua estratégia de propagação (se irá propagar apenas no nível em que se encontra, ou deseja propagar as consultas sempre para catálogos que pertencem à outros níveis, enfim, de acordo com a necessidade). Quando um catálogo decide propagar uma consulta, ele deve:

1. Definir para quais catálogos a consulta será propagada, baseado no parâmetro (*targetsLevel*) definido no processo de propagação para o catálogo.
  - a. Se o usuário busca por recursos que não se encontram no nível do SDI em que se encontra, então o catálogo deve aumentar o nível de busca para englobar outros SDIs, sempre obedecendo à hierarquia semântica definida nas ontologias. Por exemplo, se no nível estadual possui vários SDIs e um usuário busca por informações de outro estado, o catálogo deve aumentar o nível de busca para englobar SDIs de outros estados;
  - b. Cada catálogo deve chamar o método *get* do módulo que provê acesso ao serviço de tabela de DHT, para recuperar os endereços dos catálogos que vai propagar a consulta
2. A partir dos endereços recuperados, o catálogo deve iniciar o processo de propagação, chamando o método do serviço que consulta as descrições dos catálogos.

É importante observar que esta técnica utilizada restringe a quantidade de máquinas que receberão as consultas distribuídas. Desta forma, conseguimos obter resultados

significantes em relação à propagação “cega”, ou seja, para todas as máquinas presentes na DHT.

#### **4.8 Considerações finais**

Neste capítulo foram definidos os requisitos funcionais e não funcionais identificados durante a pesquisa. Em seguida, foi apresentada a arquitetura do *Spatial Data Infrastructure* implementado detalhando cada serviço componente da arquitetura. Destes os serviços descritos, ressaltamos o catálogo, que é responsável por estabelecer a federação de SDI, o serviço de workflow, que é responsável por executar as cadeias de serviços e a hierarquia semântica estabelecida entre as ontologias dos diversos SDIs. Por fim, descrevemos o algoritmo que estabelece o algoritmo de processamento de consultas e o protocolo distribuído estabelecido entre os catálogos federados.

## Capítulo 5 . Estudo de caso

---

Atualmente, são notáveis os esforços das organizações governamentais de vários países, nos diversos continentes, com o objetivo de liderar e promover a proliferação e a evolução dos SDIs. Destacam-se as principais iniciativas destas organizações no estabelecimento de algum tipo de infra-estrutura para compartilhar informações espaciais. Por exemplo, vários estados americanos (Illinois, Iowa, Wyoming), as iniciativas governamentais nacionais (União européia com Inspire, Estados Unidos com o NSDI) e, até mesmo, mundiais (como é o caso do GSDI).

Portanto, considerando o comportamento emergente, o estudo de caso apresentado neste capítulo adota o cenário das organizações observadas. Ou seja, o cenário considera a idéia de que países possuam seus SDIs, bem como os vários estados da nação, ainda sim, as secretárias e órgãos (públicos, privados ou acadêmicos) que compõem cada unidade da federação.

As informações compartilhadas por esses órgãos compreendem diversas áreas, para o estudo de caso adotado, utilizaremos recursos naturais e de infra-estrutura (rodovias). Particularmente, na Paraíba, observamos que os vários órgãos manipulam informações sobre recursos hídricos, tempo, estradas, etc.. Infelizmente, os sistemas de informação utilizados não trocam informações, de forma “interoperável”, o que provoca grande quantidade de replicação das informações citadas. Obviamente, esta replicação introduz vários problemas, como inconsistência dos dados, necessidade de grandes quantidades de mídias para armazenar estas informações, especialmente, quando se trata de informação espacial que se exige grande quantidade de espaço devido às imagens de satélites, mapas vetoriais, entre outros dados. Com o objetivo de resolver esses problemas que propomos o cenário de estudo neste ambiente. Ressaltamos que todos os dados utilizados para elaborar o ambiente foram recuperados a partir do IBGE<sup>1</sup>.

A seguir será apresentado o ambiente de implantação da arquitetura, com os serviços providos por cada SDI, a organização das comunidades semânticas estabelecidas, o exemplo de uso e um detalhamento completo do comportamento de

---

<sup>1</sup> <http://www.ibge.gov.br/>

cada serviço ou componente nos seus respectivos SDI. Por fim, apresentamos os resultados finais do processamento da consulta ilustrativa.

### **5.1 Descrição do cenário do estudo de caso**

Consideremos uma estrutura governamental de estado que, geralmente, é formada por diversas secretarias, órgãos e agências, dentre os quais, tomaremos como exemplo três departamentos e cada um com seu próprio SDI, com seus serviços espaciais a oferecer, são eles:

- Uma agência de gestão de águas, que trata com rios, poços, lagos, açudes, entre outros recursos hídricos;
- Uma superintendência de administração do meio ambiente, como, por exemplo, a SUDEMA que trata com qualidade da água (rios, poços, açudes, lagos, etc.); e
- Uma agência de infra-estrutura como o DER-PB (departamento de estradas e rodagens), que trata com dados das estradas e rodovias.

Hoje, nestes órgãos, existe bastante replicação de informação como, por exemplo, dados sobre recursos hídricos podem ser encontrados na agência de águas e na SUDEMA.

### **5.2 Organização das comunidades semânticas estabelecidas**

Como descrevemos no capítulo 4 (seção 4.6), o catálogo é o componente da arquitetura responsável por integrar os SDIs. Os catálogos dos SDIs modelam seus recursos (descrições de serviços) de acordo com as comunidades semânticas estabelecidas através das ontologias, dos agrupamentos (ou *clusters*) semânticos e dos mapeamentos entre as ontologias. Nesta seção, descreveremos as comunidades semânticas formadas para o estudo de caso.

Consideramos a estrutura das comunidades semânticas apresentadas na figura 4.4. Para o nosso estudo de caso, implementamos as comunidades semânticas no nível de agências (ou organizações) e no nível estadual (Paraíba). Assim, temos a ontologia que descreve os conceitos do SDI no nível estadual, e as ontologias “filhas”, ou seja, ontologias mais específicas de cada secretaria, órgão ou agência, no nosso exemplo, AESA (como a agência de águas), SUDEMA e DER-PB. Observe a importância desta estrutura para manter uma independência semântica entre os órgãos que estão envolvidos, porém, todos devem mapear os conceitos das suas ontologias para a ontologia estadual, que é a ontologia, diretamente, mais global. Seguindo a ordem de

abrangência da comunidade estabelecida, a ontologia da Paraíba deve mapear seus conceitos com a ontologia de um SDI nacional (poderia ser a ontologia da Agência nacional de águas ou uma ontologia mais geral) e esta para um SDI de nível mundial ou continental, dependendo da hierarquia que se pode estabelecer. Continuando o exemplo, temos o exemplo de Pernambuco, que possui suas secretarias, órgãos e agências mapeando as suas respectivas ontologias para o SDI estadual e assim para todos os estados.

Consideramos que o local onde estas ontologias ficam, deve ser conhecido por todos os SDIs. Porém, para cada SDI cabe precisar o local do mapeamento da sua ontologia para a ontologia diretamente mais abrangente, o local da sua ontologia e a da ontologia “pai” ou, diretamente, mais abrangente. Assim, a ontologia da agência de águas possui o mapeamento semântico para a ontologia da Paraíba (no exemplo, consideramos a ontologia do SDI da AESA e do Estado semelhantes), bem como a SUDEMA e o DER-PB. Devido ao comportamento recursivo estabelecido pela própria estrutura das comunidades semânticas, o nosso exemplo é baseado dentro do contexto do Estado da Paraíba. Na próxima seção começaremos a exemplificar o processo de consulta e a interação com os usuários ou agentes externos.

