



UFPE - UNIVERSIDADE FEDERAL DE PERNAMBUCO
CIN - CENTRO DE INFORMÁTICA
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

JOEL DA SILVA

**“INTEGRANDO SERVIÇOS ANALÍTICOS E
GEOGRÁFICOS PARA SUPORTE À DECISÃO NA WEB”**

ESTE TRABALHO FOI APRESENTADO À PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO DO CENTRO DE INFORMÁTICA DA UNIVERSIDADE FEDERAL DE PERNAMBUCO COMO REQUISITO PARCIAL PARA OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIA DA COMPUTAÇÃO.

ORIENTADORA: Dr^a. VALÉRIA C. TIMES
CO-ORIENTADOR: Dr. ROBERTO S. M. DE BARROS

RECIFE, MARÇO DE 2004.

Agradecimentos

Primeiramente, agradeço a Deus por tudo que tenho alcançado, pela saúde e força que Ele tem me concedido para que eu lute em prol de meus ideais. Gostaria de agradecer à minha família, e de uma forma especial, à Mara, pelo amor, carinho e apoio proporcionado durante este período em que estive cursando o mestrado.

Gostaria de agradecer aos meus orientadores, Prof.^a Valéria Times e o Prof. Roberto Barros, pela confiança depositada. Aos colegas do Projeto GOLAPA, Robson, Juliano e Viviane, pelas produtivas discussões, as quais foram de grande utilidade durante o processo de desenvolvimento desta dissertação.

Não poderia faltar o profundo agradecimento aos meus irmãos Cleber e Tarcísio, pelo companheirismo e ambiente familiar proporcionado durante estes dois anos de caminhada.

INTEGRANDO SERVIÇOS ANALÍTICOS E GEOGRÁFICOS PARA SUPORTE À DECISÃO NA WEB¹

Autor: Joel da Silva
Orientadora: Prof^ª Dr^ª. Valéria C. Times
Co-Orientador: Prof. Dr. Roberto S. M. de Barros

Resumo

Atualmente, as organizações procuram dispor de um sistema computacional que auxilie na tomada de decisões estratégicas, a fim de agregar vantagens a seus negócios. Entretanto, grande parte dessas organizações possui ferramentas distintas para o tratamento de dados de natureza analítica e geográfica. Isso muitas vezes acarreta um grande desperdício de tempo e processamento para realizar consultas que englobem, simultaneamente, dados de ambas as naturezas. Várias pesquisas têm sido desenvolvidas para integrar as funcionalidades de análise disponíveis nos sistemas de processamento analítico e geográfico. O objetivo principal dessas pesquisas é proporcionar aos usuários um sistema capaz de processar simultaneamente dados geográficos e multidimensionais, abstraindo a complexidade existente em consultar e analisar estes dados separadamente em um processo de tomada de decisões. Porém, a maioria destas pesquisas não satisfaz alguns requisitos importantes como independência de plataforma e provisão de funcionalidades integradas. É neste contexto que se insere o presente trabalho, que propõe o modelo ISAG para integração de serviços analíticos e geográficos para suporte à decisão na Web. Este modelo foi implementado por um serviço chamado GMLA Web Service, o qual é fortemente baseado em tecnologias padronizadas como Web Services, Java e XML. Os resultados das consultas processadas pelo GMLA WS são visualizados graficamente em um browser Web por meio de tabelas e/ou mapas, auxiliando os usuários em seu processo de tomada de decisões.

Palavras-chave: Integração de Serviços Analíticos e Geográficos, Suporte à Decisão, *GMLA Web Service*, Modelo ISAG.

¹ Dissertação de Mestrado em Ciência da Computação, Centro de Informática, Universidade Federal de Pernambuco, Recife, PE, Março de 2004.

INTEGRATING ANALYTIC AND GEOGRAPHICAL SERVICES FOR DECISION SUPPORT OVER THE WEB²

Author: Joel da Silva

Advisor: Prof^a Dr^a. Valéria C. Times

Co-Advisor: Prof. Dr. Roberto S. M. de Barros

Abstract

Nowadays, several organizations have a need for using a computer system supporting strategic decision making activities, in order to aggregate some advantages to their business. However, many of these organizations may still need to use distinct tools for analytic and geographical data processing. This may result in a waste of both time and processing while performing queries that simultaneously contain these two kinds of data. A lot of research has been developed for integrating the analysis functionality that is available in analytic and geographic processing systems. The main goal of these studies is to provide users with a system capable of processing both geographic and multidimensional data by abstracting the complexity of separately querying and analyzing these data in a decision making process. However, most of them do not satisfy some important requirements such as platform independence and integrated functionality. This work presents a services integration model, namely ISAG, for supporting analytic and/or geographic requests over the Web. This model has been implemented by a Web Service, called as the GMLA WS, which is strongly based on standardized technologies such as Web Services, Java and XML. The GMLA WS query results are displayed by means of a Web browser as maps and/or tables for helping users in their decision making processes.

Keywords: Analytic and Geographical Services Integration, Decision Support, GMLA Web Service, ISAG Model.

² Master of Science Dissertation in Computer Science, Informatics Center, Federal University of Pernambuco, Recife, PE, March, 2004.

Índice

1. Introdução	1
1.1 Motivação.....	2
1.2 Objetivos	3
1.3 Estrutura da Dissertação.....	4
2. Suporte à Decisão	6
2.1 Introdução.....	7
2.2 Sistemas de Suporte à Decisão	8
2.2.1 Categorização dos SSD	8
2.3 DW - <i>Data Warehouse</i>	10
2.4 OLAP - <i>On-Line Analytical Processing</i>	13
2.4.1 Consultas Multidimensionais com Operadores OLAP	15
2.5 SIG - Sistemas de Informações Geográficas.....	17
2.6 DWG - <i>Data Warehouse Geográfico</i>	20
2.7 Considerações Finais	22
3. Trabalhos Relacionados e Contextualização	23
3.1 Introdução.....	24
3.2 Propostas para Processamento Geográfico-Multidimensional.....	25
3.3 Contextualização da Dissertação	28
3.3.1 Arquitetura GOLAPA	29
3.3.1.1 Camadas e Componentes de GOLAPA.....	30
3.3.1.2 GMLA - <i>GML For Analysis</i>	33
3.3.1.3 O Metamodelo GAM e o Esquema GeoMD	35
3.3.1.4 GeoDWFrame	38
3.4 Considerações Finais	39
4. XML e Serviços Web.....	42
4.1 Introdução.....	43

4.2 XML e Padrões Relacionados	44
4.2.1 A Linguagem XML.....	44
4.2.2 XML Schema.....	45
4.2.3 XSLT - XSL Transformations.....	48
4.2.4 GML - Geography Markup Language.....	49
4.2.5 SVG - Scalable Vector Graphics.....	50
4.3 Web Services.....	52
4.3.1 Implementação de Web Services.....	52
4.3.1.1 XML.....	53
4.3.1.2 SOAP.....	53
4.3.1.3 UDDI.....	53
4.3.1.4 WSDL.....	54
4.3.2 Arquitetura dos Web Services.....	55
4.4 XMLA - XML For Analysis	56
4.4.1 Implementação do XML For Analysis.....	57
4.4.1.1 Método Discover.....	57
4.4.1.2 Método Execute.....	59
4.5 WFS - Web Feature Service.....	60
4.5.1 Arquitetura do Web Feature Service.....	61
4.5.2 Operações do Web Feature Service	62
4.5.2.1 GetCapabilities	62
4.5.2.2 DescribeFeatureType	63
4.5.2.3 GetFeature	63
4.5.2.4 Transaction	63
4.5.2.5 LockFeature	64
4.5.3 Classes de WFS	65
4.5.4 Expressões de Filtro.....	65
4.5.4.1 Tipos de Operadores.....	66
4.6 Considerações Finais	70
5. Modelo ISAG.....	72
5.1 Introdução.....	73
5.2 Estudo das Especificações Utilizadas	74
5.3 GMLA Request Schema.....	77
5.4 O Modelo de Integração ISAG.....	83
5.5 Considerações Finais	92

6. GMLA Web Service	94
6.1 Introdução.....	95
6.2 Arquitetura do <i>GMLA Web Service</i>	96
6.3 Implementação do Modelo ISAG.....	98
6.4 A Implementação das Especificações XMLA e WFS	102
6.4.1 Especificação XMLA.....	102
6.4.2 Especificação WFS	102
6.5 Fonte de Metadados.....	103
6.6 A Base de Dados Analíticos e Geográficos	104
6.7 O Cliente GMLA	106
6.8 Tipos de Requisições Processadas pelo GMLA WS	109
6.8.1 Requisições do Tipo MD.....	109
6.8.2 Requisições do Tipo GEO.....	113
6.8.3 Requisições do Tipo GEOMD.....	116
6.9 Considerações Finais	122
7. Conclusões e Trabalhos Futuros	123
7.1 Considerações Finais	124
7.2 Contribuições.....	125
7.3 Trabalhos Futuros	126
Referências Bibliográficas	123

Lista de Figuras

Figura 2.1: Os cinco estágios da utilização do <i>Data Warehouse</i> no suporte à decisão.....	12
Figura 2.2. Metáfora Visual do cubo de dados <i>Vendas</i> (adaptado de [Fid03]).	15
Figura 2.3: Representação dos operadores OLAP [Ril03].	16
Figura 2.4: Exemplo de Sobreposição de Temas (Extraído de [Fid03]).	18
Figura 2.5: Componentes de um Sistema de Informações Geográficas (Adaptado de [CCH+96]).....	20
Figura 3.1: Arquitetura GOLAPA [Fid03].	30
Figura 3.2: Diagrama do <i>GMLA Schema</i>	34
Figura 3.3: O Esquema Abstrato GeoMD [Fid03]......	36
Figura 3.4: Modelo de Implementação do Metamodelo GeoMDM [Fid03].	37
Figura 3.5: Exemplo de um esquema estrela compartilhando o mesmo contexto espacial.....	38
Figura 4.1: Exemplo de <i>XML Schema</i>	46
Figura 4.2: Representação gráfica do Esquema apresentado na figura 4.1.	47
Figura 4.3: Documento XML que obedece a estrutura definida pelo <i>XML Schema</i> apresentado na figura 4.1.	47
Figura 4.4: Componentes de um processo de transformação XSLT.....	49
Figura 4.5: Exemplo da utilização de propriedades geométricas em GML.	50
Figura 4.6: Exemplo da utilização de SVG para desenhar um mapa.	51
Figura 4.7: Arquitetura de um <i>Web Service</i> , seus componentes, operações e artefatos (adaptado de [WSCA01]).....	55
Figura 4.8: Arquitetura de uma possível aplicação baseada no padrão <i>XML for Analysis</i> (adaptado de [XMLA02]).	60
Figura 4.9: Implementação de um <i>Web Feature Service</i> (Adaptado de [WFS02]).	61
Figura 4.10: Possíveis mensagens trocadas entre uma aplicação cliente e o WFS (Adaptado de [WFS02]).....	64
Figura 4.11: Fragmento do XML Schema que define o elemento <i><Filter></i>	66
Figura 4.12: Estrutura do elemento <i><Filter></i>	67

Figura 4.13: Requisição <i>GetFeature</i> com filtro espacial <i>BBox</i> .	69
Figura 4.14: Requisição <i>GetFeature</i> combinando o operador espacial <i>BBox</i> e o operador de comparação <i>PropertyIsLessThan</i> .	70
Figura 5.1: Correspondências entre os serviços WFS e XMLA.	74
Figura 5.2: Estrutura do <i>GMLA Request Schema</i> .	79
Figura 5.3: Modelo de Integração de Serviços Analíticos e Geográficos (ISAG).	84
Figura 5.4: Expansão da atividade <i>Extrair Parâmetros da Requisição</i> .	86
Figura 5.5: Expansão da atividade <i>Validar Parâmetros da Requisição</i> .	87
Figura 5.6: Expansão da atividade <i>Consultar Serviço MD</i> .	88
Figura 5.7: Expansão da atividade <i>Consultar Serviço GEO</i> .	89
Figura 5.8: Expansão da atividade <i>Consultar Metadados</i> .	90
Figura 5.9: Expansão da atividade <i>Integrar Resultados</i> .	91
Figura 6.1: Arquitetura do GMLA WS.	97
Figura 6.2: Diagrama de Classes do GMLA WS.	99
Figura 6.3: Estrutura da base de dados <i>FoodMart 2000</i> .	105
Figura 6.4: Interface do Cliente GMLA.	107
Figura 6.5: Exemplo de comando MDX.	110
Figura 6.6: Exemplo de requisição MD.	110
Figura 6.7: Resultado gráfico da consulta MD expressa na figura 6.6.	111
Figura 6.8: Exemplo de comando MDX para listar as dez primeiras lojas com maior valor de vendas.	112
Figura 6.9: Resultado gráfico da consulta apresentada na figura 6.8.	112
Figura 6.10: Exemplo de consulta geográfica (GEO).	113
Figura 6.11: Feições geográficas resultantes da consulta GEO apresentada na figura 6.10.	114
Figura 6.12: Consulta GEO para recuperar os temas País, Estados e Cidades.	115
Figura 6.13: Resultado gráfico da requisição apresentada na figura 6.12.	116
Figura 6.14: Exemplo de requisição GEOMD de mapeamento.	117
Figura 6.15: Visualização gráfica do resultado de uma consulta GEOMD de mapeamento.	118
Figura 6.16: Exemplo de requisição GEOMD de Integração.	120
Figura 6.17: Visualização gráfica do resultado de uma consulta GEOMD de Integração.	121

Lista de Tabelas

Tabela 2.1: Principais diferenças entre os ambientes operacionais e de suporte à decisão.	11
Tabela 3.1: Comparação entre as propostas para processamento geográfico-multidimensional..	40
Tabela 5.1: Estrutura do Elemento <i>MD</i>	80
Tabela 5.2: Estrutura do Elemento <i>GEO</i>	81
Tabela 5.3: Estrutura do elemento <i>GEOMD</i>	82
Tabela 6.1: Correspondências entre GOLAPA e a arquitetura do GMLA WS.	96
Tabela 6.2: Barra de ferramentas da interface do Cliente GMLA.	108

Principais Abreviaturas

<i>API</i>	<i>Application Program Interface</i>
<i>CDB</i>	<i>Conventional Database</i>
<i>CORBA</i>	<i>Common Object Request Broker Architecture</i>
<i>CWM</i>	<i>Common Warehouse Metamodel</i>
<i>DTD</i>	<i>Document Type Definitions</i>
<i>DTS</i>	<i>Data Transformation Services</i>
<i>DW</i>	<i>Data Warehouse</i>
<i>DWG</i>	<i>DW Geográfico</i>
<i>ETL</i>	<i>Extract Transform Load</i>
<i>FME</i>	<i>Feature Manipulation Engine</i>
<i>GAM</i>	<i>Geographical and Analytical Metamodel</i>
<i>GDB</i>	<i>Geographical Database</i>
<i>GDM</i>	<i>Geographical Data Mart</i>
<i>GDW</i>	<i>Geographical DW</i>
<i>GeoMDM</i>	<i>Geographical Multidimensional Metamodel</i>
<i>GeoDWFrame</i>	<i>Geographical DW Frame</i>
<i>GETL</i>	<i>Geographical ETL</i>
<i>GIS</i>	<i>Geographic Information System</i>
<i>GML</i>	<i>Geography Markup Language</i>
<i>GMLA</i>	<i>GML for Analysis</i>
<i>GMLA WS</i>	<i>GMLA Web Service</i>
<i>GOAL</i>	<i>Geographical Information On-Line Analysis</i>
<i>GOLAP</i>	<i>Geographical On-Line Analytical Processing</i>
<i>GOLAPA</i>	<i>Geographical Online Analytical Processing Architecture</i>
<i>GOLAPE</i>	<i>GOLAP Engine</i>
<i>GOLAPI</i>	<i>GOLAP Interface</i>

<i>HTTP</i>	<i>Hypertext Transfer Protocol</i>
<i>IDL</i>	<i>Interface Definition Language</i>
<i>JMI</i>	<i>Java Metadata Interface</i>
<i>JSP</i>	<i>Java Specification Requests</i>
<i>MDAPI</i>	<i>Multi-Dimensional Application Program Interface</i>
<i>MDC</i>	<i>Meta Data Coalition</i>
<i>MDR</i>	<i>MetaData Repository</i>
<i>MDX</i>	<i>Multidimensional Expressions</i>
<i>MDXML</i>	<i>MDX Markup Language</i>
<i>MER</i>	<i>Modelo de Entidades e Relacionamentos</i>
<i>MI</i>	<i>Módulo de Integração</i>
<i>MOF</i>	<i>Meta Object Facility</i>
<i>MOLAP</i>	<i>Multidimensional OLAP</i>
<i>OCL</i>	<i>Object Constraint Language</i>
<i>ODS</i>	<i>Operational Data Store</i>
<i>ODSG</i>	<i>Geographical Operational Data Store</i>
<i>OGC</i>	<i>Open GIS Consortium</i>
<i>OLAP</i>	<i>On-Line Analytical Processing</i>
<i>OLTP</i>	<i>On-Line Transaction Processing</i>
<i>OMG</i>	<i>Object Management Group</i>
<i>SBGD</i>	<i>Sistemas de Gerenciamento de Banco de Dados</i>
<i>SGBDR</i>	<i>Sistema de Gerenciamento de Bancos de Dados Relacionais</i>
<i>SIG</i>	<i>Sistema de Informações Geográficas</i>
<i>SOAP</i>	<i>Simple Object Application Protocol</i>
<i>SQL</i>	<i>Structured Query Language</i>
<i>SVG</i>	<i>Scalable Vector Graphics</i>
<i>UDDI</i>	<i>Universal Description, Discovery and Integration</i>
<i>UML</i>	<i>Unified Modeling Language</i>
<i>VML</i>	<i>Vector Markup Language</i>
<i>W3C</i>	<i>World Wide Web Consortium</i>
<i>WFS</i>	<i>Web Feature Service</i>
<i>WSDL</i>	<i>Web Services Definition Language</i>
<i>XMI</i>	<i>XML Metadata Interchange</i>
<i>XML</i>	<i>eXtensible Markup Language</i>
<i>XMLA</i>	<i>XML for Analysis</i>

Capítulo 1

Introdução

Este capítulo relata as principais motivações para realização deste trabalho, lista os objetivos de pesquisa almejados, e, finalmente, mostra como está estruturado o restante da presente dissertação.

1.1 Motivação

Atualmente, diversos membros da Comunidade Científica em Tecnologia da Informação, vêm direcionando seus esforços na tentativa de solucionar o problema de integração entre processamento analítico e geográfico [FCT01, Fer02, Goa03, Gon99, HKS97, MS00, KMM00, KMM01, Kim01, SLT+00a]. Esta é uma tarefa árdua e merece uma atenção especial. Entretanto, as propostas encontradas atualmente estão atreladas a tecnologias proprietárias ou não satisfazem totalmente todos os aspectos da integração. O principal objetivo almejado é o provimento de um ambiente único, aberto e extensível com as capacidades de análise disponibilizadas por estas duas tecnologias, de forma que um sistema de processamento geográfico usufrua das facilidades implementadas por uma ferramenta analítica, enquanto esta receba uma contribuição considerável ao agregar tratamento específico à dimensão geográfica.

Seguindo esta mesma linha de raciocínio, o projeto GOLAPA (*Geographical Online Analytical Processing Architecture*) [FTS01, GHP03], do Centro de Informática da Universidade Federal de Pernambuco, no qual este trabalho está inserido, tem estudado formas de prover ao usuário, uma abstração da complexidade normalmente agregada às atividades que têm como objetivo efetuar consultas sobre dados analíticos e geográficos para a tomada de decisão. GOLAPA é uma arquitetura de *software* que visa ser: 1) baseada em tecnologias abertas e extensíveis e 2) otimizada para suporte à decisão em um contexto multidimensional geográfico. O primeiro objetivo trata da integração sobre o aspecto tecnológico, isto é, que sua implementação seja baseada em tecnologias como *Java* e *XML*. Já o segundo, discorre sobre a estratégia que é potencialmente a melhor solução para a integração no processamento dos referidos dados.