### 5.3 Consultando por serviços espaciais

Suponha que um usuário deseja visualizar um mapa SVG (uma especificação da W3C que é baseada em XML para apresentar gráficos no *browser*) que contém:

- os açudes Paraíba, que estão armazenados na agência de águas;
- os rios da região Sertão que estão armazenados na agência de águas;
- os rios da região Litoral (dados sobre qualidade), que são fornecidos pela SUDEMA;e
- as rodovias de todo o estado, que são mantidas pelo DER-PB.

A interação de um cliente (que se encontra no SDI da AESA, por exemplo) com o catálogo para formar esta aplicação pode ser: (1) o usuário pode consultar o catálogo através do método *getRecords* ou o (2) método *query* passando a consulta em *OCG Filter* como parâmetro destes métodos. Para formular esta consulta, o usuário deve usar os conceitos definidos no SDI local, que estão formalizados na ontologia da agência de águas.

O catálogo é um *web service* que provê suas interfaces para outras aplicações utilizarem. Portanto, desenvolvemos um cliente Web (detalhado na seção 4.6.3) que

utiliza tal serviço. Na figura 5.1 é ilustrado este primeiro processo de consulta pelo usuário.



Figura 5.1: Processo de consulta ao catálogo

Na figura 5.1 mostra o processo da formulação da consulta, no caso da figura, o usuário está utilizando o campo *feature name* para que o catálogo obtenha os dados geográficos da Paraíba (via serviço de gazetteer) e possa dar seqüência à consulta. Além disso, é selecionado o tipo de serviço para WFS (*service type*) e o tipo da feição (*feature type*) desejada, que é açude (*Dam*). Continuando a consulta, o usuário deve clicar no botão *ADD*, isto gera o código em *OGC Filter* e apresenta no campo *Filter encoding*. A figura 5.2 apresenta o passo seguinte no processo de consulta, onde o cliente opta por selecionar o *bounding box* da região sertão, rios (*Rivers*) como o tipo da feição, não optou por nenhum tipo de serviço — o que significa que qualquer serviço o interessa, seja WFS, seja WMS —, operação espacial selecionada é *inside* com o *bounding box*

selecionado. Observe que a parte que especifica o nome da feição não está preenchida e o campo *Filter encoding* mostra a consulta gerada a partir dos campos selecionados.

**Catalog Service**

- Feature name:
- Service type: All
- Feature type: River
  - Hydrography
    - River
    - Dam
    - Pala
- Logical operation:
  - Or  And
- Spatial operation:
  - Inside  Buffer
  - Contains  Overlaps
  - Disjoint
- Presentation in client:
  - Yes  No

Buttons: Add, Clear, Search

Map: Select a polygon on the map.  
 Point 1: -38.8476, -8.01571 | Point 2: -37.0898, -6.05316  
 Place: Campina Grande, Paraíba, Brazil [Go!]

**Filter encoding**

```
<ogc:Filter
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsischemaLocation="http://www.opengis.net/ogc/filter.xsd"
  xmlns:gml="http://www.opengis.net/gml"
  xsi:schemaLocation="http://www.opengis.net/ogc
```

Figura 5.2: Processo de consulta ao catálogo

Na figura 5.3 mostra o terceiro passo no processo de consulta, nesta etapa, o usuário seleciona o *bounding box* dos rios do litoral, WFS como tipo de serviço desejado (por causa dos dados de qualidade), rios como tipo de feição desejada, operação espacial *inside* entre as descrições do catálogo e o *bounding box* passado e, por fim, clica no botão *Add*, para adicionar mais uma restrição à consulta e esta ser apresentada no campo *Filter encoding*.



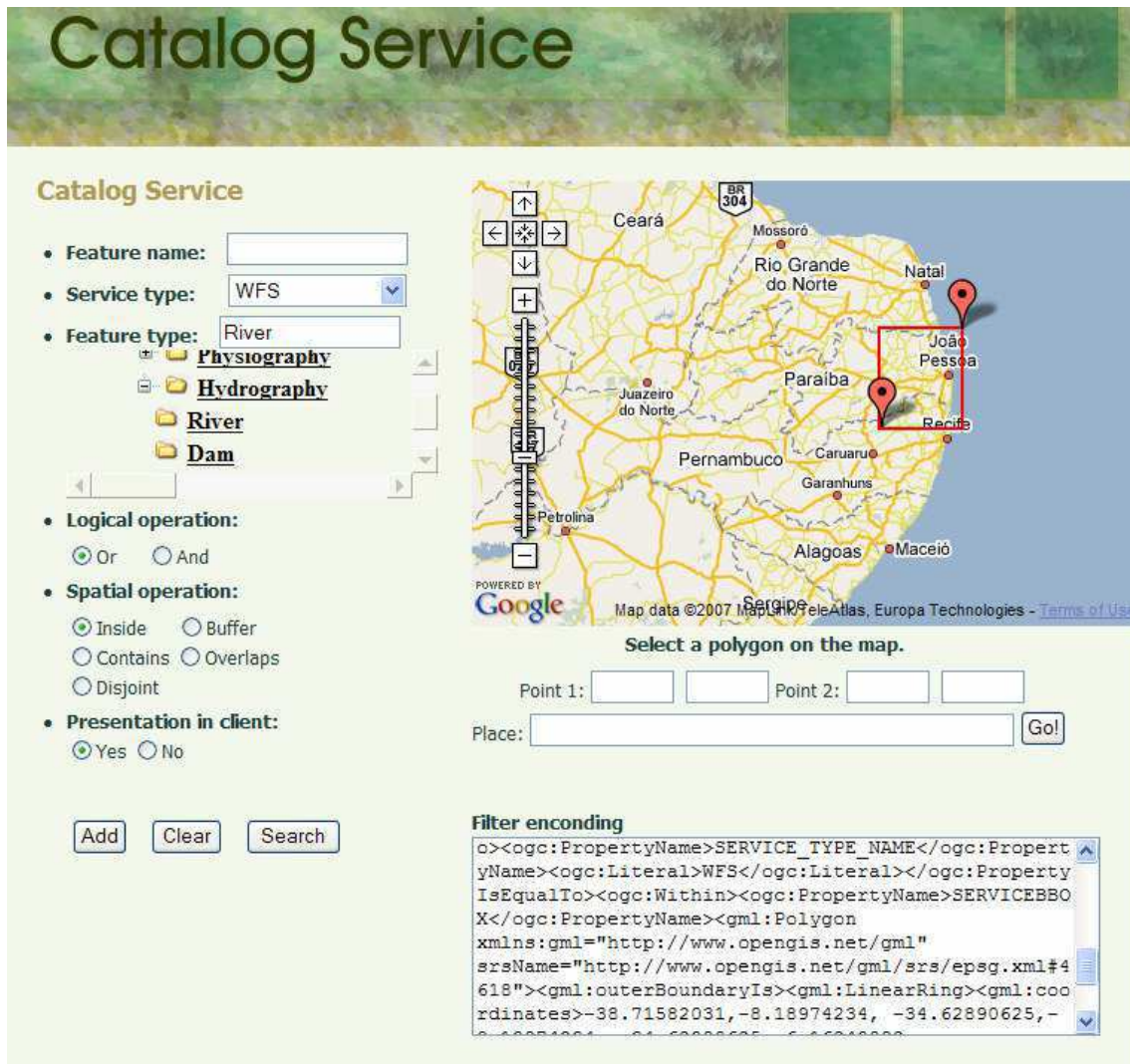


Figura 5.3: Processo de consulta ao catálogo

Finalmente, é mostrado, na figura 5.4, o último passo do processo de consulta é a requisição por estradas da Paraíba. Para tanto, o usuário deve selecionar o tipo de feição estrada (*Road* na figura 5.5), tipo de serviço *All*, que significa qualquer tipo de serviço, mais uma vez, o usuário optou por colocar o nome da Paraíba no campo *Feature name* e a operação espacial continua sendo o *inside*. Após o último clique no botão *Add*, para adicionar esta consulta, o usuário deve clicar no botão *Search*, para que o programa cliente submeta a consulta final para o serviço de catálogo. Dando início ao processamento da consulta no serviço, que será apresentado na seção que segue.