Muitas vezes, as organizações já dispõem de bases de dados analíticos e geográficos, que permitem realizar análises sobre o comportamento de seu negócio. Entretanto, como relatado em [Fer02], em muitos casos, existe uma grande complexidade agregada neste tipo de análise. Geralmente, para realizar uma consulta envolvendo os dois tipos de dados, os analistas primeiro requisitam um serviço depois o outro, para que as informações possam ser cruzadas e, finalmente o resultado esperado possa ser produzido. Isto motiva ainda mais, a realização de pesquisas que objetivam o provimento de um ambiente único, que permita ao usuário, realizar consultas analíticas e geográficas sem levar em consideração a complexidade envolvida.

1.2 Objetivos

O principal objetivo da presente pesquisa é a definição de um modelo de integração de serviços analíticos e geográficos para suporte à decisão na *Web*. Atualmente, dispomos de algumas especificações que visam estabelecer um padrão global para processamento analítico e geográfico na *Web*. Trata-se do *XML For Analysis* para processamento analítico, e o *Web Feature Service* para processamento geográfico. Em uma primeira instância, se faz necessário realizar um aprofundado estudo destas duas especificações, com o intuito de definir as possíveis correspondências existentes entre elas, para que, em um segundo momento, seja proposto um modelo que integre, em um único ambiente acessível via Internet, todas as vantagens disponibilizadas pelo processamento analítico bem como as facilidades oferecidas pelo processamento geográfico no tratamento de dados para auxílio à tomada de decisões. Dessa forma, a dificuldade existente no processo de acessar dois serviços separadamente, seria abstraída.

Após a definição do modelo de integração, tem-se como meta a implementação de um serviço *Web* que valide o modelo proposto. O serviço implementado será caracterizado como o mecanismo de integração responsável pela disponibilização de operações que permitam o envio de requisições para consulta a dados em uma base de dados analíticos e geográficos, podendo usufruir, de forma simultânea, do poder de processamento disponibilizado pelos serviços analítico e geográfico. A intenção é que, na implementação do mecanismo de integração, sejam utilizadas somente tecnologias abertas e extensíveis, como por exemplo, XML [XML03a], JAVA [JAV03], GML [GML03a], *Filter Encoding* [FEIS03], *Web Services* [WSA02], *Web Feature Service* [WFS02], *XML For Analysis* [XMLA02, XMLA03], SVG [SVG03], MOF [MOF02] e XMI [XMI02].

Caracteriza-se também como um objetivo desta pesquisa a implementação de uma aplicação cliente, que tenha capacidade de demonstrar graficamente os dados analíticos e geográficos resultantes das consultas feitas a partir do mecanismo de integração, podendo assim, validar as idéias propostas juntamente com o modelo de integração. Salienta-se que a responsabilidade de implementação de uma interface gráfica bem mais elaborada é atribuída a uma outra dissertação de mestrado ainda em andamento.

1.3 Estrutura da Dissertação

Para alcançar os objetivos almejados, bem como para apresentar alguns conceitos e tecnologias relacionadas a esta dissertação, o restante do presente documento está organizado da seguinte forma:

Capítulo 2 – Suporte à Decisão: Este capítulo apresenta conceitos relacionados aos Sistemas de Suporte à Decisão. Para isso, serão listados tópicos referentes à *Data Warehouse* e OLAP, Sistemas de Informações Geográficas e *Data Warehouse* Geográfico.

Capítulo 3 – Trabalhos Relacionados e Contextualização: No terceiro capítulo são citadas algumas pesquisas que se relacionam com a presente dissertação e também é apresentada a contextualização da mesma. Neste sentido, será apresentada a Arquitetura GOLAPA, devido a este trabalho estar voltado para um de seus componentes. Serão descritos também, outros trabalhos considerados importantes para o contexto da integração de serviços analíticos e geográficos para suporte à decisão na *Web*.

Capítulo 4 – XML e Serviços Web: Neste capítulo, são descritos alguns conceitos relativos a padrões como o XML, *XML Schema*, XSLT, GML, *Web Services*, *XML For Analysis* e *Web Feature Service*, entre outros. Estas tecnologias são consideradas os principais elementos usados na integração de serviços analíticos e geográficos na *Web*.

Capítulo 5 – Modelo ISAG: No quinto capítulo é proposto um modelo de integração de serviços analíticos e geográficos para suporte à decisão na *Web*. Este modelo se aplica à arquitetura GOLAPA (apresentada no capítulo 3), estando voltado, mais especificamente, para o componente GOLAPE da referida arquitetura. Neste capítulo também é apresentado o *GMLA Request Schema*, o qual é responsável pela definição dos tipos de requisições que poderão ser enviadas ao mecanismo de integração de serviços proposto.

Capítulo 6 – GMLA Web Service: Neste capítulo, serão descritos alguns detalhes referentes à implementação do modelo ISAG. A implementação resultou em um serviço *Web* denominado *GMLA Web Service*, que disponibiliza operações para realização de consultas a dados analíticos e/ou geográficos.

Capítulo 7 – Conclusões e Trabalhos Futuros: Finalmente, neste capítulo são efetuadas algumas considerações finais sobre os principais tópicos abordados nesta dissertação, apresentando as contribuições alcançadas e indicações para trabalhos futuros.

Capítulo 2

Suporte à Decisão

O objetivo deste capítulo é apresentar alguns conceitos relacionados aos Sistemas de Suporte à Decisão. Para isso, serão listados tópicos referentes à *Data Warehouse* e OLAP, Sistemas de Informações Geográficas e *Data Warehouse* Geográfico. Estas são tecnologias que exercem importantes funções no contexto de tomada de decisões estratégicas.

2.1 Introdução

Conceitos referentes a SIG, *Data Warehouse* e OLAP estão cada vez mais sendo interligados [FCT01, Fer02, Goa03, Gon99, HKS97, MS00, Kim01, KMM00, KMM01, SLT+00a]. Organizações procuram dispor de técnicas pertinentes a estas tecnologias, com a intenção de alcançar um ambiente de suporte à tomada de decisões robusto e eficiente. Tendo à disposição um ambiente com capacidades para processamento, tanto de dados analíticos quanto de dados espaciais, a capacidade de auxílio à tomada de decisões aumenta consideravelmente, pois conta com duas poderosas técnicas de análise de dados, as quais consideram, simultaneamente, aspectos pertinentes a natureza dos dados analíticos e também a dos dados geográficos.

Como essa dissertação trata da integração de serviços analíticos e geográficos para suporte à decisão na *Web*, neste capítulo serão listados alguns conceitos referentes às tecnologias de SIG, OLAP, *Data Warehouse* e *Data Warehouse* Geográfico. Estas tecnologias vêm dando suporte e incrementando cada vez mais o auxílio à tomada de decisões, o que possibilita aos responsáveis pelas organizações, decidirem qual a melhor estratégia a adotar, conduzindo seu negócio de forma mais adequada e eficiente.

Este capítulo está organizado da seguinte maneira: Na seção 2.2, são apresentados conceitos relativos aos Sistemas de Suporte à Decisão. A seção 2.3 discorre sobre *Data Warehouse*. Na sequência, na seção 2.4, são apresentados conceitos pertinentes à tecnologia OLAP. A seção 2.5 descreve alguns conceitos dos Sistemas de Informações Geográficas. Por sua vez, a seção 2.6 relata sobre *Data Warehouse* Geográfico. Finalmente, na seção 2.7, são apresentadas considerações finais referentes ao presente capítulo.

2.2 Sistemas de Suporte à Decisão

Os Sistemas de Suporte à Decisão (SSD) [Al80, CD96, Mal00] possibilitam o gerenciamento e a análise, de grandes bases de dados de forma eficiente e consistente, permitindo a extração de informações que auxiliem a compreender o comportamento do negócio de uma organização. Essas informações possibilitam que os diretores executivos de uma organização tomem decisões estratégicas no intuito de melhor conduzir seu negócio em meio às freqüentes oscilações do mercado. A partir dessas análises, podem ser feitas mudanças em estratégias de *marketing*, e inserção de novos produtos ou serviços no mercado. Finalmente, o principal objetivo dos Sistemas de Suporte à Decisão é fornecer informações relevantes para o processo de gerenciamento dos negócios de uma organização.

Os SSD diferenciam-se dos sistemas de suporte operacional pelo fato de que, enquanto os modelos de dados dos sistemas para suporte a decisão são orientados por assunto (modelo estrela ou flocos de neve [Inm97, Kim96]), os modelos de dados dos sistemas operacionais são orientados por transações (modelo relacional [EN99]) [Kim96]. Assim, para implementar um SSD eficiente, faz-se necessário dividir a arquitetura de dados da organização em dois ambientes de banco de dados [Kim96]: (1) um específico para dados operacionais - estes bancos por via de regra já existem e têm aplicação e muitas operações já pré-definidas e (2) outro para suporte à decisão – estes bancos normalmente precisam ser construídos a partir dos dados existentes nas bases de dados da organização.

2.2.1 Categorização dos SSD

Em seus trabalhos, Power [Pow98, Pow02] define os Sistemas de Suporte à Decisão como sendo sistemas e subsistemas interativos e baseados em computador, destinados a auxiliar os tomadores de decisões a utilizar dados, documentos, conhecimento e/ou modelos para identificar e solucionar problemas. De acordo com essa definição, em [Pow00], o autor discorre sobre cinco categorias de SSD, a saber: (1) SSD dirigidos a dados, (2) SSD dirigidos a modelos, (3) SSD dirigidos a conhecimento, (4) SSD dirigidos a documentos e (5) SSD dirigidos a comunicação. O autor justifica esta categorização devido à grande quantidade de sistemas utilizados para a tomada de decisões, e às formas com que estes são utilizados. O objetivo da categorização apresentada é auxiliar na compreensão de como integrar, avaliar e selecionar um meio apropriado para auxiliar na tomada de decisões.

Os SSD dirigidos a dados são aqueles que incluem sistemas de gerenciamento de relatórios, *Data Warehouse* [CF03, Inm97] e OLAP (*On-Line Analytical Processing*) [CD96, Tho97], Sistemas de Informações Executivas [BM03], Sistemas Espaciais de Suporte à Decisão [SM03] e Sistemas de Inteligência de Negócios [Har03]. Essa categoria é voltada para a manipulação de grandes bases de dados referentes ao negócio da organização e incluindo algumas vezes dados externos à organização.

Na categoria dos SSD dirigidos a modelos, estão aqueles sistemas que utilizam modelos financeiros e contábeis, modelos representacionais e de otimização. Estes SSD utilizam dados e parâmetros fornecidos pelos gerentes executivos para analisar a situação. Normalmente, estes tipos de sistemas não requerem grandes bases de dados.

Por sua vez, os SSD dirigidos a conhecimento são aqueles que têm a capacidade de sugerir ou recomendar ações para os gerentes. Estes sistemas envolvem programas de computador e pessoas com alto grau de experiência na resolução de problemas. Essa experiência consiste do conhecimento sobre o domínio em questão, conhecimento dos problemas referentes a este domínio, e habilidade de resolver alguns desses problemas. O conceito de *Data Mining* [HK00, NFC01] é bastante aplicado em sistemas dessa natureza.

Os SSD dirigidos a documentos são direcionados a auxiliar os tomadores de decisões na recuperação e gerenciamento de documentos não estruturados e páginas *Web*. Estes sistemas integram uma grande variedade de sistemas para a recuperação e análise de documentos. Exemplos de documentos que podem ser acessados por essa classe de sistemas são especificações de produtos, catálogos, documentos históricos da corporação, e correspondências importantes. A associação de um engenho de buscas a estes sistemas pode ser visto como uma poderosa ferramenta no auxílio à tomada de decisões.

A quinta categoria de Sistemas de Suporte à Decisão, denominada SSD dirigido a comunicação, integra sistemas baseados em computador para auxiliar os seus usuários a trabalharem juntos, de forma colaborativa, mesmo estando em locais diferentes. Isso envolve programas que possibilitam comunicação eletrônica, gerenciadores de *e-mail*, compartilhamento de arquivos, vídeo conferência, entre outros.

Obviamente, as organizações com recursos humanos e financeiros capazes de englobar características pertinentes a mais de uma das categorias listadas acima, têm a sua disposição um poderoso ambiente de suporte à decisão, e conseqüentemente dispõem de um diferencial favorável no mercado competitivo.

2.3 DW - Data Warehouse

Uma das tecnologias amplamente utilizadas no contexto de Sistemas de Suporte à Decisão é *Data Warehouse* (DW) [CD96, CF03, Inm97, Kim96]. Podem ser encontradas na literatura diferentes definições para o termo *Data Warehouse*. Porém, a definição criada por Inmon [Inm97], e dada abaixo foi amplamente difundida:

“Data Warehouse é uma coleção de dados orientada a assunto, integrada, não-volátil, e variante no tempo em suporte à decisões gerenciais”.

Seguindo a definição acima, podemos dizer que um DW é uma base de dados cujas principais características são [Inm97]:

1. ***Ser orientado ao assunto***, o que contrasta com a abordagem convencional de suporte operacional. Representa a migração para o suporte à decisão, onde o objetivo é tomar decisões sobre o "assunto" - tema dos dados armazenados. Por exemplo, os dados de venda em um sistema orientado a transações (operacional), contêm informações sobre a venda de produtos específicos para certos clientes. Dados de venda no ambiente de suporte à decisão contêm um histórico das vendas considerando período, local, produto. Quando bem especificado, este último conjunto de dados expressa a "natureza" das informações nele contidas;
2. ***Ser perfeitamente integrado***, pois deve consolidar dados de diferentes origens (e.g. sistemas legados e sistemas convencionais voltados ao processamento de transações), o que freqüentemente envolve compatibilizar codificações, unidades e medidas;

3. **Considerar a variação temporal**, uma vez que os sistemas orientados a transação capturam dados válidos em um instante de tempo, no momento do acesso. Em seguida, os dados podem ser alterados e perder sua informação "histórica". Quando se tem uma dimensão temporal, o dado é associado a um ponto no tempo. Diferentes dados podem ser comparados ao longo do eixo temporal, seja para expressarem informação coincidente no tempo ou para relacionarem um histórico da informação ao longo de um período;
4. **Ser não volátil**, pois as informações já presentes no banco raramente são modificadas. Os novos dados são absorvidos pelo banco, integrando-se com as informações previamente armazenadas.

As características apresentadas anteriormente, capacitam a tecnologia de DW como uma das mais promissoras para oferecer suporte à tomada de decisão e a distingue, consideravelmente, das tecnologias transacionais. Na tabela 2.1, podemos visualizar as principais diferenças entre ambientes operacionais e de suporte a decisão [Fid03].

Tabela 2.1: Principais diferenças entre os ambientes operacionais e de suporte à decisão.

Características	Operacional	Decisório
Objetivo	Operações diárias do negócio	Análise do negócio
Visão dos Dados	Relacional, Objeto ou Objeto-Relacional	Dimensional
Operações com os Dados	Inclusão, Alteração, Exclusão e Consulta	Carga e Consulta
Atualização	Contínua (tempo real)	Temporal (lote)
Número de Usuários	Milhares	Dezenas
Tipo de Usuário	Operacional	Gerencial
Tipos de Consultas	Pré-definida (predominante) e ad-hoc	Pré-definida e ad-hoc
Granularidade dos Dados	Detalhados	Detalhados e Resumidos
Redundância dos Dados	Ocorrência minimizada	Ocorrência maximizada
Volume dos Dados	Megabytes – Gigabytes	Gigabytes-Terabytes
Histórico dos Dados	Até a última atualização	5 a 10 anos

No contexto de SSD, os autores Brobst e Rarey [BR03], discutem em seu artigo os cinco estágios do processo de maturidade na utilização do *Data Warehouse* para suporte à decisão. A representação destes estágios pode ser visualizada na figura 2.1.

No primeiro estágio da evolução do *Data Warehouse* para suporte à decisão, normalmente se está voltado para a extração de relatórios a partir da execução de consultas pré-determinadas na base de dados da organização. Neste estágio, as consultas procuram responder à pergunta: “O que aconteceu?”. O grande desafio neste estágio é o processo de integração de dados, com a construção de um repositório consistente e confiável a partir das diversas fontes de dados referentes aos negócios da organização. Apesar do trabalho exaustivo envolvido nessa fase para a integração de dados, nela é constituída a base para os estágios subsequentes.

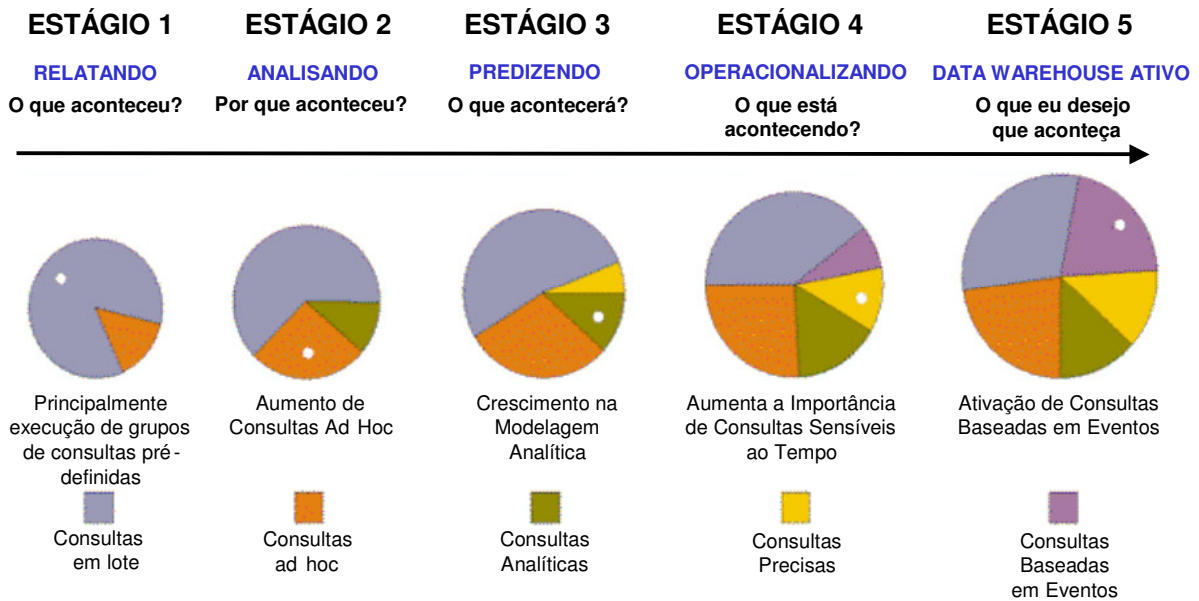


Figura 2.1: Os cinco estágios da utilização do *Data Warehouse* no suporte à decisão.

No estágio 2, os tomadores de decisão focam menos no "que" aconteceu e mais no "por-quê" determinado fato ocorreu. Para isso são utilizadas técnicas de processamento providas por ambientes OLAP. Nesta fase, para identificar o “porquê” dos fatos, ocorre o aumento de consultas *ad hoc* contra a base de dados da organização. Também ocorre a necessidade da execução concorrente de consultas e suporte a múltiplos usuários.

O próximo estágio da utilização do DW no suporte à decisão está em utilizar as informações disponíveis para tentar prever o que irá acontecer, baseando-se no comportamento que os dados vêm tendo. No estágio 3, normalmente, há um aumento da modelagem analítica e requer ferramentas de *Data Mining*, produzindo modelos que tentam prever determinados fatos com a utilização de dados históricos.