Figura 5.4: Processo de consulta ao catálogo

### 5.3 Processamento da consulta pelo catálogo

Após receber a consulta, o serviço de catálogo da AESA segue o algoritmo definido no capítulo 4. Logo em seguida, o catálogo realiza os seguintes passos:

1. O catálogo consulta o serviço de *gazetteer* do SDI local, com o objetivo de obter as feições espaciais (coordenadas) dos nomes de lugares passados na consulta, no caso foi passado o nome Paraíba. Com as feições geográficas, o catálogo poderá realizar as operações espaciais para encontrar os serviços que possuem rios, açudes e as estradas solicitadas;
2. O catálogo verifica na base de descrições do SDI local se existem serviços que atendam à consulta. Para tanto, o catálogo realiza consultas espaciais na sua base de descrições, verificando se existem serviços que possuam rios do Sertão, qualidade destes rios e estradas da Paraíba;

3. O catálogo encontra um WFS com o mapa do estado da Paraíba, um WFS que fornece os rios do Sertão e um WMS com açudes da Paraíba, entretanto, não foram encontrados serviços que possuem informações de qualidade de água nem serviços que possuam mapas das estradas da Paraíba. Desta forma, o catálogo propaga uma consulta dos serviços que faltam para os outros catálogos. Por isto, o catálogo reformula a consulta para que solicite apenas os serviços que não foram encontrados na sua base de descrições, conforme foi visto no protocolo da seção 4.7. A ontologia que o catálogo usa para reescrever a consulta é a, diretamente, mais abrangente, que funciona como uma ontologia global para os SDIs das secretarias e órgãos estaduais, que neste exemplo é a ontologia do SDI do estado da Paraíba;

Segue no trecho de código 5.1 um exemplo da consulta inicial feita pelo usuário ao catálogo do SDI da AESA. O trecho de código 5.1 mostra a consulta por Rios do sertão (observe as coordenadas já considerando a feição do Sertão), com qualidade (observe o termo *hasQuality*, extraído da ontologia da AESA), estradas da Paraíba (*Roads*) e, por fim Açudes

```

<ogc:Filter>
  <ogc:Or>
    <ogc:And>
      <PropertyIsEqualTo>
        <PropertyName>FeatureName</PropertyName>
        <Literal>River</Literal>
      </PropertyIsEqualTo>
      <PropertyIsEqualTo>
        <PropertyName>hasQuality</PropertyName>
        <Literal>yes</Literal>
      </PropertyIsEqualTo>
    <Overlaps>
      <PropertyName>Geometry</PropertyName>
      <gml:Polygon srsName="urn:ogc:def:crs:EPSG:6.6:4618">
        <gml:exterior>
          <gml:LinearRing>
            <gml:posList>
              -37.001821,-7.175744 -36.992499,-7.180127 -36.976542,-7.160474
              ... -37.019035,-7.190465 -37.001821,-7.175744
            </gml:posList>
          </gml:LinearRing>
        </gml:exterior>
      </gml:Polygon>
    </Overlaps> </ogc:And>
  </ogc:Or>
  <ogc:And>
    <PropertyIsEqualTo>
      <PropertyName>FeatureName</PropertyName>
      <Literal>Road</Literal>
    </PropertyIsEqualTo>
    <gml:posList>

```

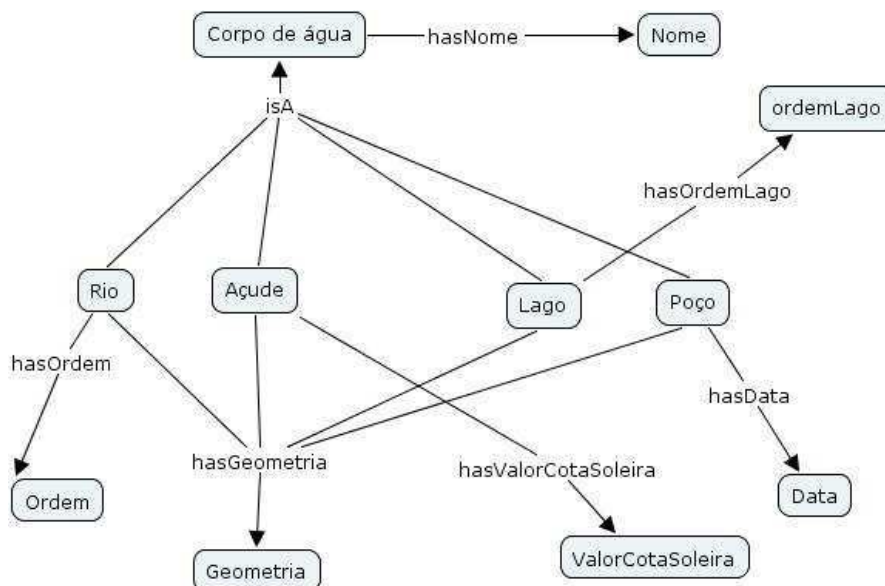
```

-38.76516,-6.91093 -38.75917,-6.9155 -38.75912,-6.91657 ... -
38.75622,-6.9036 -38.76516,-6.91093
  </gml:posList>
</ogc:And>
<ogc:And>
  <PropertyIsEqualTo>
    <PropertyName>FeatureName</PropertyName>
    <Literal>Dam</Literal>
  </PropertyIsEqualTo>
  <gml:posList>
    -38.76516,-6.91093 -38.75917,-6.9155 -38.75912,-6.91657 ... -
38.75622,-6.9036 -38.76516,-6.91093
  </gml:posList>
</ogc:And>
</ogc:Or>

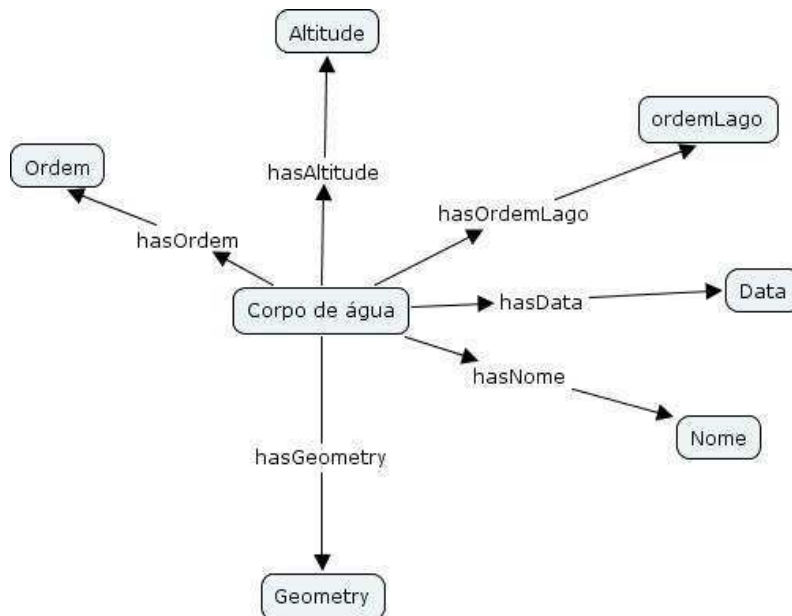
```