Enquanto os estágios 1, 2 e 3 estão direcionados ao suporte à decisão estratégico, o estágio 4 trata-se de um suporte à decisão tático. Entenda-se suporte à decisão estratégico como provimento de informações necessárias para tomada de decisões a longo prazo, incluindo segmentação de mercado, estratégias de gerenciamento de produtos, análise de rentabilidade, previsões e muitos outros. Por sua vez, suporte à decisão tático não está voltado ao desenvolvimento da corporação e sim para decisões imediatas, como por exemplo o gerenciamento de estoque para reposição na hora certa e o planejamento de rotas para entrega de encomendas.

O estágio mais avançado da evolução do DW é denominado de *Data Warehouse Ativo*. Neste quinto estágio, o foco está em automatizar o processo de tomada de decisões, de modo que o fator humano não adicione valor significativo. Um bom exemplo disso são as páginas *Web* de algumas organizações, as quais traçam um perfil do cliente que está visitando a página através de um rastreamento de suas ações enquanto este navega na página, para posterior oferta personalizada de produtos, preços e promoções. Outro exemplo é integração de um DW ativo com a tecnologia de prateleiras com legendas eletrônicas, onde os preços dos produtos são alterados automaticamente sem utilização de trabalho manual, de acordo com a ocorrência de determinado evento que indica a mudança nos preços.

A operacionalização e o suporte à decisão ativado por eventos, dos estágios 4 e 5, provêm a capacidade de execução das estratégias desenvolvidas a partir das análises tradicionais no sistema de DW, caracterizadas nos estágios 1, 2 e 3.

2.4 OLAP - *On-Line Analytical Processing*

Sistemas OLAP [CD96, Tho97] são uma categoria de software específica para realizar processamento analítico (consultas de suporte à decisão) nos dados de um DW, de forma que este processamento deve: (1) ocorrer com alto desempenho, consistência e interatividade e (2) auxiliar a tomada de decisão em uma organização através da interpretação desses dados em uma variedade de visões multidimensionais. Estas visões multidimensionais são suportadas graças à estrutura dimensional do DW e podem ser compreendidas como eixos e pontos de um espaço multidimensional, onde cada eixo pode ser encarado como uma dimensão ou perspectiva (e.g. tempo, área geográfica, sexo) e os pontos como um valor medido e correspondente à interseção desses eixos [Fid03].

Um conceito muito importante quando se refere ao paradigma multidimensional e ao processamento analítico on-line é o de cubo de dados, o qual possibilita a formação das visões multidimensionais. Os cubos são subconjuntos de dados do DW, organizados e sumarizados em estruturas multidimensionais. Essa forma de organização provê um mecanismo que permite um menor tempo de resposta a consultas complexas. Outras definições importantes incluem:

- As **DIMENSÕES** provêm uma descrição categórica pela qual as medidas são separadas para análise, ou seja, fornecem informações descritivas. Uma dimensão é formada por um conjunto de elementos de um mesmo tipo.
- Os **MEMBROS** são os elementos de uma dimensão. Por exemplo, em uma dimensão denominada *tempo*, os membros poderiam ser *Ano de 2000*, *Ano de 2001* e *Ano de 2002*.
- As **MEDIDAS** ou **FATOS** especificam os valores numéricos sumarizados para análise, tal como data, preço, quantidade vendida, ou seja, fornecem informações quantitativas.
- Uma **HIERARQUIA** pode ser definida como sendo o agrupamento de membros de uma dimensão. Uma hierarquia pode conter **NÍVEIS**, que são responsáveis por prover uma maior ou menor granularidade dos dados. Por exemplo, a hierarquia de uma dimensão tempo pode conter níveis para ano, semestre, mês e assim por diante.

Para representar graficamente alguns dos conceitos dados anteriormente, é apresentada na figura 2.2, uma metáfora visual para um cubo de dados denominado *Vendas*. Os elementos presentes nas dimensões desse cubo são chamados de membros e estes são organizados (*i.e.* agregados) em níveis de hierarquias [Tho97]. Por exemplo, a dimensão *Produto* organiza seus membros em níveis como *Categoria*, *Marca* e *Descrição*, os quais compõem uma hierarquia, de forma que a granularidade dos membros de um nível chamado “filho” é sempre menor do que a do seu nível “pai”. Por exemplo, o nível *Marca* tanto pode ser filho do nível *Categoria* quanto pode ser pai do nível *Descrição*. Ressalta-se que em uma dimensão do DW podem existir mais de uma hierarquia, as quais são mapeadas nas dimensões do cubo [Fid03].

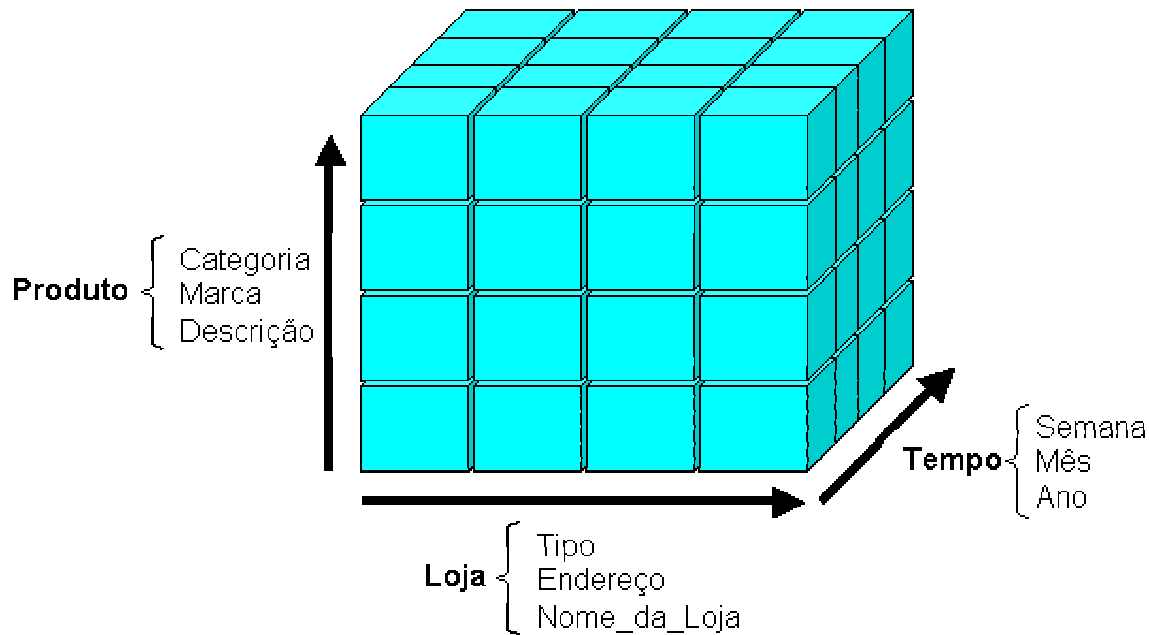


Figura 2.2. Metáfora Visual do cubo de dados *Vendas* (adaptado de [Fid03]).

2.4.1 Consultas Multidimensionais com Operadores OLAP

Para realizar consultas em bases de dados multidimensionais, pode ser utilizada a linguagem MDX (*Multidimensional Expressions*) [MDX03a, MDX03b]. Com a sintaxe definida pela MDX é possível realizar consultas em um cubo de dados multidimensionais, a fim de obter visões configuráveis dos dados em diferentes ângulos e níveis de agregação. Essas consultas podem ser realizadas através da aplicação de operadores OLAP [CD96, Fid03, FH01] como *SLICE/DICE*, *ROTATE* ou *PIVOT* e *DRILL-DOWN/DRILL-UP*.

Embora a sintaxe MDX seja, em muitas formas, semelhante à sintaxe da SQL (*Structured Query Language*) [Dat89, Mei93, SQL92], esta não é uma extensão da SQL. Algumas das funcionalidades suportadas pela MDX podem ser fornecidas pela SQL mas, muitas vezes, não de forma eficiente e intuitiva [CT97, GBL+96, MDX03b, Rao96].

Com as operações de *DRILL-DOWN/DRILL-UP* torna-se possível percorrer entre as hierarquias dimensionais, para oferecer visões mais refinadas ou mais agregadas dos dados. A operação *drill-down* de OLAP (ou *drill-up*) é uma técnica específica de análise segundo a qual o usuário navega ao longo dos vários níveis de agregação de uma informação, desde o seu nível

mais sumarizado (*drill-up*, também chamado de *Roll-Up*) até o mais detalhado (*drill-down*). Por exemplo, ao se ver dados de vendas do Brasil, uma operação de *drill-down* na dimensão Região poderia mostrar os estados do Centro-sul e os estados do Sul. Um *drill-down* posterior na região Sul poderia mostrar as cidades de Curitiba, Florianópolis e Porto Alegre. Em resumo, o *drill-down* fornece resultado mais detalhado e no *drill-up*, o resultado é mais sumarizado.

Com **ROTATE** ou **PIVOT** é possível modificar a orientação do cubo, trocando a posição das dimensões entre os eixos do cubo, gerando uma nova configuração de análise dimensional. Basicamente, se imaginarmos o resultado de uma operação OLAP dado em uma tabela, com a operação *pivot* as informações que estavam nas colunas passam a estar nas linhas e vice-versa.

Por sua vez, a utilização do operador OLAP **SLICE/DICE** faz com que o foco do usuário seja transferido para uma camada de dados particular ou subcubo de dados, respectivamente, pela secção do cubo em fatias ou pela extração de um subconjunto de dados agregados, fixando-se valores de dimensão.

A figura 2.3, extraída de [Ril03], demonstra graficamente o resultado da utilização de alguns destes operadores OLAP, em uma consulta a um cubo de dados multidimensionais.

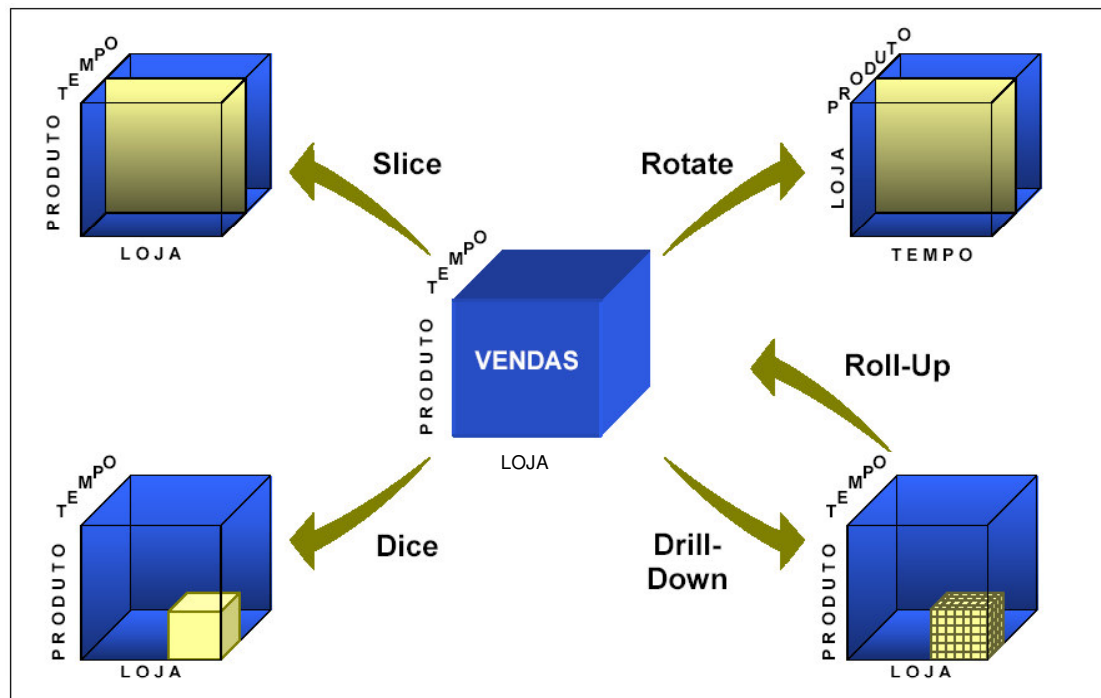


Figura 2.3: Representação dos operadores OLAP [Ril03].

Outra abordagem para consultas analíticas em bases de dados multidimensionais está em [CT97]. Neste artigo, os autores discorrem sobre uma linguagem de consulta a dados multidimensionais denominada *Multidimensional Calculus* (MD-CAL). A MD-CAL é baseada na realização de cálculos em uma tabela de medidas, e, segundo os autores, oferece um suporte de alto nível para a análise de dados multidimensionais. De acordo com a sintaxe da linguagem, funções escalares e agregadas podem ser embutidas nas expressões de cálculos de forma natural.

Outro estudo voltado para a análise de dados multidimensionais é apresentado em [GBL+96]. Neste, os autores introduzem o operador *Data Cube*, que possibilita a realização de agrupamentos, subtotais e cruzamentos para análise de dados. Com *Data Cube* também é possível a utilização de operadores analíticos como *drill-down* e *roll-up*.

2.5 SIG - Sistemas de Informações Geográficas

Os Sistemas de Informações Geográficas ou SIG [Chr96, Cla97, Dem97, MGR91, Mur95, RGH+91, RG93, Wor95] são também utilizados no suporte à decisão. Os SIG constituem-se de ferramentas para visualização, manutenção e recuperação de informações geo-referenciadas (*i.e.* dados que descrevem eventos/objetos localizados sobre a superfície terrestre).

Os SIG são uma ferramenta importante para áreas que necessitam de aplicações computacionais capazes de manipular e analisar informações baseadas em sua localização no espaço. No contexto de Suporte à Decisão, os SIG são aplicados em diversos ramos como planejamento urbano e rural, controle ambiental, gerenciamento de serviços de utilidade pública, e administração de recursos naturais [Ylu00].

A tecnologia de SIG [Chr96, MGR91, RGH+91, RG93, Wor95] tem sido definida como um sistema computacional (*i.e. hardware + software + peopleware*) que viabiliza: modelagem, captura, armazenamento, análise e apresentação - normalmente em mapas - de dados geo-referenciados. Estes dados são comumente representados a partir da união de três características básicas: 1) não-espacial - descreve o fenômeno geográfico; 2) espacial - informa sua localização e 3) temporal - retrata a data de sua coleta [CCH+96]. Para o *OGC (Open GIS Consortium)* [OGC99a, OGC99b, OGC99c, OGC99d, OGC01] estes tipos de dados são interpretados como Feições Geográficas, as quais representam uma abstração sobre um

fenômeno do mundo real que deve possuir um domínio temporal e uma localização no globo terrestre.

Para a realização de análises em um SIG, normalmente são disponibilizados aos usuários, mapas que demonstram visualmente o comportamento dos dados armazenados em um banco de dados geográfico. Em [FI96], os autores citam que um mapa é uma representação, em uma dada escala e sobre uma superfície plana, de uma seleção de características da superfície terrestre.

Em um SIG, a idéia de sobreposição de mapas é implementada empregando-se o conceito de sobreposição de temas ou camadas, onde, para um mesmo espaço geográfico podem ser criados diversos temas de dados geo-referenciados, um para cada acontecimento geográfico a ser representado. Esta abordagem facilita análises sobre os dados geo-referenciados, pois permite a combinação de dois ou mais temas para o processamento de uma determinada consulta [Fid03]. Na figura 2.4, pode ser visualizado um exemplo que representa a sobreposição de temas, conceito amplamente utilizado nas consultas sobre dados espaciais e comumente definido como visão geográfica.

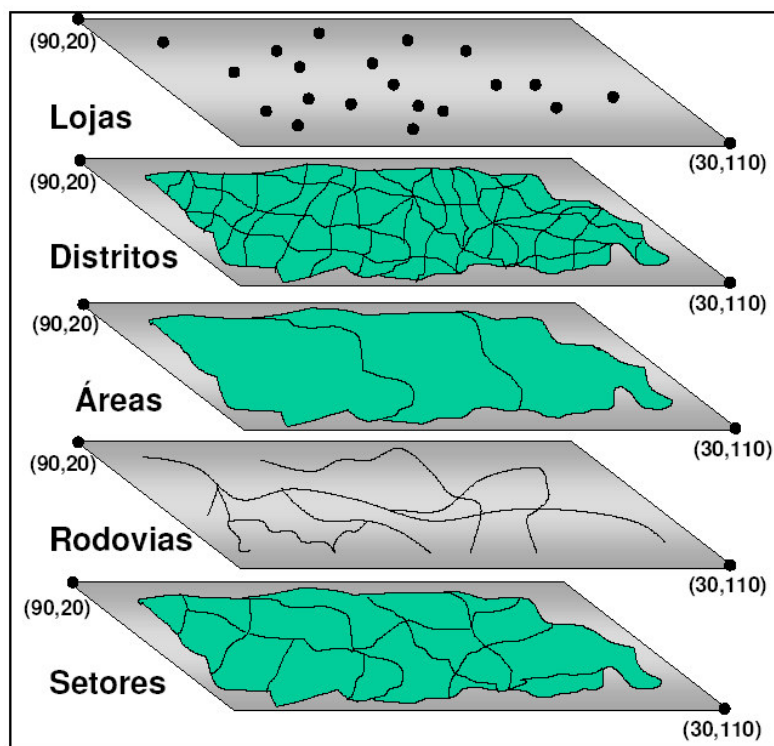


Figura 2.4: Exemplo de Sobreposição de Temas (Extraído de [Fid03]).

Em [CCH+96], o autor discorre sobre uma lista de componentes básicos de um Sistema de Informações Geográficas. A estrutura apresentada na figura 2.5 representa o relacionamento existente entre estes componentes. Em um nível mais externo se encontra a interface com o usuário, a qual disponibiliza a forma como o sistema é operado. No nível intermediário, se encontram os módulos responsáveis pelo processamento dos dados espaciais, incluindo entrada, edição, análise e visualização. Em um nível mais interno, se encontra o sistema de gerenciamento de banco de dados geográficos, o qual deve suportar o armazenamento e recuperação dos dados espaciais. A seguir, cada um desses componentes é descrito brevemente.

- **Interface com Usuário:** Em substituição às antigas interfaces em modo texto, as quais eram baseadas em comandos, pouco interativas e complexas para o usuário, cada vez mais são disponibilizadas interfaces gráficas para manipulação das funcionalidades oferecidas pelos SIG, as quais são constituídas de menus, caixas de diálogo, e representação gráfica de operadores;
- **Entrada e Integração de Dados:** Isto compreende o conjunto de operações a serem aplicadas antes da utilização efetiva dos dados. Normalmente este componente abrange funcionalidades que permitem a entrada de dados através de sua digitalização, leitura dos dados de forma digital, e procedimentos para validação dos dados e tratamento de erros oriundos da captura dos dados georeferenciados;
- **Consulta e Análise Espacial:** Este é um dos mais importantes componentes de um SIG. Dentre as funcionalidades oferecidas por este componente, estão a seleção e pesquisa sobre informações geográficas, transformações de escala ou projeção, sobreposição de camadas de dados e execução de operações espaciais;
- **Visualização e Plotagem:** É responsável pela apresentação gráfica dos resultados de consultas e análises espaciais de maneira que o usuário possa interpretar facilmente estes dados;

- **Armazenamento e Recuperação de Dados:** A este módulo estão delegadas responsabilidades intrínsecas a um sistema de gerenciamento de banco de dados, como por exemplo manter os dados armazenados de forma consistente, controlar o acesso concorrente, garantir a recuperação total ou parcial dos dados em caso de falhas e garantir a segurança das informações. Este componente realiza o gerenciamento de dados geográficos, incluindo a provisão de tipos de dados espaciais, métodos de acesso espacial eficientes e linguagens de consulta espacial.

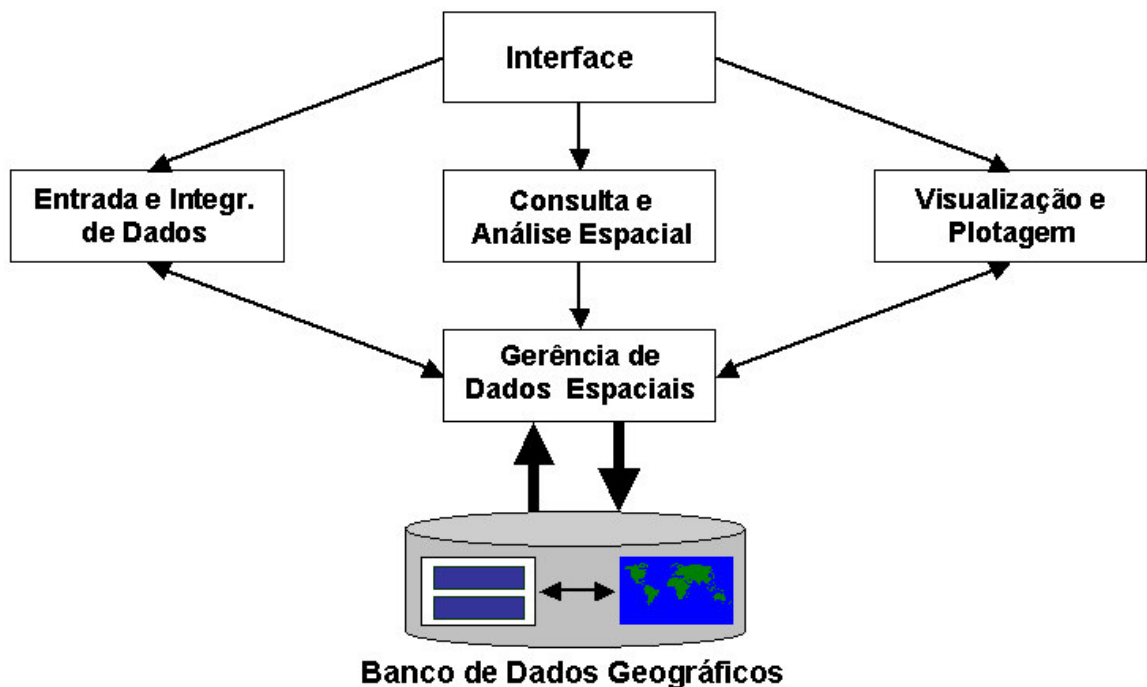


Figura 2.5: Componentes de um Sistema de Informações Geográficas (Adaptado de [CCH+96])

2.6 DWG - Data Warehouse Geográfico

Como mencionado anteriormente, os SIG constituem um instrumento útil e muito particular para suporte à decisão. O primeiro enfoque mais abrangente da utilização dos recursos de SIG no apoio à decisão foi a definição do conceito de Sistemas Espaciais de Suporte à Decisão (SESD) [SM03], que diferem dos SIG, embora repousem sobre a mesma tecnologia. Um Sistema de Suporte à Decisão tradicional é definido como um sistema computacional de auxílio à decisão que contém bases de dados, um conjunto de modelos otimizados para apuração dos dados visando apoio à decisão e uma interface amigável e flexível que permita ao tomador de decisões,

consultar e manipular as bases de dados e modelos, seguindo critérios próprios e em tempo real [Spr80]. Um SIG pode incluir todos estes atributos tornando-se um SESD, utilizando intensivamente seus recursos de análise espacial, ora na visualização de "camadas" de informação comuns a uma dada região, ora na construção de mapas temáticos ou na análise estatística e de prognósticos.

Para obtenção de bons resultados, os SESD tradicionalmente devem ser apoiados por um *Data Warehouse* Geográfico (DWG) [Fid03, Boh04, BSG00, ZFG01] que por sua vez se encarregam de armazenar e gerenciar as informações geográficas ou espaciais sobre o negócio da organização. Estas informações vão desde o endereço dos fornecedores até a distribuição espacial das filiais da empresa.

Um DWG é uma extensão da abordagem tradicional de *Data Warehouse* com um componente espacial, o qual pode ser especificado por três enfoques [SC98]:

- O primeiro referindo-se à necessidade de uma ferramenta que execute as agregações espaciais, conversões geográficas e georeferenciamento;
- O segundo referindo-se à modelagem do banco de dados de apoio ao DWG que deve ser feita seguindo uma combinação entre as características de um banco de dados espacial e de um banco de dados voltado para DW (incluindo suporte à multidimensionalidade e escalabilidade);
- Finalmente, o enfoque sobre a análise estatístico-espacial dos dados e a apresentação alternativa dos resultados em mapas ou tabelas.

O objetivo mais importante desses enfoques é sempre permitir que sejam feitas: (1) consultas ao DWG a partir dos atributos geográficos, (2) análise espacial destes resultados, (3) refinamentos sucessivos e (4) agregações dos resultados que podem ser visualizadas instantaneamente por áreas geográficas.

Como pode-se notar, algumas dessas possibilidades são viabilizadas somente se existir um suporte de SGBD adequado e uma ferramenta OLAP de apuração de dados contemplada com muitas facilidades de SIG. Por outro lado, se considerarmos um SIG que possa acessar uma base de dados modelada como DWG, vamos sentir a falta da flexibilidade que uma ferramenta OLAP possui para tratar dados analíticos. Assim, as duas ferramentas, SIG e OLAP,

se ressentem da falta de integração, seja no ambiente de *Data Warehouse* ou para uso geral no suporte à decisão [SC98].

2.7 Considerações Finais

Este capítulo apresentou conceitos relacionados a SIG, *Data Warehouse*, OLAP e *Data Warehouse* Geográfico. Estas tecnologias são utilizadas para auxiliar no processo de tomada de decisões. Com as facilidades disponibilizadas por ferramentas dessa natureza, os dados sobre o negócio de uma organização podem ser analisados com mais facilidade, podendo abranger o aspecto analítico e geográfico.

Com o que foi apresentado, podemos supor que um ambiente para suporte à decisões estratégicas, seja altamente favorecido se englobar técnicas e procedimentos com capacidades de processar, de forma simultânea, dados analíticos e espaciais. O que possibilitaria, o tratamento de aspectos relacionados à natureza dos dois tipos de dados.

A grande motivação para a construção de um sistema de suporte à decisão, é que os “tomadores de decisões” das organizações tenham a capacidade, tanto de realizar simples consultas que demonstrem o comportamento atual do negócio, quanto de analisar a razão deste comportamento. Além disso, e o que talvez seja considerado o aspecto mais importante no contexto da tomada de decisões, é a capacidade para realizar previsões a respeito do comportamento futuro do negócio, incluindo também funcionalidades para redirecionar este comportamento a fim de que ele satisfaça as metas da organização. No que se refere aos sistemas espaciais de suporte à decisão, um requisito importante é a manipulação e representação gráfica de dados com características espaciais, possibilitando por exemplo, a visualização em mapas, da distribuição das lojas em uma determinada região, bem como da distribuição de seus clientes. Outro fato é a possibilidade de aplicar operações espaciais sobre os dados geográficos.

No próximo capítulo, são apresentadas algumas pesquisas encontradas na literatura que tratam da integração de conceitos e funcionalidades oferecidas pelas tecnologias descritas neste capítulo. Ou seja, estas pesquisas têm seus esforços voltados para unir o poder de processamento, e facilidades oferecidas pelos SIG, DW e ferramenta OLAP, para o auxílio à tomada de decisões.

Capítulo 3

Trabalhos Relacionados e Contextualização

O objetivo deste capítulo é apresentar algumas pesquisas que se relacionam com o trabalho descrito nessa dissertação. Neste sentido, serão analisadas algumas propostas disponíveis na literatura que são voltadas para a integração dos processamentos analítico e geográfico. Como contextualização da pesquisa apresentada neste documento, será apresentada a arquitetura GOLAPA e a descrição de seus componentes. Todos estes trabalhos, são considerados de grande importância no contexto da integração de serviços analíticos e geográficos na *Web*.

3.1 Introdução

Este capítulo tem como objetivo apresentar algumas das pesquisas encontradas na literatura, as quais possuem ligação com o tema discutido nesta dissertação. Neste sentido, são descritas brevemente as abordagens utilizadas pelos projetos *MapCube* [SLT+00a, SLT+00b], *GeoMiner* [HKS97], *GOAL* [Goa03] e *SIGOLAP* [FCT01, Fer02]. Todas estas propostas estão voltadas para o fornecimento de um ambiente que possibilite a realização de processamento geográfico-multidimensional. Sendo assim, todas são de grande importância para o presente trabalho, visto que ele trata da integração de serviços analíticos e geográficos para suporte à decisão na *Web*.

Entretanto, as propostas citadas acima possuem algumas deficiências, pois algumas delas não abrangem de forma completa todos os aspectos pertinentes aos processamentos analítico e geográfico. Outro fato, é que estas propostas não estão baseadas em padrões abertos, e muitas vezes, estão fortemente atreladas a tecnologias proprietárias. Estes fatores motivaram a proposição da arquitetura GOLAPA [FTS01, Fid03], que será outro trabalho analisado neste capítulo. Este, terá uma atenção especial, devido a esta dissertação estar inserida no contexto desta arquitetura. Ao final deste capítulo, será apresentado um estudo comparativo, abrangendo todas as abordagens mencionadas.

As demais seções deste capítulo estão organizadas da seguinte forma: A seção 3.2 apresenta uma breve descrição de alguns trabalhos voltados para a integração de processamento analítico e geográfico. A seção 3.3 contém a contextualização da pesquisa descrita nesta dissertação. Na seção 3.3.1, são listados alguns conceitos relativos à arquitetura GOLAPA. Na seção 3.3.1.1 são apresentadas as camadas e componentes de GOLAPA. A seção 3.3.1.2 propõe o *GML For Analysis Schema* (GMLA). A seção 3.3.1.3 apresenta o metamodelo GAM e o esquema GeoMD. Na seção 3.3.1.4, é descrito brevemente o arcabouço GeoDWFrame. Finalmente, na seção 3.4 são tecidas algumas considerações finais sobre os tópicos abordados no capítulo.

3.2 Propostas para Processamento Geográfico-Multidimensional

Assim como a pesquisa descrita neste documento, outros trabalhos como o *MapCube* [SLT+00a], *GeoMiner* [HKS97], *GOAL* [Goa03] e *SIGOLAP* [FCT01, Fer02], estão voltados para o fornecimento de um ambiente para processamento geográfico-multidimensional.

Map Cube é um operador para realizar agregações e visualizações de dados espaciais em um álbum de mapas. *Map Cube* baseia-se no operador *Cube* [GBL+96] e assim como o próprio, efetua todas as possíveis combinações de agregações por cada dado de cada dimensão do DW. A principal diferença entre os operadores *Cube* e *Map Cube* é a saída de seus dados, pois o *Cube* apenas os representa em tabelas, enquanto que o *Map Cube*, em tabelas e mapas. As vantagens da proposta *Map Cube* é que ele une a representação tabular e por mapas em uma única visão dos dados. Um fator que facilita a análise dos dados é o conceito de álbum de mapas. A sintaxe do operador é relativamente simples e inspirada em SQL [Dat89, Mel93, SQL92]. Entretanto, não contempla operações espaciais sobre os dados, não podendo, dessa forma, ser considerada uma proposta de integração entre processamento analítico e geográfico. São mencionados conceitos relativos a *Data Warehouse* Geográfico, entretanto não prevê a utilização de metadados como meio de identificar relacionamentos entre os dados analíticos e geográficos.

GeoMiner é um projeto para mineração de dados geográficos que consiste em três módulos básicos: (1) módulo construtor de cubos de dados geográficos, (2) módulo de OLAP geográfico e (3) módulo de mineração de dados geográficos. Destes, apenas os módulos (1) e (2) caracterizam-se como pesquisas relacionadas a este trabalho. Em [HSK98, Ste97], Han *et al.* propuseram dois algoritmos para construção eficiente de cubos de dados geográficos: interseção de ponteiro e conexão de objeto. Estes algoritmos geram cubos analíticos e espaciais baseando-se em critérios que definem uma materialização seletiva de um conjunto de objetos geográficos. Os critérios definidos são: (1) a frequência de acesso desse conjunto, (2) seu custo de armazenamento e (3) o benefício que o mesmo trará para construção de outros cubos geográficos derivados desse conjunto. A partir destes cubos realiza-se as operações OLAP tradicionais. A proposta emprega conceitos de *Data Warehouse* Geográfico, entretanto menciona a utilização de soluções proprietárias como o SIG *MapInfo* [MI03].

GOAL (*Geographical Information On-Line Analysis*) é um projeto que objetiva integrar DW/OLAP com SIG. Para alcançar este fim, Kouba *et al.* [KMM00] implementaram

uma arquitetura cujo componente responsável por coordenar esta tarefa chama-se Módulo de Integração (MI). A proposta GOAL sugere a integração de SIG com DW/OLAP via uma taxonomia entre classes/objetos de SIG e uma dimensão geográfica do DW/OLAP. Para isso, três tipos de correspondências dinâmicas (classe, instância e ação) são propostos. Além disso, faz uso de metadados para suportar todo o processo das correspondências dinâmicas. Como estudo de caso, os autores utilizaram os programas *GT Media98* e *ArcView* [ESRI03] como componentes SIG e o servidor MSOLAP³ [MSQL03] como componente OLAP. Dado que estes programas possuem interfaces de comunicação diferentes, o Módulo de Integração tem que ser ajustado para oferecer o devido suporte a cada um deles. A comunicação com o *ArcView* foi implementada através de *scripts Avenue* (i.e. a linguagem nativa do *ArcView*) e bibliotecas em *Delphi* [DEL03]. A comunicação com *GT Media98* não foi citada. Já a comunicação com o MSOLAP foi realizada a partir da implementação de um provedor OLAP para OLE DB [OLE03] utilizando *Visual C++* [VC03]. Esta proposta está fortemente atrelada à plataforma Windows. Isto ocorre por não se basear em padrões abertos para possibilitar a integração.

SIGOLAP [FCT01, Fer02] objetiva integrar SIG e OLAP via uma arquitetura em três camadas, a qual é baseada em mediadores e faz uso de um metamodelo de integração, utilizando metadados para auxiliar no processo de integração. Para validação desta arquitetura, foi implementada uma aplicação para o contexto de uma seguradora atuando no ramo de saúde [Fer02]. As tecnologias usadas na implementação da solução *SIGOLAP* foram: para OLAP - *SQL Server* com *MSOLAP*; para SIG - *MSAccess* [MA03] e *AutoDesk MapguideServer* [AMG03]; para o modelo de integração e mediação - *SQL Server* [MSQL03] e *Visual Basic* [VB03] e para a interface com o usuário - *Visual Basic Script*. Como pode-se notar, a proposta é fortemente atrelada a plataforma *Windows*, e não baseia-se em padrões abertos nem na utilização de um *Data Warehouse* Geográfico.

Da avaliação do estado da arte das propostas para processamento geográfico-multidimensional, conclui-se que todas possuem alguma limitação, por não abranger de forma satisfatória ambos os processamentos (i.e. analítico e espacial), ou pelo fato de estarem voltadas para a utilização de soluções proprietárias ou dependentes de plataforma. Exemplo disso são os projetos *GOAL* e *SIGOLAP*, que de fato propõem uma abordagem para integração entre OLAP e SIG, entretanto a utilização de tecnologias proprietárias faz com que estas não sejam independentes de plataforma e sobretudo dificulta a extensibilidade e o intercâmbio de seus

³ Microsoft OLE DB Provider for OLAP Services – Disponível no CD de Instalação do MS SQL Server

dados. De forma resumida, pode-se dizer que a maioria das propostas aborda o problema de forma unilateral. Isto porque, de forma não integrada, um *DW/OLAP* é usado para melhorar as consultas do *SIG* ou um *SIG* é usado para oferecer suporte de operações geográficas a um *DW/OLAP*. Assim, conclui-se que as soluções propostas não abordam a questão de integração entre processamento analítico e geográfico de forma: (1) aberta - que não seja dependente de plataforma e (2) única - que integre, em uma única arquitetura, funcionalidades de *DWG*, *OLAP* e *SIG*. Estes fatores foram os principais motivadores para a proposição de GOLAPA, apresentada na seção 3.3.1, a qual visa superar as limitações apresentadas por estas propostas.

É importante salientar que além dos trabalhos citados anteriormente, existem na literatura outras pesquisas focadas em aspectos pertinentes à integração de ambientes para processamento analítico e geográfico, estando voltadas mais especificamente para pontos como linguagens de consulta analítica-geográfica, indexação e agregação espacial. Apesar deles não constituírem o foco principal desta dissertação, alguns destes trabalhos serão brevemente descritos nesta seção.

Em [FPR+00], uma linguagem de consulta para suportar a integração de operadores multidimensionais e espaciais é proposta. Neste trabalho, os autores apresentam uma abordagem conceitual, a qual permite que um usuário final, trabalhando em um ambiente de banco de dados geográficos possa utilizar dados estatísticos armazenados em banco de dados estatístico.

Outra pesquisa, voltada para a definição de uma linguagem de consulta é apresentada em [PR02]. Neste trabalho, os autores apresentam uma linguagem para expressar consultas em um banco de dados geográfico orientado a objetos, utilizando operadores posicionais e analíticos. Ainda no que se refere à linguagens de consultas, pode ser citado o trabalho de Kamp [Kam98], o qual apresenta uma linguagem de consulta orientada a objetos, que permite formular consultas envolvendo aspectos espaciais e temporais.

Em [PKZ+01, PTK+02], os autores apresentam uma abordagem para prover operações do tipo OLAP em sistemas de *data warehouses* espaciais. No artigo apresentado em [PKZ+01], é proposta uma estrutura de dados chamada aR-Tree, a qual combina técnicas de indexação espacial e materialização para promover um suporte para consultas de agregação espacial.

Nesta mesma linha de pesquisa, em [ZLR+03a], os autores apresentam uma abordagem que tem como intuito possibilitar a manipulação de dados espaciais em sistemas OLAP, pela agregação de uma hierarquia espacial. Neste trabalho, os autores estendem a consulta ao cubo de dados introduzindo predicados espaciais e funções que expressam explicitamente os relacionamentos espaciais entre dados e uma tabela de fatos e de dimensões. Essa abordagem utiliza um mecanismo de índice espacial para derivar a hierarquia espacial por pré-agregação e materialização. Mais informações podem ser encontradas em [ZLR+03b].

Em [PT01], os autores investigam a utilização de pré-agregação em *data warehouses* espaciais, com intuito de permitir que consultas de agregação espacial sejam realizadas com um menor tempo de resposta. Ainda no contexto de agregação espacial, em [PZ04], os autores relatam que a agregação espacial é um importante passo na geração de cubos de dados espaciais para suportar operações características de OLAP, como por exemplo drill-down/roll-up. Os autores então apresentam uma nova metodologia, a qual visa melhorar o desempenho em atividades de agregação espacial.

Na pesquisa apresentada em [HKS00], os autores mostram um estudo sobre métodos para OLAP espacial, propondo a integração de métodos pertinentes às aplicações OLAP tradicionais com técnicas de implementação de bases de dados espaciais. Neste trabalho, é proposto um modelo de *data warehouse* geográfico que consiste de medidas e dimensões espaciais e não espaciais. Também é apresentado pelos autores um método para construção de cubos de dados espaciais, o qual é chamado de materialização seletiva baseada em objetos (*object-based selective materialization*).

Finalmente, em [BRM01], os autores dão a definição de ferramentas para OLAP espacial, bem como apresentam conceitos relacionados a elas. Neste trabalho são descritas características essenciais e desejáveis dessa nova categoria de ferramentas para processamento analítico espacial.