**Trecho de código 5.1: OGC Filter contendo a consulta inicial ao catálogo**

4. Neste passo, a consulta é tratada pelos catálogos que recebem a consulta. Todos os catálogos que recebem a consulta devem traduzir os conceitos utilizados na consulta que foi recebida (utilizando a ontologia global, ou seja, a ontologia do estado da Paraíba) para uma consulta que utiliza os conceitos de cada SDI local. Todo o processo de tradução é auxiliado pelos arquivos de mapeamento que cada SDI possui para a ontologia global. A figura 5.3a apresenta parte da ontologia da AESA que, casualmente, é a mesma do estado da Paraíba. A figura 5.3b apresenta a ontologia da SUDEMA. É importante notar que na ontologia da AESA e na ontologia global (no caso, a ontologia estadual) a classe “Corpo de Água” é uma super classe para açudes, rios, e poços. Entretanto, na ontologia da SUDEMA não existem os conceitos de rios, poços, ou açude, todos são tratados como uma única classe chamada “Corpo de Água”.



**Figura 5.5 a): Exemplo de parte da ontologia da AESA e do Estado;**



**Figura 5.5 b): Exemplo de parte da ontologia utilizada na SUDEMA.**

No trecho de código 5.2 e 5.3 é mostrado exemplos de arquivos de mapeamentos entre as ontologias em OWL-S. Assim, o conceito de “Corpo de Água” na SUDEMA é mapeado diretamente para o mesmo conceito na ontologia do estado. Todavia, o conceito de “Rio” na ontologia estadual é mapeado para o conceito de “Corpo de Água” na ontologia da SUDEMA através da propriedade “hasType” definida na tag “DataType” no conceito “river”.

```

<owl:Class rdf:ID="BodyOfWater1SUDEMA">
  <owl:equivalentClass rdf:resource="BodyOfWaterAESA"/>
</owl:Class>

```

**Trecho de código 5.2: Mapeamento entre conceitos equivalentes entre as ontologias envolvidas no processo de tradução.**

```

<rdf:RDF xmlns="http://localhost/ontologias/sudema.owl#"
  xml:base="http://localhost/ontologias/sudema.owl"
  xmlns:globalSDI="http://localhost/ontologias/globalSDI.owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#">
  <owl:Ontology rdf:about="">
    <owl:imports
  rdf:resource="http://localhost/ontologias/aesa.owl"/>
  </owl:Ontology>
  <owl:Class rdf:ID="Body_of_Water">
    <rdfs:subClassOf rdf:resource="&owl;Thing"/>
    <rdfs:subClassOf>
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <owl:Class rdf:about="&globalSDI;Dam"/>
          <owl:Class rdf:about="&globalSDI;Lake"/>

```

```

        <owl:Class rdf:about="&globalSDI;Pole"/>
        <owl:Class rdf:about="&globalSDI;River"/>
    </owl:unionOf>
</owl:Class>
</rdfs:subClassOf>
</owl:Class>
<owl:DatatypeProperty rdf:ID="hasType"/>
<owl:Class rdf:ID="Geometry"/>
</rdf:RDF>

```

**Trecho de código 5.3: União dos conceitos da ontologia global no conceito de “Corpo de Água” da ontologia da SUDEMA**

5. Em seguida, cada catálogo dos SDI da rede submete a consulta (já utilizando os conceitos locais) para sua base de descrição local, com o objetivo de tentar encontrar cadeias ou serviços que atendam a requisição que foi distribuída. No caso, a SUDEMA encontra apenas informações de qualidade da água dos rios do sertão e o catálogo do DER-PB encontra apenas serviços que possuem mapas das rodovias e estradas do estado;
6. Após verificada as respectivas bases de descrições, os catálogos da SUDEMA e do DER descartam a parte da consulta propagada que foi atendida e repassam a parte que não conseguiram atender, pois a consulta propagada somente o DER e a SUDEMA conseguiram atender, mas nenhum conseguiu atender a requisição completa.. Desta forma, a SUDEMA repassa a consulta por estradas da Paraíba (já que foi encontrado localmente um WFS com a qualidade dos rios) para outros catálogos da rede, com exceção do catálogo que recebeu na primeira requisição, ou seja, o catálogo da AESA. Enquanto o catálogo do DER-PB repassa a consulta por qualidade de rios para os outros catálogos, já que este retornou um WFS contendo mapas das estradas da Paraíba. Além de propagar a consulta para os outros catálogos, cada catálogo que repassa a consulta deve subtrair em uma unidade um atributo chamado *hopCount* (este atributo segue junto à requisição e limita a quantidade de catálogos que a mensagem ficará sendo propagada na rede);
7. O catálogo que iniciou o processo, ou seja, o catálogo da AESA fica recuperando os resultados dos catálogos distribuídos durante certo tempo. Após essa etapa, é iniciada a etapa de construção das cadeias que seguirá os algoritmos de formação de cadeia definidos no capítulo 4. Contudo, foram recebidas referências de dois serviços (WFS) que possuem dados de qualidade de rios e das estradas e rodovias da Paraíba. O módulo MCSG forma a cadeia de serviço com os serviços que foram encontrados localmente e repassa para o



serviço de *workflow* executar a cadeia. A figura 5.4 representa a cadeia sendo formada e a procedência de cada SDI.