3.3 Contextualização da Dissertação

O trabalho apresentado nesta dissertação faz parte do Projeto GOLAPA [GHP03]. Este projeto direciona seus esforços para tentar solucionar o problema da integração entre processamento

analítico e geográfico. Para isso, foi proposta a arquitetura GOLAPA (*Geographical On-Line Analytical Processing Architecture*) [FTS01, Fid03], a qual será apresentada na próxima seção.

3.3.1 Arquitetura GOLAPA

Como descrito em [Fid03], para lidar com sistemas grandes e complexos deve ser enfatizado o uso de uma disciplina/sistemática que permita obter resultados a baixo custo e com qualidade. Dentro deste contexto, a arquitetura de *software* desempenha um importante papel. Esta provê uma visão, com um elevado nível de abstração, da organização global do sistema de *software* bem como dos relacionamentos/colaborações existentes entre seus subsistemas e/ou componentes [Men02].

GOLAPA é uma arquitetura de *software* proposta pelo Projeto GOLAPA, do Centro de Informática da Universidade Federal de Pernambuco⁴ e, visa ser: (1) baseada em tecnologias abertas e extensíveis e (2) otimizada para suporte à decisão em um contexto multidimensional-geográfico. O primeiro objetivo trata da integração sobre o aspecto tecnológico, isto é, que sua implementação seja baseada em tecnologias como *CORBA*, *Java* e/ou *XML*. Já o segundo discorre sobre a estratégia que é, potencialmente, a melhor solução para a integração dos processamentos analítico e geográfico.

Para alcançar o primeiro objetivo, optou-se por *Java* e *XML* como linguagens francas para implementação de *GOLAPA*. Esta escolha deve-se ao fato destas linguagens serem amplamente difundidas e permitirem seus usos com outras tecnologias (e.g. *CORBA* e *Web Services*). Quanto ao segundo objetivo, optou-se por aplicar uma arquitetura baseada na utilização de um *Data Warehouse* Geográfico, devido a esta abordagem ser a mais apropriada para aplicações que requerem alto desempenho de consulta e que não exigem que seus dados estejam permanentemente sendo atualizados - características desejáveis para uma arquitetura de suporte à decisão.

⁴ Centro de Informática da Universidade Federal de Pernambuco, <http://www.cin.ufpe.br/>

3.3.1.1 Camadas e Componentes de GOLAPA

As cinco camadas da arquitetura GOLAPA foram distribuídas conforme apresentado na figura 3.1 [Fid03]. GOLAPA dispõe de três camadas essenciais para o suporte multidimensional-geográfico: Camadas (I), (II) e (III), que respectivamente tratam dos dados, serviços e interface gráfica de um sistema com capacidades de processamento tanto analítico quanto geográfico. Além disso, duas camadas de suporte: Camadas (A) e (B) que respectivamente tratam dos dados operativos e da construção/manutenção do *DWG*. Por GOLAPA ser uma arquitetura em camadas, esta segue a especificação que uma camada qualquer (*e.g.* II) provê serviço para a camada imediatamente superior (*i.e.* III), bem como faz uso dos serviços da camada imediatamente inferior (*i.e.* I). Ressalta-se que as camadas de *GOLAPA* agrupam seus subsistemas/componentes de forma a prover o máximo de coesão e o mínimo de acoplamento - fatores essenciais para uma boa manutenção e reuso de software [Men02].

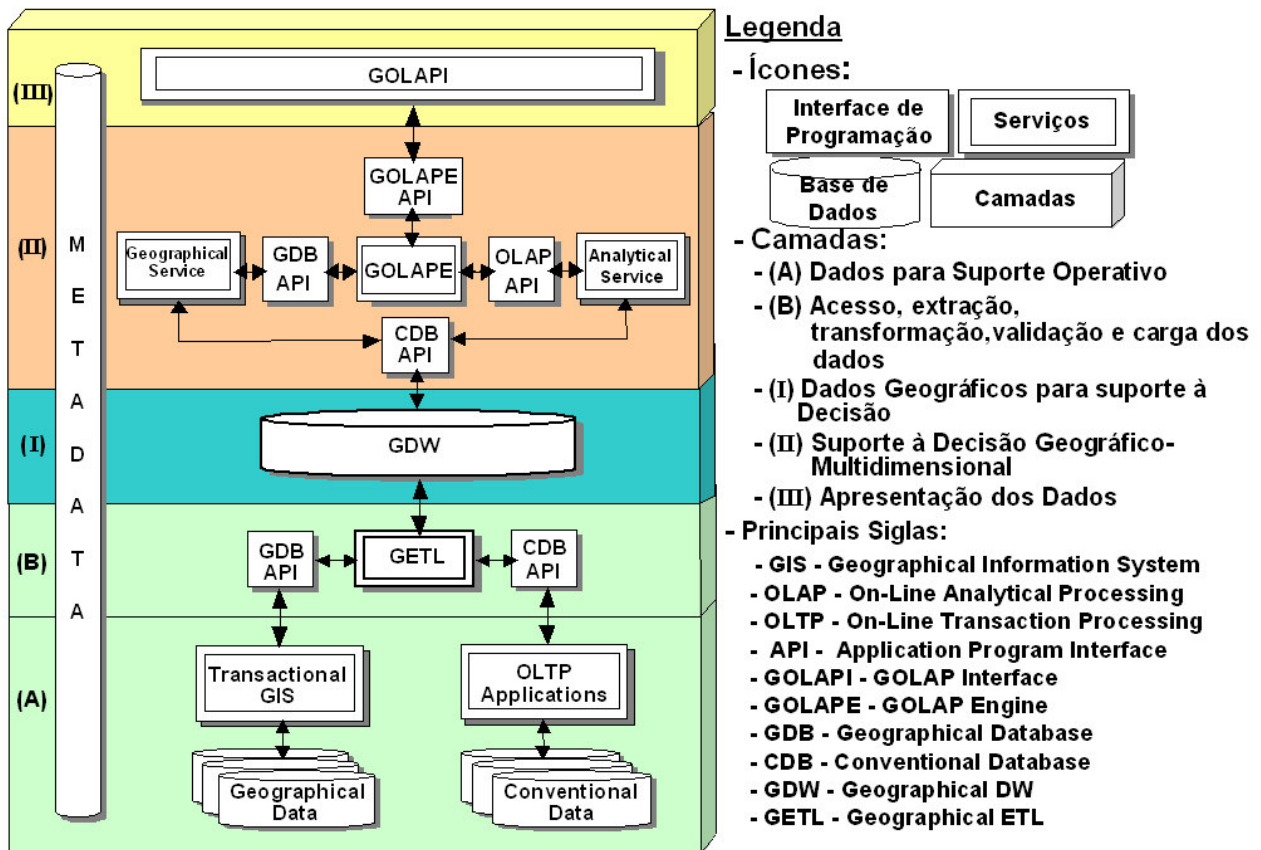


Figura 3.1: Arquitetura GOLAPA [Fid03].

As camadas presentes na arquitetura GOLAPA podem ser descritas da seguinte maneira:

Camada (A) - Dados para suporte operativo: Correspondente às fontes de dados transacionais. Ressalta-se que tanto o *SIG* como outras ferramentas desta camada são inerentemente para processamento *OLTP*.

Camada (B) - Acesso, extração, transformação, validação e carga de dados: Responsável pela conversão dos dados do ambiente operativo para o de decisão.

Camada (I) - Dados para suporte à decisão: Ela explicitamente divide o ambiente transacional do ambiente de suporte à decisão. É garantido que nenhuma operação ocorrida no ambiente transacional seja refletida no ambiente de suporte à decisão, a não ser pela ocorrência de uma recarga no *Data Warehouse Geográfico*. O *DWG* presente nesta camada tem as propriedades geográficas definidas em GML [GML03a].

Camada (II) - Suporte à decisão geográfico-multidimensional: Responsável pela manipulação das funcionalidades referentes ao processamento geográfico, analítico, e analítico-geográfico. O componente que irá executar esse processamento é o componente *GOLAPE*, o qual tem habilidades de processamento tanto de dados analíticos como de dados geográficos. É nesta camada que está inserido o trabalho desenvolvido nesta dissertação, a qual propõe um modelo de integração de serviços analíticos e geográficos e implementa o mesmo baseando-se na tecnologia dos *Web Services*, Java e XML.

Camada (III) - Apresentação dos dados: Responsável pela visualização dos dados. Esta pode ser tanto uma interface gráfica local, quanto um cliente remoto, ou um navegador para a Internet. Apesar desta dissertação estar voltada para a camada (II), foi desenvolvido um protótipo para apresentação gráfica dos dados resultantes do processamento geográfico-multidimensional.

Os principais componentes da arquitetura GOLAPA podem ser descritos da seguinte maneira:

GELT (*Geographical Extraction, Load and Transformation*) - Este componente representa uma ferramenta ETL [Kim96] (Extração, Transformação e Leitura) tradicional com suporte a dados geográficos. Em caso de não se dispor de uma ferramenta de ETL geográfico para DWG, um serviço similar pode ser provido, fazendo uso de uma ferramenta exclusiva para ETL Geográfico (*i.e.* FME [FME03]) com uma ETL tradicional (*i.e.* DTS [MSQL03]) [Fid03].

GDW (*Geographical Data Warehouse*) - Representa o *Data Warehouse* Geográfico, o qual possui grandes semelhanças com as abordagens tradicionais de esquemas de *Data Warehouse* (*i.e.* Estrela e Blocos de Neve) [Inm97, Kim96]. Entretanto, sua estrutura suporta o armazenamento tanto de dados analíticos quanto de dados geográficos. Um modelo conceitual de um *Data Warehouse* Geográfico é definido em [Fid03].

Metadata - Este componente é de grande importância para a arquitetura GOLAPA. Ele representa a base de metadados, a qual irá conter as informações necessárias para a integração entre os processamentos analítico e geográfico. Esta base armazena as informações referentes às correspondências existentes entre dados analíticos e geográficos. Isso se torna possível com a utilização dos conceitos pertinentes ao metamodelo GAM e o esquema GeoMD [Fid03].

Geographical Service - Representa um serviço voltado para processamento de dados geográficos, o qual executa somente operações de leitura sobre os referidos dados.

GDB/CDB API - Estes correspondem às interfaces de programação utilizadas para acessar serviços geográficos e convencionais (*i.e.* FME [FME03], JDBC [JDBC03]), respectivamente, os quais são responsáveis por extrair os dados que serão carregados no *Data Warehouse* Geográfico.

Analytical Service - Representa o serviço responsável pelo processamento dos dados analíticos. É pela integração deste com o componente *Geographical Service*, que o componente GOLAPE provê o processamento analítico-geográfico.

GOLAPE (*Geographical On-Line Analytical Processing Engine*) - Este pode ser considerado o principal componente utilizado para prover a integração entre o processamento analítico e geográfico. Resumidamente, ele é responsável por receber requisições, enviadas por uma aplicação cliente, e classificá-las em: (1) geográfica, (2) multidimensional ou (3) geográfica-multidimensional. Em seguida, ele processa estas requisições e envia o resultado ao cliente.

GOLAPE API - Corresponde à interface de programação GOLAPE, e também está inserido no contexto desta dissertação. Esta interface possibilita a comunicação entre os componentes GOLAPE e GOLAPI.

GOLAPI (*Geographical On-Line Analytical Processing Interface*) - Representa a interface gráfica utilizada para consulta e visualização de consultas geográficas, multidimensionais e geográficas-multidimensionais. O foco deste componente é prover uma metáfora visual para consultar e representar informações analíticas e/ou geográficas.

3.3.1.2 GMLA - *GML For Analysis*

Outro trabalho inserido no contexto da arquitetura GOLAPA e da presente dissertação é o *GML For Analysis Schema* (GMLA) [FST+03a]. GMLA é um *XML Schema* [SCHE03] (seção 4.2.2) definido a partir de GML [GML03a] (seção 4.2.4) e XMLA [XMLA02, XMLA03] (seção 4.4) e voltado para a descrição e validação de dados geográficos-multidimensionais. Da maneira como foi definido, GMLA importa, estende e integra os dois esquemas anteriores (*i.e.* GML e XMLA). Dessa forma, dados que eram descritos por dois esquemas distintos passam a ser descritos e integrados semanticamente por um único esquema. Este esquema resultante é utilizado para definir a estrutura dos documentos contendo dados multidimensionais-geográficos, integrados pelo componente GOLAPE.

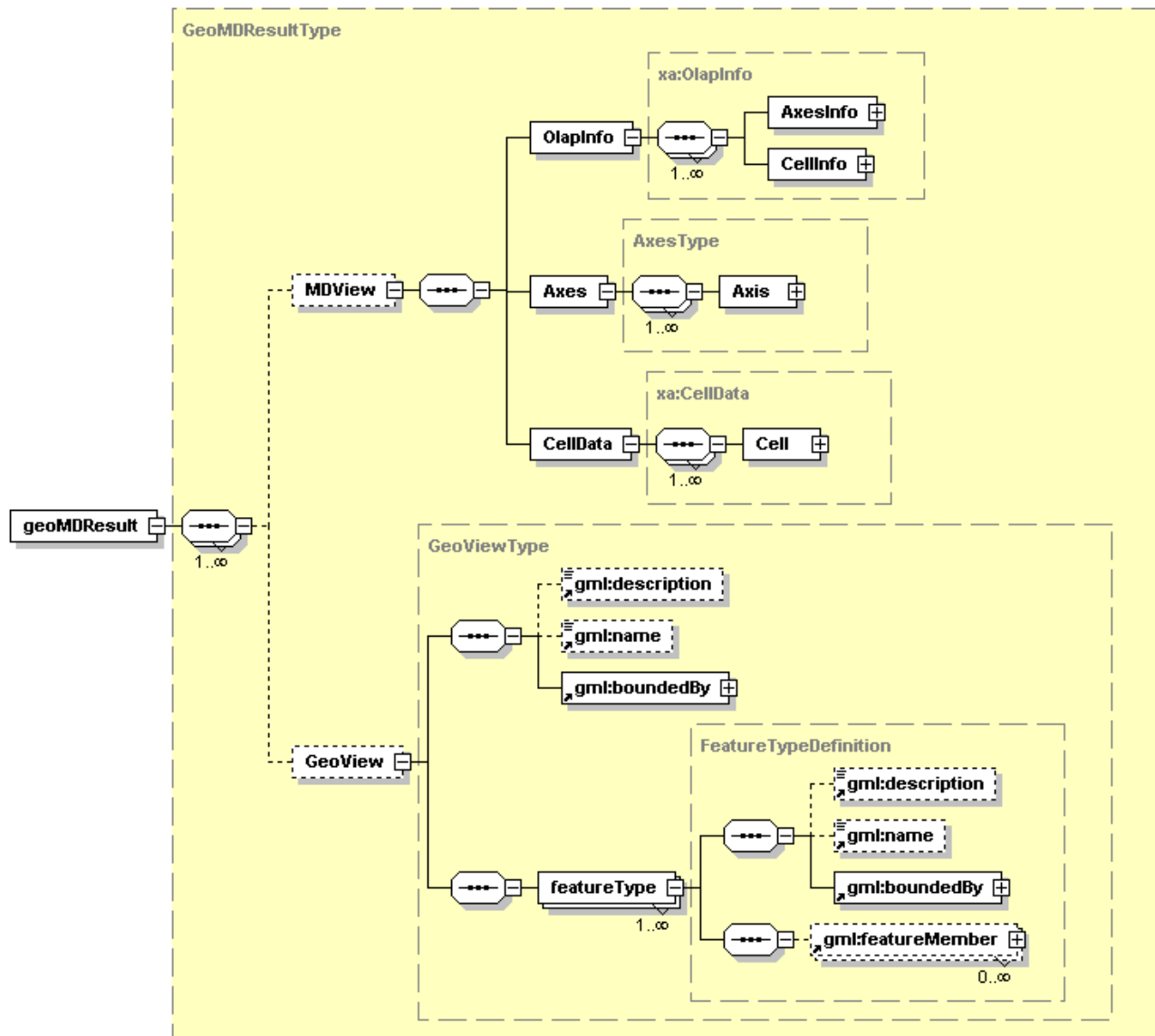


Figura 3.2: Diagrama do *GMLA Schema*.

Como pode ser visualizado no diagrama da figura 3.2, gerado a partir da ferramenta *XML Spy* [XSPY03], o elemento raiz do GMLA é o *geoMDResult*. O esquema GMLA pode descrever dados somente geográficos, somente multidimensionais, ou ainda geográficos-multidimensionais. No caso de dados com natureza somente geográfica, estes serão descritos utilizando o elemento *GeoView*, o qual possui propriedades oriundas de GML e organiza as feições geográficas, juntamente com suas geometrias, em camadas/temas por meio do elemento *featureType*. No caso da necessidade de descrever apenas dados multidimensionais, somente é utilizado o elemento *MDView*, o qual herda algumas definições de XMLA e utiliza os sub-

elementos *OlapInfo*, *Axes* e *CellData* para apresentar as propriedades pertinentes aos dados multidimensionais. Estes subelementos descrevem metadados de um subcubo multidimensional (*i.e.* *OlapInfo*), informações pertinentes aos eixos que compõem o subcubo (*i.e.* *Axes*), bem como os dados em si (*i.e.* *CellData*).

Vale ressaltar que, por ser definido através de um padrão aberto e extensível, o esquema GMLA não está voltado somente para o contexto da arquitetura GOLAPA, podendo ser utilizado por outras aplicações para descrição e intercâmbio de dados geográficos-multidimensionais. Por basear-se em XML, o esquema GMLA pode, se necessário, ser facilmente estendido e adaptado de acordo com o contexto de uma determinada aplicação que necessite descrever ou transportar dados com características de natureza multidimensional e/ou geográfica.

3.3.1.3 O Metamodelo GAM e o Esquema GeoMD

Ainda no contexto da arquitetura GOLAPA, e da integração entre tecnologias de processamento analítico e processamento geográfico, pode-se listar dois outros trabalhos relacionados, o metamodelo GAM (*Geographical Analytical Metamodel*) e o Esquema GeoMD (*Geographical Multidimensional*) [Fid03].

GAM é utilizado pelo componente *Metadata* de GOLAPA, o qual dispõe de metadados que permitem a integração entre o processamento analítico e geográfico. Isso ocorre devido a esta fonte de metadados fornecer informações que identificam as correspondências existentes entre os dados analíticos e geográficos, armazenados no *Data Warehouse* Geográfico.

Por sua vez, o esquema abstrato GeoMD apresenta, em um elevado grau de abstração, os relacionamentos intra-domínio (*i.e.* As subordinações existentes entre os metadados de um domínio. Estas subordinações definem a hierarquia na qual os metadados serão organizados, por exemplo, *ÁreaGeográfica* é subordinada de *BancoDeDadosGeográficos* (ver figura 3.3) porque para esta existir deverá necessariamente existir um *BancoDeDadosGeográficos*, o qual irá armazenar as áreas geográficas) e extra-domínio (*i.e.* As associações existentes entre os metadados dos dois domínios) dos ambientes de processamento geográfico e processamento analítico. Isto facilita a compreensão sobre a integração dos dois ambientes que, ao mesmo tempo, possuem pontos de interseção e metadados com nomenclaturas divergentes.

GeoMD usa a notação de UML (*Unified Modeling Language*) [BRJ99] para expressar seus domínios, metadados e relacionamentos. Os domínios de GeoMD agrupam seus metadados segundo a notação de pacote e posteriormente os modelam segundo a notação de classes sem atributos e operações. A figura 3.3 apresenta os relacionamentos e subordinações existentes entre o ambiente analítico (OLAP) e geográfico (SIG), como descrito pelo esquema GeoMD.