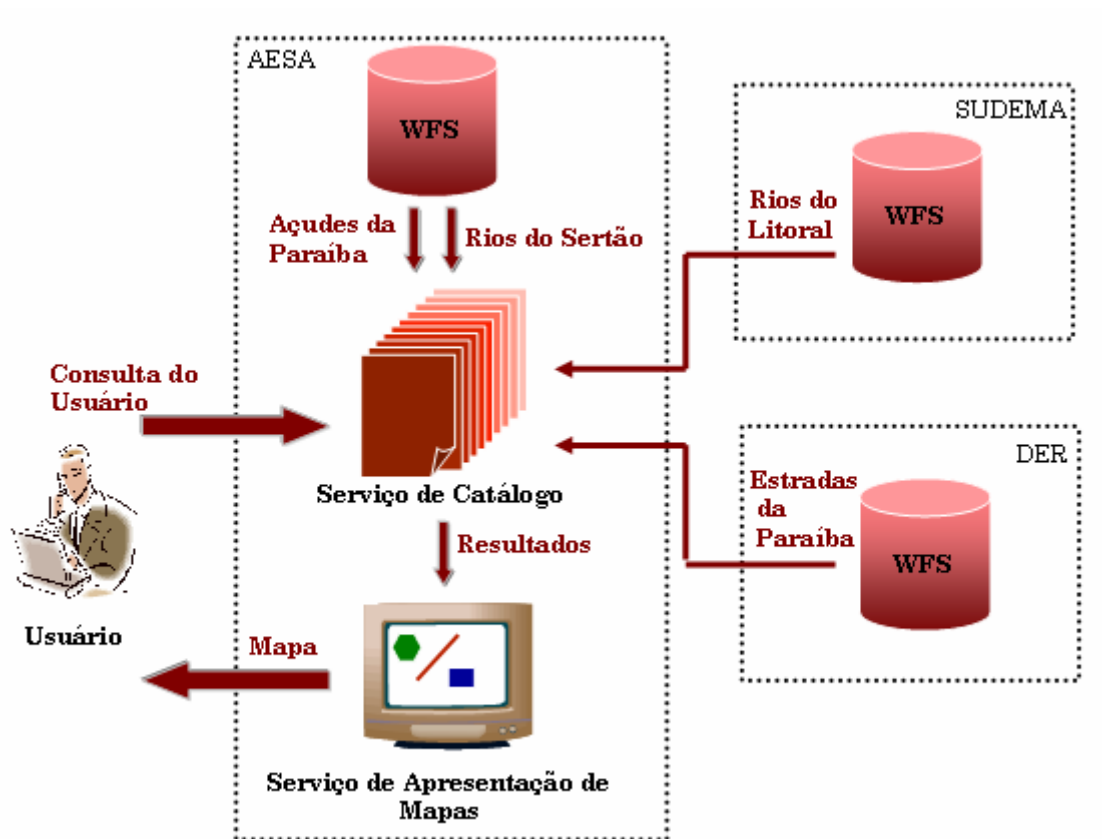
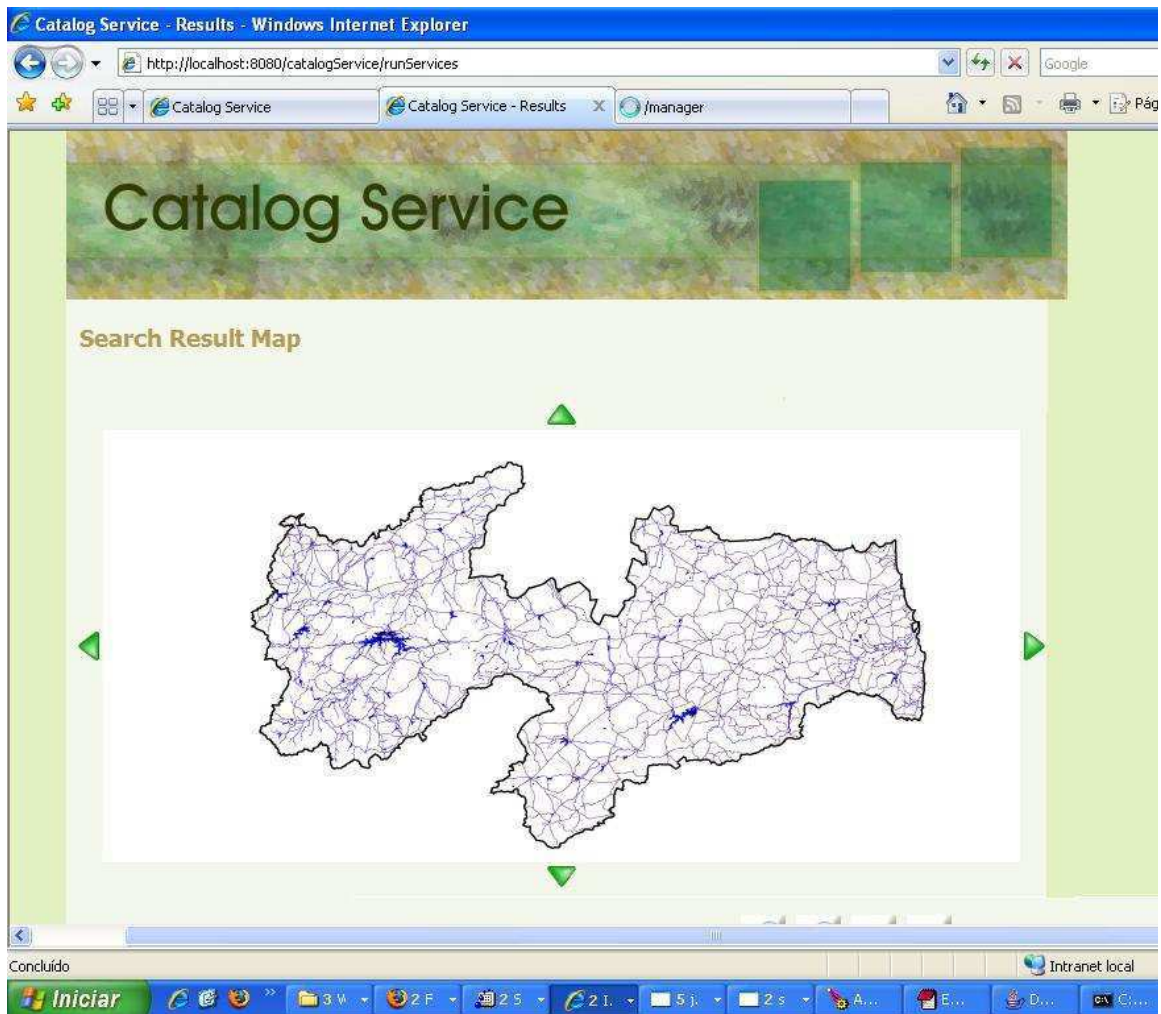


Figura 5.6: Serviços retornados pelos catálogos distribuídos em outros SDI e a cadeia formada com o serviço de apresentação.

8. O serviço de *workflow* executa a cadeia e monitora o comportamento dos serviços envolvidos, para tentar tratar algum tipo de erro, falha, ou atraso que venha acontecer na cadeia. Também, gerencia as entradas e as saídas de cada serviço da cadeia, transformando o GML retornado pelos serviços de WFS de SDIs com ontologias diferentes para uma única sintaxe em comum, bem como transformar este GML em SVG, se o usuário desejar ver os resultados no serviço de apresentação de mapas. Como todos possuem a ontologia do SDI da Paraíba em comum, então o serviço de *workflow* deve utilizar a ontologia da Paraíba para descrever a saída final destes serviços.

O resultado final é retornado para o cliente que pode verificar os resultados apresentados no mapa, como mostrado na figura 5.3.



**Figura 5.7: Resultado final apresentado no mapa, com os rios do sertão, estradas e rodovias do estado e açudes.**

#### **5.4 Considerações finais**

Este capítulo apresentou um estudo de caso que ilustra o processamento de uma consulta por serviço usando o catálogo SWS-GIS numa situação que envolve problemas reais do estado da Paraíba. Foi mostrada a utilização da ferramenta por um usuário que constrói aplicações baseada em *web services* e que busca por informações espaciais para apresentar para os clientes. Ainda mais, foi mostrado como os algoritmos apresentados no capítulo 4. Foi mostrada também como se dá a interação entre os diversos catálogos que configuram a integração entre vários SDIs.



## Capítulo 6 . Conclusão

---

Devido aos avanços das tecnologias de SDI e *web services*, a evolução da interoperabilidade de sistemas de informações geográficas tem obtido auspiciosos patamares. Consequentemente, há uma melhor disseminação e utilização da informação espacial. Esta evolução, aliada ao fato do avanço das tecnologias de localização de serviços espaciais, tem pressionado os SDIs para desenvolver catálogos espaciais mais eficazes, e que, ao mesmo tempo, utilizem descrições de serviços com maior poder de expressividade semântica, considerando a dimensão espacial das descrições de serviços e permitindo o compartilhamento entre os catálogos de vários SDIs.