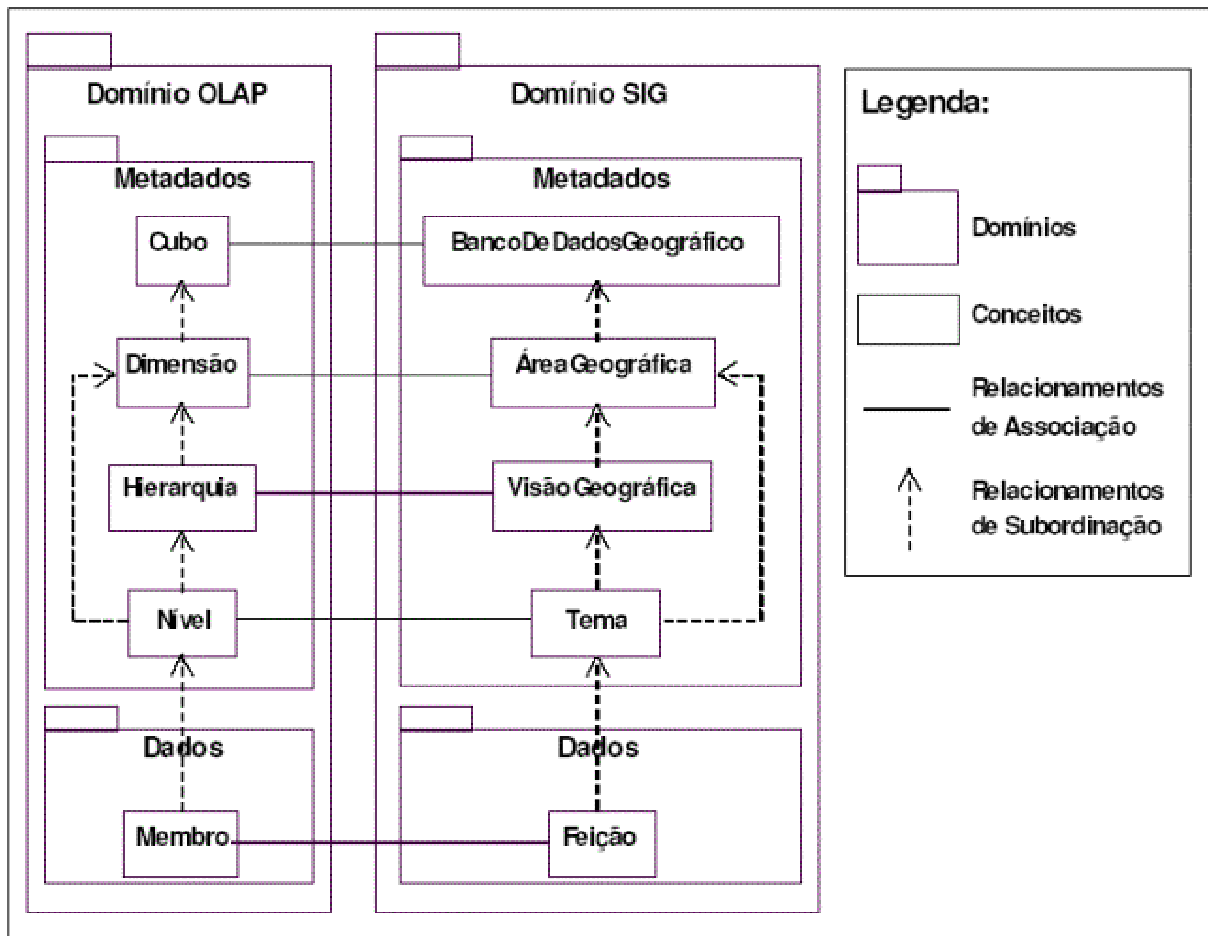


Figura 3.3: O Esquema Abstrato GeoMD [Fid03].

O modelo de implementação do esquema GeoMD pode ser visualizado na figura 3.4. Este modelo é denominado de GeoMDM e foi definido originalmente em [Fid03]. Ele representa os relacionamentos existentes entre os metadados armazenados em uma fonte de metadados, a qual disponibiliza informações que indicam os relacionamentos existentes entre os dados analíticos e geográficos. Através desses relacionamentos, é possível verificar a existência de correspondências entre dados analíticos e geográficos, bem como possibilitar a recuperação dos dados que satisfazem estas correspondências.

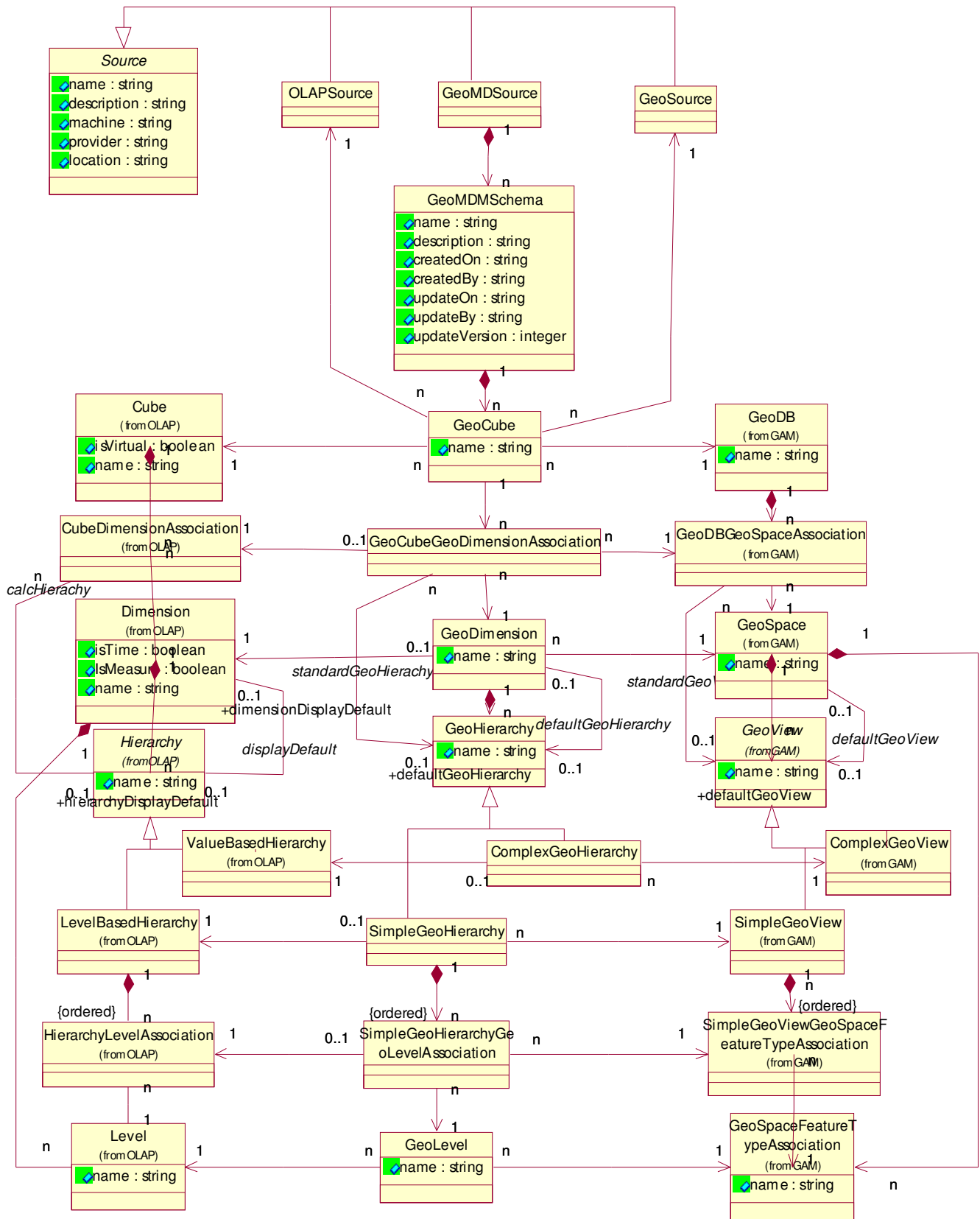


Figura 3.4: Modelo de Implementação do Metamodelo GeoMDM [Fid03].

3.3.1.4 GeoDWFrame

Uma vez que a arquitetura GOLAPA está voltada para a utilização de um *Data Warehouse* Geográfico, nesta seção será descrito brevemente o arcabouço GeoDWFrame (*Geographical DW Frame*) [FST+03b, Fid03]. Ele tem objetivo de definir um conjunto de conceitos e princípios de projeto para a construção do esquema de um DWG. Segundo [Fid03], um DWG trata-se de uma extensão de um DW tradicional focada para suporte a dados geográficos. Ou seja, estende-se o modelo estrela pela inserção de propriedades geográficas, as quais são normalmente definidas em uma única dimensão geográfica, possibilitando assim, além das funcionalidades tradicionais de um DW, a agregação em mapas ou tabelas de dados geo-referenciados. A figura 3.5 esboça um esquema de DWG em discussão. Nesta, as dimensões Loja, Cliente e Fornecedor compartilham o mesmo contexto espacial definido pela dimensão Geográfica.

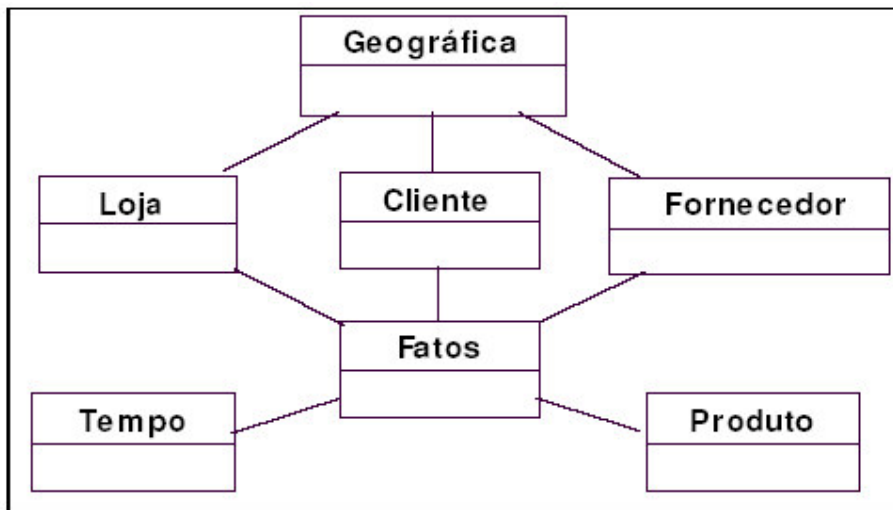


Figura 3.5: Exemplo de um esquema estrela compartilhando o mesmo contexto espacial.

GeoDWFrame classifica as dimensões de um *DWG* como Convencionais, Geográficas e Híbridas. Entretanto, *GeoDWFrame* subclassifica as dimensões Geográficas em Primitivas e Compostas e as dimensões Híbridas em Micro, Macro e Conjunta. Uma dimensão Convencional contém em todos os seus níveis, apenas dados convencionais. Por exemplo, uma dimensão Produto com os campos Marca, Tipo, Categoria e Subcategoria. Por outro lado, uma dimensão Geográfica contém em todos os seus níveis, apenas dados geográficos que possuem referência espacial. Por exemplo, uma dimensão Localização com os campos País, Região, Estado, Município, Endereço e suas respectivas coordenadas. É considerado um dado

convencional, todo atributo descritivo sobre um objeto geográfico que não possua referência espacial. Isto é, o exemplo anterior sem coordenadas. Quanto à dimensão Híbrida, esta contém em seus níveis tanto dados convencionais como geográficos. Por exemplo, uma dimensão Cliente com os níveis País, Estado, Município, Sexo, Faixa Etária, CPF, Endereço e suas respectivas coordenadas.

No modelo de DWG definido por [Fid03], uma dimensão Geográfica Primitiva contém apenas um campo para a chave primária e os outros com as coordenadas geográficas de determinado tema. Uma dimensão Geográfica Composta contém um conjunto de atributos descritivos sobre dados geográficos, sua chave primária e as chaves estrangeiras para suas dimensões primitivas.

Uma dimensão Híbrida Micro é aquela que possui dados convencionais e dados pontuais ou lineares (*e.g.* temas sobre endereços e rodovias). Neste tipo de dimensão, os temas são representados por campos descritivos mais sua chave estrangeira para a respectiva dimensão geográfica. Uma dimensão Híbrida Macro é aquela que possui dados convencionais e dados geográficos poligonais (*e.g.* temas sobre regiões, estados ou municípios). Nesta abordagem, os dados geográficos são representados por sua chave estrangeira para uma determinada dimensão geográfica composta. Finalmente, uma dimensão Híbrida Conjunta combina as duas abordagens anteriores.

3.4 Considerações Finais

Este capítulo, basicamente, apresentou as pesquisas que possuem pontos em comum com a presente dissertação. Um dos pontos importantes que foi apresentado foi a Arquitetura GOLAPA, juntamente com a descrição de suas camadas e principais componentes. A importância da arquitetura GOLAPA para o presente trabalho se deve ao fato das pesquisas aqui apresentadas estarem voltadas, principalmente, para o componente GOLAPE, da referida arquitetura. Foram também analisadas outras propostas intencionadas a fornecer um ambiente para processamento geográfico-multidimensional. Entretanto, estas propostas possuem algumas deficiências, que serviram de motivação para a proposição de GOLAPA.

De forma resumida, na tabela 3.1, é apresentado um estudo comparativo entre as abordagens listadas neste capítulo. Alguns dos pontos analisados foram: (1) a possibilidade da

utilização de operadores espaciais nas consultas que envolvem dados analíticos e geográficos; (2) as tecnologias utilizadas, ou seja, se a abordagem está voltada para a utilização de padrões abertos e independentes de plataforma, ou se está atrelada a uma única plataforma e faz uso de soluções proprietárias; (3) a utilização de metadados para guiar o processo de integração, indicando as correspondências existentes entre os dados analíticos e geográficos; e (4) utilização de um *Data Warehouse* Geográfico como repositório único de dados integrados e não voláteis, mantendo informações históricas dos dados oriundos das fontes analíticas e geográficas. Quanto às operações OLAP sobre dados analíticos, vale salientar que todas as abordagens oferecem suporte.

Tabela 3.1: Comparação entre as propostas para processamento geográfico-multidimensional.

	Operações Espaciais	Tecnologias Utilizadas	Uso de Metadados	Uso de DWG
Map Cube	Não	Não informado	Não	Sim
GeoMiner	Sim	Utiliza soluções proprietárias	Não	Sim
GOAL	Não	Limitada ao sistema operacional Windows e não utiliza padrões abertos	Sim	Não
SIGOLAP	Sim	Utiliza tecnologias proprietárias e está fortemente atrelada ao Windows	Sim	Não
GOLAPA	Sim	Voltada para a utilização de padrões abertos e extensíveis	Sim	Sim

Como pode ser observado na tabela 3.1, a arquitetura GOLAPA se preocupa com questões não abordadas pelas outras propostas. Um ponto forte é a utilização de padrões abertos e extensíveis, o que torna a solução independente de plataforma e possibilita maior interoperabilidade no intercâmbio dos seus dados. Este fato é ainda mais evidente com a utilização de um *Data Warehouse* Geográfico, que contém os dados geográficos descritos em GML, que é um padrão bastante utilizado para armazenamento, descrição e intercâmbio de dados geográficos. Ressalta-se que estes pontos ficarão evidentes nos próximos capítulos, onde serão discutidos pontos relacionados à definição e implementação de alguns dos componentes da referida arquitetura.

Os conceitos apresentados neste capítulo, sobre a arquitetura GOLAPA, como por exemplo metamodelo GAM e o esquema GeoMD, foram empregados no desenvolvimento do trabalho apresentado na presente dissertação, mais precisamente na implementação do componente GOLAPE, descrito no capítulo 6. No próximo capítulo, são apresentadas as tecnologias utilizadas no processo de integração do processamento analítico e geográfico. Vale

salientar ainda, que todos estes padrões apresentados no capítulo 4 são abertos, satisfazendo os objetivos de padronização, extensibilidade e independência de plataforma, da arquitetura GOLAPA.

Capítulo 4

XML e Serviços Web

Neste capítulo, serão apresentados a linguagem de marcação XML e alguns dos padrões definidos a partir dela, como por exemplo, o *XML Schema*, *XSLT*, *GML*, *SVG*, *Web Services*, *XML For Analysis e Web Feature Service*, entre outros. Estas tecnologias são consideradas os principais elementos que possibilitaram a integração de serviços analíticos e geográficos na *Web*.

4.1 Introdução

Para possibilitar a integração de serviços analíticos e geográficos na *Web*, inúmeros padrões tecnológicos foram estudados. O intuito era utilizar padrões abertos e extensíveis na implementação de uma solução para o problema. Nesse sentido, este capítulo apresenta uma relação de conceitos relacionados às tecnologias que podem ser consideradas elementos fundamentais para o processo de integração de serviços analíticos e geográficos na *Web*.

As vantagens oferecidas por estas tecnologias, no contexto do trabalho apresentado nesta dissertação, devem-se ao fato de todas elas serem definidas como padrões abertos, extensíveis e voltados para a Internet. Padrões como XML foram definidos de forma extensível e hoje possibilitam a definição de outros padrões a partir dele, como é o caso do SVG, GML, *Web Services* e muitos outros. Esses fatos, entre muitos outros, motivam fortemente a utilização dos conceitos apresentados nas próximas seções na proposição de soluções para a integração de serviços para processamento analítico e geográfico na *Web*.

Procurando introduzir algumas das tecnologias envolvidas neste trabalho, as próximas seções desse capítulo estão organizadas da seguinte maneira: A seção 4.2 apresenta alguns conceitos pertinentes a XML e padrões relacionados. Na seqüência, na seção 4.3, é discorrido sobre os *Web Services*. A seção 4.4 aborda *XML For Analysis* e a seção 4.5 é composta por tópicos relacionados à especificação *Web Feature Service*. Finalmente, na seção 4.6, são apresentadas algumas considerações finais sobre este capítulo.

4.2 XML e Padrões Relacionados

Nesta seção, serão apresentados alguns conceitos relacionados à linguagem XML (*Extensible Markup Language*) [XML00], e também, aspectos pertinentes a outros padrões definidos a partir dela.

4.2.1 A Linguagem XML

XML é uma linguagem definida pelo *World Wide Web Consortium* (W3C) [W3C03], o órgão que define os padrões para a *Web*. Ela teve origem principalmente pela necessidade de uma melhora no HTML (*Hyper Text Markup Language*) [HTTP03] e uma simplificação no SGML (*Standard Generalized Markup Language*) [SGML03]. O SGML, que apesar de ser uma linguagem poderosa, não teve muita aceitação devido sua complexidade. Já o HTML, que se expandiu facilmente e teve uma boa aceitação, começou a apresentar deficiências, principalmente no desenvolvimento de aplicações *Web* mais complexas. Quando o HTML foi desenvolvido, tinha-se apenas a preocupação de se obter uma linguagem que pudesse combinar texto, imagens e botões em um navegador para a Internet (*e.g. Microsoft Internet Explorer* [MIE03], *Netscape Navigator* [NNA03]), sem a preocupação com o intercâmbio de informações pela *Web*.

Com isso foi criada a linguagem XML, uma linguagem de descrição de documentos, baseada no SGML, mas que herda algumas características do HTML e contém outras funcionalidades que foram especificamente desenvolvidas para serem usadas na Internet. Com o XML pretendeu-se aproveitar os benefícios do SGML e, simultaneamente, torná-lo mais simples, mantê-lo leve e otimizá-lo para a *Web*.

Com a concepção do XML, como sendo uma linguagem de marcação extensível e tendo uma boa disseminação, possibilitou-se uma maior interoperabilidade entre aplicações, sem falar nas vantagens adquiridas quanto ao intercâmbio de informações na *Web*. Dessa forma, inúmeros outros padrões foram especificados a partir de XML, entre eles podemos citar *XML Schema* [SCHE03], *DTD* [DTD98], *XSLT* [XSLT03], *GML* [GML03a], *XPath* [XPA03], *XLink* [XLI03], *XPointer* [XPO03], *SVG* [SVG03], *XMLA* [XMLA02], *WFS* [WFS02] e *Web Services* [WSA02]. Uma lista de especificações elaboradas a partir dos princípios definidos pelo XML

pode ser encontrada em [W3C03]. Além disso, em [XML03b] é possível encontrar uma relação de organizações que produzem suas próprias linguagens baseadas em XML.

4.2.2 XML Schema

Quanto à sua composição, um documento XML pode ser classificado como bem formado ou mal formado, válido ou inválido. Um documento bem formado é aquele que obedece às regras de sintaxe estabelecidas pelo W3C na especificação XML. Ou seja, para um documento ser considerado bem formado ele precisa conter um ou mais elementos, e um deles, o chamado elemento raiz, precisa conter todos os outros elementos. Cada elemento precisa também ser aninhado corretamente dentro de quaisquer elementos delimitadores, sem que haja sobreposição de *tags*. Conseqüentemente, um documento mal formado é aquele que não obedece à sintaxe XML. Por outro lado, um documento é considerado válido, se possui associado a ele algum documento DTD (*Document Type Definition*) [DTD98] ou um *XML Schema* [SCHE03], os quais são utilizados para definir aspectos relacionados à estrutura e ao tipo dos elementos de um documento XML. Obviamente, o documento XML precisa obedecer às regras estabelecidas pela DTD ou *XML Schema* para ser considerado válido. Dessa forma, um documento XML é considerado inválido se não estiver conforme a definição contida em sua DTD ou *XML Schema*.

XML Schema é uma especificação oficialmente recomendada pelo W3C para definição de esquemas de aplicações XML. Como declara [Hol01], *XML Schema* representa um avanço significativo frente a DTD, pois não permitem apenas especificar a sintaxe de um documento, como faria uma DTD, mas também eles podem definir os tipos de dados reais do conteúdo de cada elemento, herdar as definições de outros esquemas, usar esquemas com vários *namespaces* [XNa03], criar tipos de dados simples e complexos, especificar o número mínimo e o número máximo de vezes que um elemento pode ocorrer, criar tipos de lista, criar grupos de atributos, restringir os intervalos dos valores que os elementos podem conter, mesclar fragmentos de vários esquemas, exigir que os valores de atributo ou elemento sejam exclusivos, entre outras vantagens.

A figura 4.1 contém um pequeno exemplo definido em *XML Schema*. Nesse esquema, é definido um elemento principal denominado *contatos*. Neste exemplo, foram definidos dois tipos complexos, *TipoContato* e *TipoEndereco*. Um elemento do tipo *TipoEndereco* pode conter os subelementos *rua*, *bairro*, *cep*, *cidade* e *UF*, todos do tipo *string*, sendo que o conteúdo do sub elemento *UF* fica restrito a apenas 2 caracteres. Um elemento do tipo *TipoContato*, pode conter os subelementos *nome*, *email* e *fone*, todos do tipo *string*, e um subelemento do tipo *TipoEndereco*. A figura 4.2, gerada a partir da ferramenta *XML Spy* [XSPY03], apresenta uma representação gráfica do *XML Schema* apresentado na figura 4.1. Na figura 4.3, é apresentado um exemplo de documento XML que obedece a estrutura definida pelo esquema anterior.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xsd:element name="contatos">
    <xsd:complexType>
      <xsd:sequence maxOccurs="unbounded">
        <xsd:element name="contato" type="TipoContato"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:complexType name="TipoEndereco">
    <xsd:sequence>
      <xsd:element name="rua" type="xsd:string"/>
      <xsd:element name="bairro" type="xsd:string"/>
      <xsd:element name="cep" type="xsd:string"/>
      <xsd:element name="cidade" type="xsd:string"/>
      <xsd:element name="UF">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:length value="2"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="TipoContato">
    <xsd:sequence>
      <xsd:element name="nome" type="xsd:string"/>
      <xsd:element name="email" type="xsd:string"/>
      <xsd:element name="fone" type="xsd:string"/>
      <xsd:element name="endereco" type="TipoEndereco"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

Figura 4.1: Exemplo de *XML Schema*.

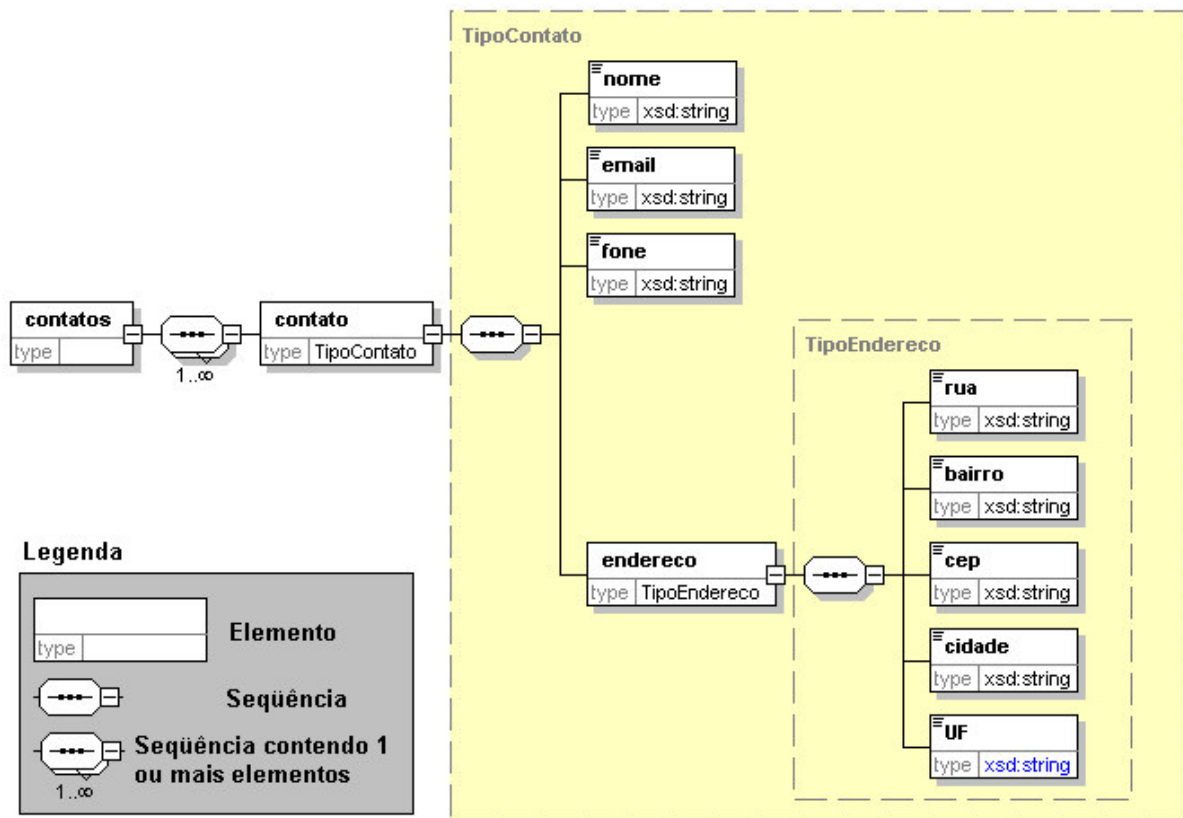


Figura 4.2: Representação gráfica do Esquema apresentado na figura 4.1.

```
<?xml version="1.0" encoding="UTF-8"?>
<contatos xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="contatos.xsd">
  <contato>
    <nome>João</nome>
    <email>joao@joao.com.br</email>
    <fone>99 9999 9999</fone>
    <endereco>
      <rua>Nome da Rua</rua>
      <bairro>Nome do Bairro</bairro>
      <cep>99999-000 </cep>
      <cidade>Nome da Cidade</cidade>
      <UF>UF</UF>
    </endereco>
  </contato>
</contatos>
```

Figura 4.3: Documento XML que obedece a estrutura definida pelo XML Schema apresentado na figura 4.1.

4.2.3 XSLT - XSL Transformations

A XSL (*Extensible Stylesheet Language*) [XSL03] possui duas partes: uma linguagem de transformação e outra de formatação. A linguagem de transformação permite alterar os documentos XML para outras formas, enquanto a linguagem de formatação configura e estiliza documentos de várias maneiras. Essas duas partes da XSL funcionam de forma bastante independente e pode-se pensar em XSL como sendo duas linguagens, e não uma. Na prática, inicialmente se transforma um documento e posteriormente formata-o, e esse é um dos principais motivos para o W3C considerar a XSLT (*XML Stylesheet Language for Transformations*) [XSLT03] como primeiro estágio no processo de formatação [Hol01]. Neste documento, será dada ênfase a linguagem XSLT, entretanto, maiores informações sobre a parte de formatação de objetos XSL, ou XSL-FO, podem ser encontradas em [XSL03] e [XSLFO03].

A XSLT é utilizada para manipular documentos, alterando e trabalhando com sua marcação como desejar. Uma das transformações mais comuns é de documentos XML para documentos HTML, entretanto, pode-se utilizar a XSLT para transformar arquivos XML em outro arquivo XML, com uma estrutura diferente da original, ou ainda para formatos DOC, PostScript e PDF.

Para criar uma transformação são necessários dois documentos, ou seja, o documento a transformar e a folha de estilo que especifica a transformação, sendo que ambos devem ser documentos XML bem formados. Para melhor compreensão do processo de transformação, na figura 4.4 são apresentados os componentes de um processo de transformação, onde são utilizados um documento XML de entrada e um arquivo XSLT para gerar um outro documento XML com uma estrutura diferente. No processo em questão, os arquivos XML e XSLT, que podem ser vistos como uma árvore de nodos, são submetidos a um processador XSLT que, de acordo com as regras de transformação estabelecidas no documento XSLT, irá produzir um novo documento como resultado. Exemplos de processadores XSLT são o Xalan (versão para Java [XJ03] e para C++[XC03]) e o XT [XT03]. Existem também processadores XSLT embutidos em *parsers* XML como o MSXML 4.0 [MSXML03], e no *parser* XML da ferramenta XML Spy [XSPY03]. O *Browser Web Internet Explorer* [MIE03] também possui uma espécie de processador XSLT, porém nas versões atuais (*i.e.* versão 6 e antecessoras) permite somente transformações simples, de XML para HTML.

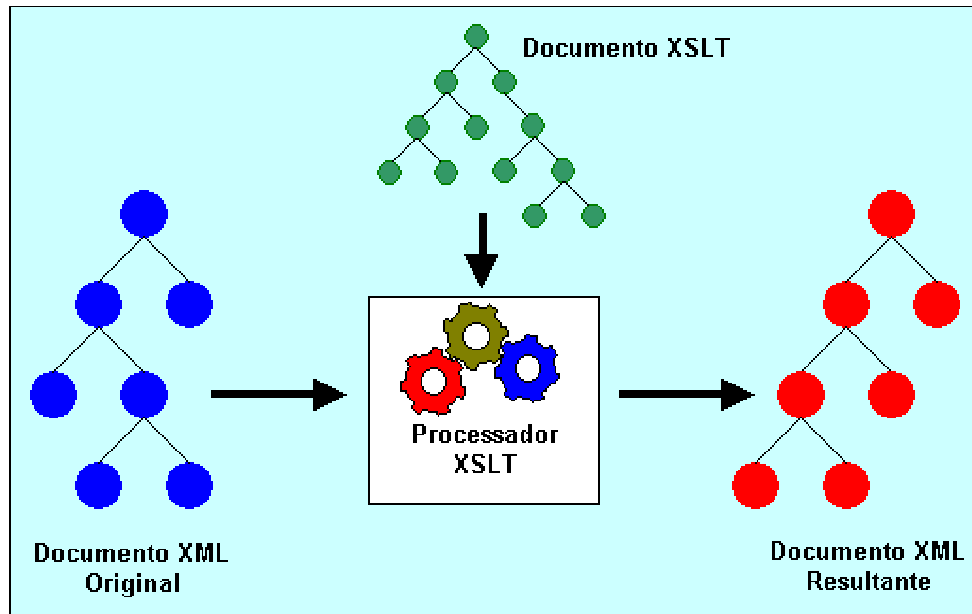


Figura 4.4: Componentes de um processo de transformação XSLT.

4.2.4 GML - *Geography Markup Language*

A linguagem GML (*Geography Markup Language*) [GML03a] foi desenvolvida pelo OGC (*Open GIS Consortium*) [OGC03] para possibilitar a codificação de dados geográficos em XML, permitindo tanto o armazenamento quanto o transporte de informações espaciais em formato XML. A especificação GML baseia-se no modelo abstrato do OpenGIS [OGC99], que define uma feição geográfica como sendo uma abstração de um fenômeno do mundo real que está associada a uma localização relativa à superfície da Terra. Assim sendo, pode-se considerar que a representação digital das localizações do mundo real pode ser feita através de um conjunto de feições, onde o estado de uma feição é definido por um conjunto de atributos, onde cada propriedade é uma tripla {nome, tipo, valor}.

Desde sua proposta inicial, a versão 1.0 de 12/05/2000, até a data de escrita deste documento, a *GML* já sofreu quatro atualizações. Como a versão 3.0 foi lançada em 29 de janeiro de 2003 e existem poucas implementações da mesma, optou-se por discutir e utilizar sua predecessora (*i.e.* a versão 2.1.2). Isto porque ela possui implementação, os documentos *XML* gerados por ela são conformes à versão 3.0 [GML03b] e, sobretudo, porque trata-se de uma versão testada e estabilizada.

A GML 2.1.2 só suporta geo-dados vetoriais (*i.e.* ponto, linha e polígono) e seu XML Schema corresponde à descrição de um meta-esquema, o qual oferece um conjunto de elementos/atributos que servem de base à construção do esquema de uma aplicação geográfica.

Com feições geográficas do tipo polígono, linha e ponto, é possível representar, respectivamente, as bordas de uma determinada cidade, a geometria dos rios que cortam esta cidade, e a localização de escolas, aeroportos e hospitais dentro dos limites desta cidade. Para exemplificação, na figura 4.5, é possível visualizar um trecho de código que descreve uma Loja, com duas propriedades descritivas, *nome* e *endereço*, e a propriedade geométrica *location* definida por um elemento *Point* de GML.

```
<Loja>
  <nome> Loja Matriz</nome>
  <endereco> Av. sem nome, 999</endereco>
  <gml:location>
    <gml:Point srsName="http://www.opengis.net/epsg.xml#4326">
      <gml:coord>
        <gml:X>25.0</gml:X>
        <gml:Y>30.0</gml:Y>
      </gml:coord>
    </gml:Point>
  </gml:location>
</Loja>
```

Figura 4.5: Exemplo da utilização de propriedades geométricas em GML.

4.2.5 SVG - Scalable Vector Graphics

SVG [SVG03], trata-se de uma gramática XML recomendada pelo W3C [W3C03], que permite a descrição de gráficos vetoriais em duas dimensões, textos e imagens formato raster [MGR91].

Com a utilização de SVG é possível a construção de gráficos de alta resolução, com os quais o usuário pode interagir. SVG permite que, por exemplo, através de eventos de mouse, um objeto gráfico ou grupo de objetos seja selecionado e determinadas ações sejam executadas.

Os visualizadores SVG, como o *SVGViewer* da Adobe [SVG03], utilizam o sistema cartesiano de coordenadas e desenharam a imagem na tela à medida que o código SVG vai sendo interpretado, permitindo assim, a sobreposição de imagens e a criação de efeitos de transparência. Geralmente, os gráficos SVG são visualizados em *browsers Web*, como o *Internet Explorer* [MIE03], entretanto, se faz necessário instalar um *plugin SVG* [SVG03] para que isso

ocorra. Porém, em virtude da grande aceitação e disseminação do SVG, é provável que futuras versões dos *browsers Web* sejam distribuídas com esta funcionalidade acoplada.

No contexto do presente trabalho, a tecnologia SVG exerce uma importante função. No processo de integração de serviços analíticos e geográficos na *Web*, o SVG foi utilizado para desenhar os mapas, resultantes de consultas para suporte à decisão na *Web*. Isso foi possível através da conversão de feições geográficas armazenadas em GML, para objetos gráficos descritos em SVG.

Um exemplo da utilização de SVG para construção de um mapa é apresentado na figura 4.6, a qual apresenta o mapa do Brasil, com a divisão por estados. Neste caso, a geometria de cada estado foi representada por um objeto gráfico em SVG e o nome do estado é representado por uma legenda. Para a visualização, foi utilizado o *browser Web Internet Explorer* com o *plugin SVG SVGViewer* [SVG03] da Adobe.



Figura 4.6: Exemplo da utilização de SVG para desenhar um mapa.

4.3 Web Services

Conforme relatado em [WSA02] pelo W3C (*World Wide Web Consortium*) [W3C03], *Web Service* pode ser descrito como sendo um sistema de software identificado por uma URI (*Uniform Resource Identifier*), o qual publica interfaces onde ligações são definidas e descritas usando XML (*eXtensible Markup Language*) [XML03a]. Estas definições de serviços podem ser descobertas por outros sistemas de software. Estes sistemas podem então interagir com um *Web Service* de uma maneira prescrita por essa definição, usando mensagens baseadas em XML que são conduzidas por protocolos da Internet.

Como descrito anteriormente, *Web Services* constituem uma arquitetura computacional distribuída e baseada em serviços autodescritivos que podem ser publicados, localizados e executados através da *Web*. Os desenvolvedores de *Web Services* não precisam ter conhecimento sobre a plataforma, modelo de objeto, ou a linguagem de programação que foi usada para implementar o serviço. Eles só precisam saber como solicitar o serviço para que possam receber a resposta da solicitação adequadamente [AF03].

A necessidade de integrar fontes heterogêneas de informação, a distribuição de aplicações via *Web*, com intuito de substituir a arquitetura cliente-servidor tradicional, e o fato de XML ser cada vez mais utilizada como um suporte neutro de informação, servem como fatores de motivação para a implementação de *Web Services*.

4.3.1 Implementação de *Web Services*

Os aplicativos acessam um *Web Service* utilizando os protocolos padrões da *Web*, sem necessidade de se preocuparem como cada *Web Service* foi implementado. Todos esses padrões são definidos pelo W3C, tornando a sua implementação viável em qualquer plataforma. Nas subseções a seguir, temos a descrição dos quatro padrões básicos utilizados para a implementação de um *Web Service*.

4.3.1.1 XML

XML tornou-se o padrão para definir formatos de troca de informação na Internet. É similar ao HTML (*Hyper Text Markup Language*), já que usa *tags* para codificar informação. A mais importante característica do XML se resume em separar a interface com o usuário (apresentação) dos dados estruturados. Alguns conceitos sobre XML foram apresentados na seção 4.2.

4.3.1.2 SOAP

O protocolo SOAP (*Simple Object Application Protocol*) [SOAP03] é baseado em XML e serve para trocar informações em ambientes distribuídos. Consiste de um envelope que define um arcabouço para descrever o que consta na mensagem e como processá-la, codifica regras de negócio e tem uma convenção para chamadas e respostas de procedimentos.

Para se manipular SOAP é necessário um *Parser XML* e um *Servidor Web* que rode HTTP (*Hyper Text Transfer Protocol*). SOAP é uma especificação do W3C patrocinado pela *Microsoft*⁵, *IBM*⁶, entre outros.

Na troca de informações entre um *Web Service* cliente e um provedor de serviços, as mensagens são codificadas em XML, obedecendo ao formato SOAP e enviadas ao provedor. Quando o provedor do(s) serviço(s) recebe este envelope SOAP, ele o interpreta recuperando os parâmetros necessários para a execução da requisição. O resultado do processamento é codificado em um envelope SOAP e enviado para a aplicação que efetuou a requisição. No caso de ocorrer algum erro de processamento no servidor, um envelope SOAP contendo uma mensagem de erro é enviada para a aplicação solicitadora.