Neste trabalho, apresentamos uma arquitetura que promove o compartilhamento de recursos espaciais de forma “interoperável” entre vários SDIs. Para tanto, foi estabelecido uma rede de comunidades semânticas (*clusters* semânticos) entre os catálogos dos SDIs. Onde cada catálogo descreve os serviços utilizando ontologias de domínio e recupera os serviços utilizando a dimensão espacial destas descrições. Destarte, é possível realizar consultas espaciais sobre as descrições dos catálogos, que formam uma rede semântica, cujas consultas são distribuídas e seguem um protocolo baseado na dimensão espacial das descrições, otimizando continuamente o processo de consulta nos catálogos distribuídos. Apresentamos as principais contribuições na seção 6.1 e o conjunto de trabalhos futuros identificados na seção 6.2.

### 6.1 Principais contribuições

A principal contribuição desta dissertação é o desenvolvimento do SWS-GIS, uma arquitetura orientada a serviços para o compartilhamento de serviços espaciais em ambientes de SDIs. Esta arquitetura torna-se relevante devido aos seguintes aspectos:

- **Arquitetura orientada a serviços:** devido à flexibilidade possibilitada pela a arquitetura, o usuário do SDI, apenas deve acessar o catálogo para localizar os serviços espaciais, o que permite ao cliente localizar e utilizar todas as ferramentas que necessita mediante as consultas. Da mesma forma que a arquitetura permite a facilidade de obter os serviços desejados, esta também permite a flexibilidade para que o usuário possa escolher e buscar qual o melhor

serviço que atende aos seus requisitos, compondo novas aplicações sempre que necessário. Outra vantagem desta arquitetura é que atende tanto aos usuários que possuem já alguns serviços espaciais, como, por exemplo, um serviço de apresentação de mapas que deseja obter apenas os dados; quanto aos usuários que não possuem nenhuma ferramenta para utilizar;

- **Interoperabilidade entre os SDIs:** a arquitetura desenvolvida permite que vários SDIs troquem informações (descrições de serviços e dados do WFS) de forma transparente ao usuário. Portanto, um usuário que se encontra num SDI da Paraíba, por exemplo, pode consumir serviços providos pelo SDI de Pernambuco como se o seu próprio catálogo contivesse a descrição de tal serviço. Porém, o catálogo local localiza o serviço desejado utilizando a rede de semântica estabelecida para que os termos utilizados sejam intercambiáveis entre os catálogos.
- **Flexibilidade das comunidades semânticas:** Através dos agrupamentos semânticos (*clusters*), é possível manter separadamente as ontologias de cada SDI, de forma que cada SDI defina o conjunto de conceitos independente dos outros SDI da rede, bastando, apenas, que faça o mapeamento para a ontologia global da comunidade. Assim, diferente de se obter uma ontologia global para todos os SDIs, esta arquitetura permite que vários SDIs cooperem, já que, como visto no trabalho de Hameed [HPS04], o desenvolvimento de uma única ontologia, muitas vezes, é inviável. Ao mesmo tempo, que limita, consideravelmente, a quantidade de mapeamentos que cada SDI precisa fazer, diferente da estratégia de vários mapeamentos entre todas as ontologias;
- **Consultas espaciais sobre os recursos dos catálogos:** É possível realizar consultas espaciais sobre as descrições de serviços dos catálogos. Por isto, os usuários podem selecionar serviços que possuem dados relativos apenas a uma determinada área geográfica desejada;
- **Protocolo de consultas espaciais distribuídas:** Ao consultar os catálogos distribuídos é estabelecido um protocolo de otimização da consulta baseado na dimensão espacial dos recursos obtidos. Desta forma, ao decidir propagar uma consulta pela rede de catálogos, cada catálogo deve buscar (na rede) apenas pelos serviços que ainda não atendem a requisição espacial. Portanto, a

quantidade de mensagens e informações trafegadas é melhorada consideravelmente;

- **Execução de serviços baseada em *workflow***: Existe um serviço de *workflow* que executa as cadeias de serviços obtidas pelo catálogo. Com este serviço, é possível monitorar a execução das cadeias de serviços. A grande vantagem de um serviço de *workflow* é que o usuário não necessita executar a cadeia, eximindo-o de tratar com os detalhes de execução dos serviços. Outra grande vantagem deste serviço é que é possível ao usuário optar por outros serviços de *workflow*, se assim desejar melhorar a qualidade da execução e monitoramento das cadeias de serviços.

Foi elaborado um estudo sobre diversos trabalhos relacionados que apresentamos no capítulo 3 e identificamos as características apresentadas na tabela 6.1. Devemos ressaltar que o SWS-GIS realiza a descoberta de serviços espaciais em catálogos (padronizados pela OGC) distribuídos de forma a promover a interoperabilidade das informações dos catálogos, além disso, a arquitetura desenvolvida leva em consideração os fatores de execução das cadeias de serviços, o que define uma solução mais completa, tratando de aspectos de recuperação e execução dos serviços espaciais.

Trabalhos analisados	Características				
	<i>Utiliza federação de catálogos espaciais</i>	<i>Utiliza buscas espaciais em catálogos</i>	<i>Padrão OGC</i>	<i>Forma de interação da cadeia de serviços (Static, Client, Workflow)</i>	<i>Trata processamento de consultas distribuídas</i>
<i>Lemmens et. al</i>	Não	Não	Não	Client	Não
<i>Hubner et. al</i>	Sim	Sim, para buffer	Parcialmente	Client	Não
<i>Boucelma et. al</i>	Não	Não	Sim	Static	Parcial*

<b>Trabalhos analisados</b>	<b>Características</b>				
	<i>Utiliza federação de catálogos espaciais</i>	<i>Utiliza buscas espaciais em catálogos</i>	<i>Padrão OGC</i>	<i>Forma de interação da cadeia de serviços (Static, Client, Workflow)</i>	<i>Trata processamento de consultas distribuídas</i>
<i>Sivashanmugam et.al</i>	Não	Não	Não	<i>Client</i>	Não
<i>Friis-Christensen</i>	Não	Não	Sim	Não trata	Não
<i>Lutz</i>	Não	Não	Não	Não trata	Não
<i>Zimmerman et al</i>	Não	Sim	Não	Não trata	Sim
<i>Ghosh e Paul</i>	Não	Sim	Sim	Não trata	Não
<i>Xiujun, M et al</i>	Sim	Sim	Não	Não trata	Parcial
<i>SWS-GIS</i>	Sim	Sim	Sim	<i>Workflow</i>	Sim

**Tabela 6.1 Diferenças entre as soluções apresentadas e o trabalho desenvolvido**

\*Não considera consultas espaciais nos catálogos distribuídos.

De acordo com a tabela 6.1, os resultados apresentados atingem os objetivos desta dissertação. Não obstante alguns pontos de melhoria que foram identificados ao longo da elaboração deste trabalho, que são apresentados na seção que segue.

## **6.2 Trabalhos futuros**

A tecnologia de *web services* está em constante mudança, seja definindo novos padrões, seja evoluindo as tecnologias existentes. Da mesma forma, o OGC e os outros órgãos relacionados à definição de padrões de geoprocessamento estão em constante evolução, o que obriga sempre os SIGs estarem em constante evolução, bem como os próprios

avanços das tecnologias. Tais evoluções impactam diretamente no SWS-GIS. Além disso, ao longo do nosso trabalho identificamos várias melhorias de funcionalidades ainda em andamento ou não implementadas, segue o detalhamento de cada uma:

- **Casamento automático entre as ontologias:** Atualmente, as ontologias são mapeadas utilizando um processo semi-automático que depende fortemente do administrador do sistema. São necessárias estratégias que permitam uma heurística de mapeamento de ontologias mais eficaz e que diminua a participação humana deste processo;
- **Desenvolvimento de uma estrutura de indexação distribuída mais eficiente:** Como no trabalho de Xiujun et. al [XGK+06] poder-se-ia indexar os recursos utilizando uma estrutura como a R+tree ou outra estrutura de dados. Isto traria melhorias significativas ao protocolo de consultas distribuídas.
- **Desenvolvimento de serviços de workflow mais eficientes:** Atualmente, o serviço de *workflow* desenvolvido permite apenas a execução das cadeias. É necessário, e a arquitetura é flexível ao prever isto, desenvolver serviços de *workflow* para monitorar contratos de QoS, tratamento e recuperação de falhas nas cadeias executadas, entre outras possibilidades. Outro trabalho futuro poderia avaliar a viabilidade da migração de serviços de *workflow* para outras formas de execução de cadeias, como, por exemplo, BPEL (*Business Process Execution Language for Web Services*) [ACD+04];
- **Maior tratamento de QoS:** Realizar um estudo mais aprofundado sobre quais os requisitos de QoS podem ser adicionados às descrições de serviços. Considerando estes novos recursos no protocolo de recuperação de serviços distribuídos;
- **Verificação da correção das composições:** um modelo formal para a verificação da correção de composições de serviços pode ser elaborado para garantir uma maior precisão nas composições de serviços descobertas.
- **Testes envolvendo um ambiente de larga escala:** como o objeto de estudo é geograficamente distribuído, devem ser realizados maiores simulações utilizando a rede da Internet para interligar catálogos de SDI em diversos estados e países. Por falta de tempo, tivemos que restringir nosso estudo de caso nesta dissertação.

## Referências Bibliográficas

- [Ali07] Alignapi. <<http://gforge.inria.fr/projects/alignapi/>> Acessado em 2007.
- [Alm06] Almeida, D., Baptista, S., Silva E., Campelo, C., Figueiredo, Lacerda, Y., *Context-Aware System Based on Service-Oriented Architecture*, AINA, Proceedings of the 20th International Conference on Advanced Information Networking and Applications - Volume 1 (AINA'06), IEEE Computer Society, 2006, 01, 205-210
- [ACD+04] Andrews, T., M., Curbera F., I., Dholakia, H., S., S., Yaron, G., B., Johannes Klein, M., Frank Leymann, I., Kevin Liu, S., Dieter Roller, I., Doug Smith, S. S., Ivana Trickovic, S., and Sanjiva Weerawarana, I. *Bpel4ws. business process execution language for web services*, Packt Publishing, Outubro, 2004.
- [ACK+03] Alonso, G., Casati, F., Kuno, H., and Machiraju, V. *Web Services*. Springer, Novembro 2003.
- [Alm03] Alameh, N. *Chaining geographic information web services*. IEEE Internet Computing 07 (September 2003), 22-29.
- [Anz05] Anzlic, *the spatial information council of australia and new zealand*. Disponível em: <<http://http://www.anzlic.org.au/>>. Acessado em 10 dez. 2005.
- [BC05] Bernard, L., and Craglia, M. *Sdi - from spatial data infrastructure to service driven infrastructure*. First Research Workshop on Cross-learning on Spatial Data Infrastructures and Information Infrastructures. Enschede, the Netherlands (2005).
- [BCP03] Bass, L., Clements, P., Kazman, R., *Software Architecture in Practice*, Addison-Wesley Professional, segunda edição, Abril 2003.
- [BEL02] Boucelma, O., Essid, M., and Lacroix, Z. *A wfs-based mediation system for gis interoperability*. In GIS '02: Proceedings of the 10th ACM international symposium on Advances in geographic information systems (New York, NY, USA, 2002), ACM Press, pp. 23-28.

- [BHL01] Berners-Lee, T., Hendler, J., and Lassila O.. *The Semantic Web*. Scientific American, maio de 2001.
- [Bis98] Bishr, Y. *Overcoming the semantic and other barriers to gis interoperability*. International Journal of Geographical Information Science 12 (1998), pp. 299-314.
- [BKA+05] Bernard, L., Kanellopoulos, I., Annoni, A., and Smits, P. *The european geoportal- one. step towards the establishment of a european spatial*. Computers, Environment and Urban Systems 29 (2005), pp. 15-31.
- [BFM+03] Bruijn ,J., Martín-Recuerda, F., Manov, D., and Ehrig, M.. *State-of-the-art survey on ontology merging and aligning v1*. Deliverable D4.2.1, SEKT, 2004
- [CGM+94] Chawathe, S., Garcia-Molina, H., Hammer, J., Ireland, K., Papakonstantinou, Y., Ullman, J. D., and Widom, J. *The TSIMMIS project: Integration of heterogeneous information sources*. In 16th Meeting of the Information Processing Society of Japan (Tokyo, Japan, 1994), pp. 7-18.
- [CDL+04] COX, S., Daisey, P., Lake, R., Portele, C., and Whiteside, A. *Gml. geography marckup language*, 2004.
- [CBR+04] Cromptvoets, J., Bregt, A., Rajabifard, A., and Williamson, I. *Assessing the worldwide developments of national spatial data clearinghouses*. International Journal of Geographical Information Science 18, 7 (2004), 665-689.
- [CSC04] Cruz, I. F., Sunna, W., Chaudhry, A. *Ontology alignment for real-world applications*. In Proceedings of the 2004 Annual National Conference on Digital Government Research (Seattle, WA, May 24 - 26, 2004). ACM International Conference Proceeding Series. Digital Government Research Center, 1-2.
- [DA05] Davis, C., and Alves, L. L. *Local spatial data infrastructures based on a serviceoriented architecture*. GEOINFO - Brazilian Symposium on GeoInformatics (November 2005).

[Fed05] *Federal geographic data committee*. Disponível em <<http://www.fgdc.org>>. Acessado em 10 dez. 2005.

[FKT01] Foster, I., Kesselman, C., and Tuecke, S. *The anatomy of the grid: Enabling scalable virtual organizations*. International Journal of Supercomputer Applications, Lecture Notes in Computer Science 15, 3 (2001), 200-222.

[Fow02] Fowler M., *Patterns of Enterprise Application Architecture*, Addison-Wesley Professional, primeira edição, Novembro 2002.

[Gsd05] *Gsdi, global spatial data infrastructure association*. Disponível em <<http://www.gsdi.org/>>. Acessado em 10 dez. 2005.

[GG95] Guarino, N., and Giaretta, P.. *Towards a terminological clarification*. In N.J.I. Mars, editor, *Towards Very Large Knowledge Bases*, IOS Press, 1995.

[GOS07] *Geospatial One-Stop*. Disponível em <<http://www.fws.gov/data/gisGOS.html>> Acessado em maio de 2007

[HPS04] Hameed, A., Preece, A. and Sleeman, D.. *Ontology reconciliations*, In: Handbook on Ontologies, (2004), Edited by S. Staab and R. Studer, Springer-Verlag, Germany

[HSV+04] Hübner, S., Spittel, R., Visser, U., and Vögele, T. J. *Ontology-based search for interactive digital maps*. IEEE Intelligent System 19 (June 2004), 80-86.

[Ida07] *Idaho GIS Data*. Disponível em: <[http://www.idwr.state.id.us/gisdata/gis\\_data.htm](http://www.idwr.state.id.us/gisdata/gis_data.htm)>. Acessado em 15 mai. 2007.

[Ill07] Illinois Natural Resources Geospatial Data Clearinghouse. Disponível em: <<http://www.isgs.uiuc.edu/nsdihome/>>. Acessado em 15 mai. 2007.

[Ins05] *Inspire, the infrastructure for spatial information in europe*. Disponível em <<http://www.ec-gis.org/inspire>>. Acesso em 10 dez. 2005.



- [Iow07] *Iowa Geographic Information Council*. Disponível em: <<http://www.iowagic.org/about/committee/clearinghouse> >. Acessado em 15 mai. 2007.
- [IWS07] IBM Web Services developers work. Disponível em: <<http://www.ibm.com/developerworks/webservices>> Acessado em 15 de maio 2007.
- [Jen07] Jena – A Semantic Web Framework for Java < <http://jena.sourceforge.net/> > Acessado em maio 2007
- [Kuh05] Kuhn, W. *Geospatial semantics: Why, of what, and how?* J. Data Semantics III (2005), 1-24.
- [LGM+05] Longley, P. A., Goodchild, M. F., Maguire, D. J., and Rhind, D. W. *Geographic Information Systems and Science*. John Wiley & Sons, April 2005.
- [Lut05] Lutz, M., *Ontology-based service discovery in spatial data infrastructures*, Workshop On Geographic Information Retrieval In Proceedings of the 2005 workshop on Geographic information retrieval. ACM Press (2005), 45-54.
- [LWR+06] Lemmens, R., Wytzisk, A., de By, R., Granell, C., Gould, M., Oosterom, P., van, *Integrating Semantic and Syntactic Descriptions to Chain Geographic Services*, IEEE Internet Computing, (Setembro - Outubro 2006), 42-51
- [ML05] Maguire, D. J., and Longley, P. A. *The emergence of geoportals and their role in spatial data infrastructures*. Computers, Environment and Urban Systems 29, 1 (2005), 3-14.
- [Mar04] Martell, R. Ogc catalogue services - ebrim (iso/ts 15000-3) profile of csw, 2004.
- [NSI07] National Spatial Information Framework Clearing House Node. em <<http://www.nsif.org.za/sddf/spatial2.html>> Acessado em maio de 2007.
- [OGC07] *The Open Geospatial Consortium* < <http://www.opengeospatial.org/>> Acessado em maio de 2007
- [Opn07] *Open DHT* < <http://opendht.org/> > Acessado em maio de 2007.

[OLA07] Owl Lite Alignment. <<http://www.iro.umontreal.ca/~owlola/alignment.html>> Acessado em 2007

[OOA07] OWL ontology aligner. <<http://align.deri.org:8080/deri/align.jsp>> Acessado em 2007.

[Owl05] *Owl. web ontology language*. Disponível em <<http://www.w3.org/TR/owl-features/>>. Acessado em 10 dez. 2005.

[PBM02] Pazinato E., Baptista C.S., Miranda R.A.V., *GeoLocalizador: um Sistema de Referência Espaço-Temporal Indireta utilizando um SGBD Objeto-Relacional*. In proceedings: Brazilian Symposium on GeoInformatics. SBC, 2002

[PG06] Paul, M., Ghosh, S., K., *An approach for service oriented discovery and retrieval of spatial data*, International Conference on Software Engineering In Proceedings of the 2006 international workshop on Service-oriented software engineering, ACM Press, 2006, 88-94

[Per03] Percivall, G. *Opengis reference model*, 2003.

[Pro07] Protegé. <<http://protege.stanford.edu/>> Acessado em 2007.

[RW01] Rajabifard, D. A., and Williamson, P. I. *Spatial data infrastructures: Concept, sdi hierarchy and future directions*. GEOMATICS'80 Conference, Tehran, Iran (2001).

[RDF07] *Resource Description Framework*: <<http://www.w3.org/RDF/>> Acessado em 15 mai. 2007.

[SBH06] Shadbolt ,N., Berners-Lee , T., Z. G., Hall, W. *The Semantic Web Revisited*. IEEE Intelligent Systems 21, 3 (2006), 96-101.

[Sil07] Silva, E., R.. Uma arquitetura orientada a serviços para roteamento personalizado. 2007. 103f. Dissertação (Mestrado) - Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática. Campina Grande, Paraíba. 2006

- [SiB04] Sirin, E., Bijan P., *The OWL-S Java API*, Poster, Third International Semantic Web Conference (ISWC2004), Hiroshima, Japan, November 2004.
- [SVS04] Sivashanmugam, K., Verma, K., and Sheth, A. *Discovery of web services in a federated registry environment*. IEEE International Conference on Web Services (ICWS'04) (2004), 270-282.
- [SF07] Smits, P., and Friis-Christensen, A., *Resource Discovery in a European Spatial Data Infrastructure*, IEEE Trans. Knowl. Data Eng, 19 (1),2007, 85-95
- [Son04] SONNET, J. *Ows 2 common architecture: Wsdl soap uddi*,2004.
- [Tai05] Tait, M. G. *Implementing geoportals: applications of distributed gis*. Computers, Environment and Urban Systems 29, 1 (January 2005), 33-47.
- [TNM07] *The National Map*. Disponível em <<http://nationalmap.gov/>> Acessado em maio de 2007
- [Ukg07] *UK Gigateway - Centre for Ecology and Hydrology*. Disponível em <<http://www.gigateway.org.uk/>> Acessado em maio de 2007
- [VBR05] Venugopal, S., Buyya, R., and Ramamohanarao, K. *A taxonomy of data grids for distributed data sharing, management and processing*. Tech. rep., havard, 2005.
- [VS06] Viegas, R., Soares, V., *Querying a Geographic Database using an Ontology-Based Methodology*, Brazilian Symposium on GeoInformatics - GEOINFO 2006
- [W3C05] *W3c. web services architecture requirements*. Disponível em <<http://www.w3c.org/TR/wsa-reqs>>. Acessado em 10 dez. 2005.
- [Wyo07] *Wyoming Natural Resources Data Clearinghouse*. Disponível em: <<http://www.sdvc.uwyo.edu/clearinghouse/>>. Acessado em 15 mai. 2007.
- [XGK+06] Xiujun, M., Gang, L., Kunqing, X., Meng, S., *A peer-to-peer approach to Geospatial Web Services discovery*, Proceedings of the 1st international conference on Scalable information systems, ACM Press, 2006

[Xqu05] *Xquery. xml query language.* Disponível em <<http://www.w3.org/TR/xquery/>>. [Acessado em 10 dez. 2005.].

[ZHS+04] Zhao, B. Y., Huang, L., Stribling, J., Rhea, S. C., Joseph, A. D., and Kubiawicz, J. D. *Tapestry: A resilient global-scale overlay for service deployment.* IEEE Journal on Selected Areas in Communications 22, 1 (2004), 41-53.

[ZCL+04] Zhao, P., Chen, A., Liu, Y., Di, L., Yang, W., and Li, P. *Grid metadata catalog service-based ogc web registry service.* In GIS '04: Proceedings of the 12th annual ACM international workshop on Geographic information systems (New York, NY, USA, 2004), ACM Press, pp. 22-30.

[ZWW06] Zimmermann, R., Wei-Shinn, K., Wang, H., Zand, A., Bardet, J., *A Distributed Geotechnical Information Management and Exchange Architecture,* Internet Computing (setembro/outubro 2006), 10(5), 26-33