4.3.1.3 UDDI

UDDI (*Universal Description, Discover and Integration*) permite que as demais aplicações tenham conhecimento das operações e integrem as funcionalidades que o *Web Service* está dispondo. UDDI é um diretório para *Web Services*. Para transformar a atual *Web* semi-estática numa *Web* dinâmica, dotada de serviços colaborativos, é necessário criar um diretório mundial

⁵ Microsoft Corporation, <http://www.microsoft.com>

⁶ IBM Corporation, <http://www.ibm.com>

para consultas sobre os serviços disponíveis, possibilitando que empresas divulguem e pesquisem serviços de forma dinâmica.

Tal como uma lista telefônica, o UDDI contém páginas brancas e amarelas que permitem procurar os serviços em função do nome da empresa ou do tipo de atividade associada. Porém, o UDDI oferece também páginas verdes, que indicam como negociar com as empresas que figuram no diretório, designando, por exemplo, os seus procedimentos comerciais ou descrevendo os serviços disponíveis. Brancas, amarelas ou verdes, as páginas UDDI possuem um modelo de dados baseado em XML, e o seu acesso exige a utilização do SOAP, tanto para as preencher como para pesquisar os serviços que as contêm [UDDI03].

4.3.1.4 WSDL

Um documento WSDL (*Web Services Description Language*) permite aos desenvolvedores, expor a sintaxe de um *Web Service*. Em formato XML, este documento descreve serviços de rede como um conjunto de métodos orientados a documentos ou procedimentos. Proporcionam às aplicações cliente, todos os detalhes necessários para conectar e usar o *Web Service* [WSDL01].

Normalmente, o fornecedor publica seus serviços no UDDI por meio de um documento WSDL, o qual contém as informações necessárias para que uma aplicação cliente seja implementada para trocar informações com o provedor do serviço. Neste documento WSDL, são apresentadas informações como a localização do serviço, operações disponíveis e tipo dos parâmetros de entrada e saída.

Com as informações fornecidas pelo documento WSDL, uma aplicação pode ser implementada em qualquer linguagem de programação, sendo apenas necessário que essa linguagem suporte o envio e recebimento de mensagens em formato SOAP. Esta é uma das grandes vantagens dos *Web Services*, pois provê total independência de plataforma e de linguagem de programação, o que contrasta altamente com a tradicional arquitetura Cliente/Servidor, onde, geralmente, a aplicação cliente fica atrelada às tecnologias em que o servidor foi implementado.

4.3.2 Arquitetura dos Web Services

A arquitetura básica dos *Web Services* é constituída pelas tecnologias descritas acima (*i.e.* XML, SOAP, WSDL, UDDI). As quais possibilitam a troca de mensagens, descrição de serviços, publicação e recuperação de descrições de *Web Services*.

A arquitetura básica dos *Web Services* define uma interação entre agentes de *software* através de uma troca de mensagens entre consumidores e provedores de serviços. Solicitadores são agentes de *software* que requerem a execução de um serviço. Provedores são agentes de *software* que provêem um serviço. Esses agentes de *software* podem ser tanto solicitadores como provedores de serviços. Os provedores são responsáveis por publicar a descrição do(s) serviço(s) provido(s), enquanto que os solicitadores precisam estar habilitados a recuperar a descrição dos serviços [WSA02].

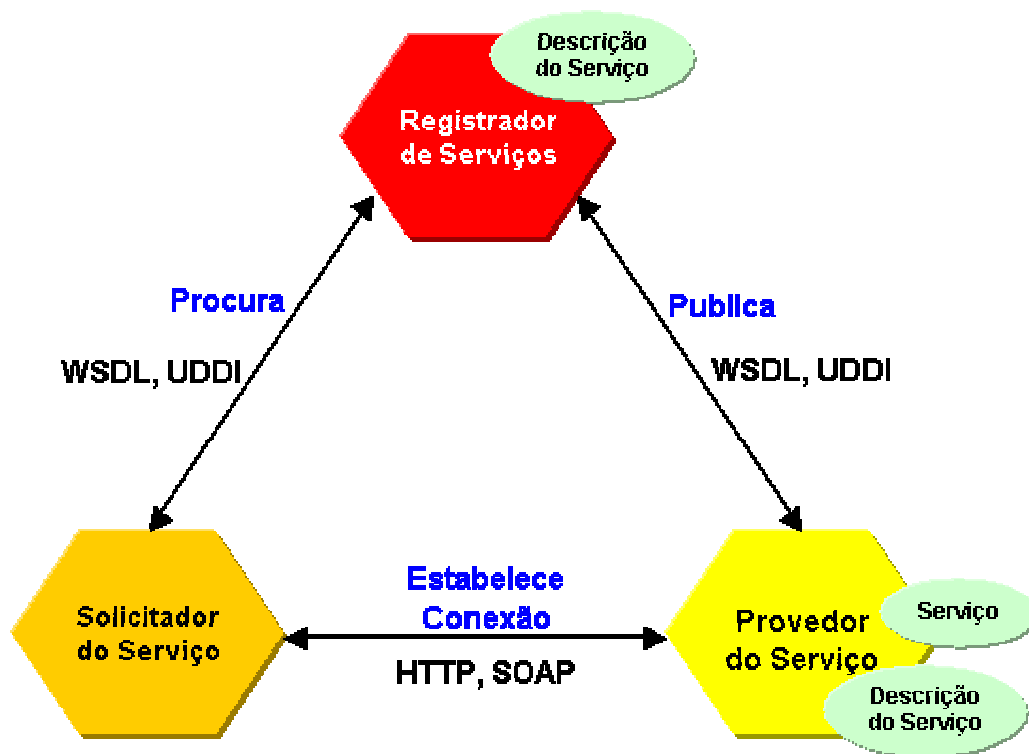


Figura 4.7: Arquitetura de um *Web Service*, seus componentes, operações e artefatos (adaptado de [WSCA01]).

Na figura 4.7, encontra-se a representação dos componentes, funções e artefatos da arquitetura dos *Web Services*. Nela podemos visualizar que há uma interação entre componentes, onde operações como publicação, busca e estabelecimento da conexão são realizadas.

A função do provedor é fornecer o serviço e publicar a descrição (WSDL) no UDDI. No UDDI, todas as descrições de serviços ficam registradas, permitindo que solicitadores de serviços pesquisem por um determinado serviço.

Assim que o solicitador de serviços encontra no UDDI um serviço que satisfaz suas necessidades ele recupera a descrição (*i.e.* documento WSDL). Com as informações contidas neste documento, estabelece a conexão com o provedor do serviço e iniciam a troca de mensagens através do protocolo HTTP obedecendo ao formato SOAP. Como pode-se notar, os artefatos principais desta arquitetura são o serviço oferecido e a descrição do mesmo, a qual é realizada através do WSDL.

4.4 XMLA - XML For Analysis

XML For Analysis [XMLA02, XMLA03] é uma API XML [XML03a] baseada em SOAP [SOAP03], e criada através de uma iniciativa das empresas *Microsoft Corporation* e *Hyperion Solutions Corporation*⁷, com intuito de proporcionar um acesso aberto a bancos de dados multidimensionais, padronizando assim interação entre aplicações cliente e servidores de dados OLAP (*OnLine Analytical Processing*) [CD96] através da Internet.

Com as técnicas tradicionais de acesso a dados, tal como as baseadas em OLE DB [OLE03] e ODBC [ODBC03], um componente cliente que é fortemente amarrado ao componente provedor de dados, precisa ser instalado na máquina cliente para acessar os dados do servidor. Essa forte ligação pode criar dependências entre os componentes como uma plataforma específica de *hardware*, uma linguagem de programação específica, e versões dos componentes cliente e servidor [XMLA02]. Este tipo de arquitetura, fortemente acoplada e com dependências entre cliente e servidor, impossibilita uma livre interação para acesso a dados através da Internet, que seja ao mesmo tempo independente de plataforma, linguagem de programação ou de qualquer outra restrição.

⁷ Hyperion Solutions Corporation , www.hyperion.com

Para sanar esta deficiência das técnicas atuais de acesso a dados, a especificação do *XML For Analysis* [XMLA02] foi construída em conformidade com padrões abertos da Internet como HTTP [HTTP03], XML [XML03a] e SOAP [SOAP03]. A especificação define dois métodos, *Discover* e *Execute*, que serão analisados com mais detalhes a seguir, e possibilitam o envio e recebimento de mensagens em formato XML para descoberta e manipulação de dados, sem que ocorram alterações nos dados do servidor.

4.4.1 Implementação do *XML For Analysis*

A implementação da especificação de *XML For Analysis* geralmente é feita baseada na arquitetura dos *Web Services* [WSA02] (ver seção 4.3), onde um servidor disponibiliza um documento que descreve os serviços oferecidos por ele. Esse documento é definido conforme a especificação WSDL [WSDL01], onde constam informações como localização do serviço, métodos acessíveis, parâmetros necessários, tipo dos parâmetros e das respostas. De posse desse documento, pode ser implementado um *Web Service* cliente, para interagir com o serviço *Web* através do envio e recebimento de mensagens no formato SOAP [SOAP03]. É importante salientar que a implementação do *Web Service* cliente é independente de linguagem ou plataforma.

A API de comunicação baseada em XML, denominada *XML For Analysis* disponibiliza dois métodos para acesso: *Discover* e *Execute*. Estes métodos são invocados usando o protocolo SOAP [SOAP03], dessa forma, as entradas e saídas estão no formato XML. O XML *namespace* [XNa03] utilizado por estes métodos é `"urn:schemas-microsoft-com:xml-analysis"`.

4.4.1.1 Método *Discover*

O método *Discover* é utilizado para obter informações e metadados do *Web Service*. Tais informações podem incluir uma lista de fontes de dados disponíveis e dados sobre o provedor para uma fonte de dados em particular. São utilizadas propriedades para definir quais dados serão obtidos, bem como o formato desses dados. Devido a essa interface genérica e o uso de propriedades, consegue-se extensibilidade sem a necessidade de reescrever funções existentes [XMLA02].

Uma chamada ao método *Discover* pode ser feita de acordo com a seguinte sintaxe:

```
Discover (  
    [parâmetro de entrada] RequestType As EnumString,  
    [parâmetro de entrada] Restrictions As Restrictions,  
    [parâmetro de entrada] Properties As Properties,  
    [resultado da requisição] Result As Rowset)
```

Os parâmetros envolvidos em uma chamada ao método *Discover* podem ser descritos da seguinte forma:

- ***RequestType***: Este parâmetro consiste de um valor de enumeração *RequestTypes*, que especifica o tipo de informação que será retornada. É utilizado pelo método *Discover* para determinar a estrutura do conteúdo retornado no parâmetro *Result*. Por exemplo, o valor enumerado *DISCOVER_DATASOURCES* retorna uma lista de fontes de dados disponíveis no servidor ou *Web Service*;
- ***Restrictions***: Este parâmetro precisa ser incluído, mas pode ser vazio. Ele é utilizado para restringir os dados que serão retornados no parâmetro *Result*;
- ***Properties***: Este parâmetro também precisa ser incluído, mas pode ser vazio. Ele consiste em uma coleção de propriedades *XML For Analysis* que permitem ao usuário, controlar alguns aspectos do método *Discover*, tais como: especificar o formato do retorno, e fornecer informações sobre a fonte de dados que será consultada. As propriedades disponíveis e seus valores podem ser obtidos utilizando o valor enumerado *DISCOVER_PROPERTIES* no parâmetro *RequestType* do método *Discover*;
- ***Result***: Este parâmetro contém o resultado retornado pelo servidor, de acordo com os valores especificados nos parâmetros listados anteriormente.

4.4.1.2 Método *Execute*

O método *Execute* é utilizado para executar Expressões Multidimensionais (MDX) [MDX03a], ou outros comandos específicos do provedor em uma determinada fonte de dados *XML For Analysis*.

Uma chamada ao método *Execute* pode ser feita de acordo com a seguinte sintaxe:

```
Execute (  
    [parâmetro de entrada] Command As Command,  
    [parâmetro de entrada] Properties As Properties,  
    [resultado da requisição] Result As Resultset )
```

Os parâmetros do comando *Execute* podem ser descritos da seguinte maneira:

- ***Command***: Este parâmetro consiste de uma expressão multidimensional (MDX) [MDX03a], ou outra expressão compatível com o servidor. Essa expressão é executada pelo servidor e o seu resultado é retornado no parâmetro *Result*. Outro padrão que provavelmente substituirá o MDX é o mdXML [MDXML03], que ainda encontra-se em fase de desenvolvimento;
- ***Properties***: Este parâmetro consiste de uma coleção de propriedades *XML for Analysis*. Estas propriedades permitem controlar alguns aspectos do método *Execute*, tal como definir a informação requerida para a conexão e o formato dos dados retornados;
- ***Result***: Contém o resultado da execução da expressão enviada no parâmetro *Command* formatado de acordo com as informações passadas pelo elemento *Properties*.

Informações detalhadas a respeito dos tipos dos parâmetros utilizados para realizar chamadas ao método *Discover*, tal como *Rowset*, *EnumString*, *Restrictions*, *Resultset*, *Command* e *Properties* podem ser encontradas no documento que especifica o padrão *XML for Analysis* [XMLA02].

Uma possível implementação de uma aplicação baseada na especificação *XML For Analysis* [XMLA02], pode ser feita conforme ilustrado na figura 4.8.

Em uma aplicação semelhante à representada na figura 4.8, um *Web Service* cliente, munido das informações sobre a localização de um serviço e operações que ele disponibiliza, o que é normalmente obtido através de um documento WSDL, faz uso dos protocolos HTTP e SOAP para enviar chamadas ao provedor. O provedor manipula as chamadas aos métodos *Discover* e *Execute* e devolve o resultado para o cliente. É importante lembrar que tanto as requisições do cliente quanto as respostas do servidor trafegam em formato XML, mais especificamente de acordo com o protocolo SOAP.

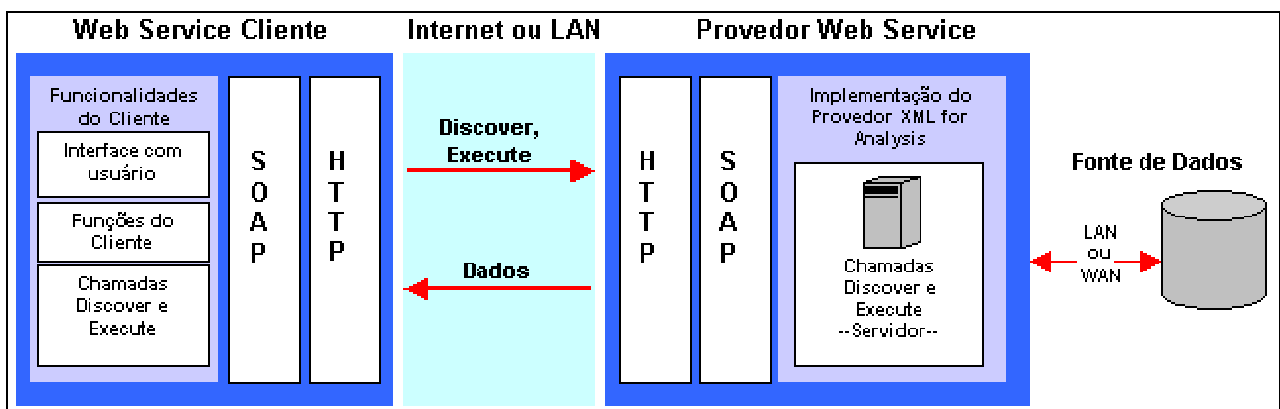


Figura 4.8: Arquitetura de uma possível aplicação baseada no padrão *XML for Analysis* (adaptado de [XMLA02]).

4.5 WFS - *Web Feature Service*

Com intuito de possibilitar uma maior interoperabilidade no acesso e na manipulação de dados geográficos através da *Web*, o *Open GIS Consortium* (OGC) [OGC03] desenvolveu dois documentos: (1) *OpenGIS® Web Map Service Implementation Specification* [WMS03] e (2) *OpenGIS® Geography Markup Language (GML) 2.0 Implementation Specification* [GML03a]. O primeiro documento especifica interfaces *Web* baseadas em um modelo que suporta requisições e respostas utilizando HTTP [HTTP03] e XML [XML03a] para acessar informações geográficas armazenadas em um servidor. O segundo documento descreve a especificação para codificação de dados geográficos em XML, permitindo tanto o armazenamento quanto o transporte de informações espaciais em formato XML [WFS02].

Seguindo a mesma linha de raciocínio, uma terceira especificação foi desenvolvida pelo OGC. Trata-se da *OpenGIS® Web Feature Service Implementation Specification* [WFS02], que propõe interfaces para descrição de operações que manipulam feições geográficas em ambiente distribuído por meio do protocolo HTTP. Entende-se por feições geográficas ou *features*, objetos geográficos que podem conter no mínimo uma propriedade geométrica e uma ou mais propriedades descritivas. No formato vetorial, as formas geométricas mais utilizadas são representadas por pontos, linhas e polígonos. Operações para manipulação de dados incluem habilidades para criar, apagar, alterar, obter ou pesquisar feições baseadas em restrições espaciais (*e.g.* adjacência e distância) e não espaciais (*e.g.* faixa etária = 1 e sexo = M).

4.5.1 Arquitetura do *Web Feature Service*

Com a implementação de um *Web Feature Service*, possibilita-se que aplicações cliente, através de requisições HTTP, obtenham de um WFS, dados espaciais codificados em GML [GML03a]. Uma requisição WFS consiste da descrição de uma operação de consulta ou alteração de dados que serão aplicadas a uma ou mais feições. Assim que o WFS recebe a requisição ele a processa e envia a resposta ao cliente. Se algum erro impossibilita que a requisição seja executada corretamente, o WFS retorna informações sobre o erro ocorrido. A implementação de um WFS, pode ser representada conforme podemos visualizar na figura 4.9.

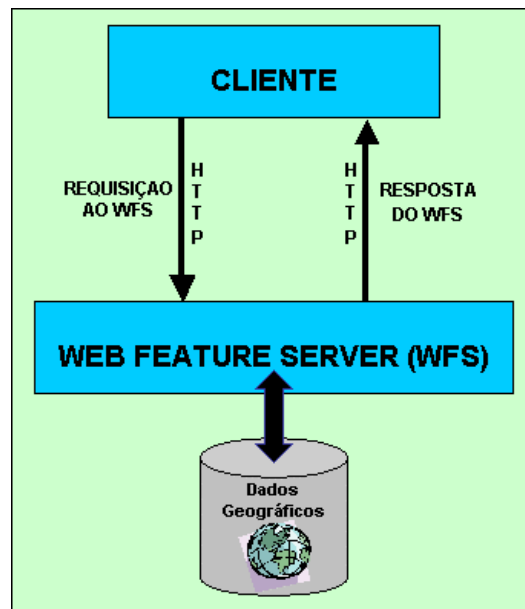


Figura 4.9: Implementação de um *Web Feature Service* (Adaptado de [WFS02]).

Os passos que normalmente são envolvidos no processamento de uma requisição WFS podem ser descritos da seguinte maneira:

1. A aplicação cliente deve requisitar o documento de habilidades que contém a descrição de todas as operações que o WFS suporta e a lista de todos os tipos de feições geográficas que ele pode servir. Este documento de habilidades pode ser obtido através da operação *GetCapabilities*;
2. A aplicação cliente (opcionalmente) faz uma requisição para o WFS para obter a definição de um ou mais tipos de feições que o WFS pode servir, através da operação *DescribeFeatureType*;
3. Baseado nessa definição do tipo de feição, a aplicação cliente gera uma requisição para ser enviada ao servidor;
4. A requisição é postada para um Servidor *Web*;
5. O WFS é invocado para ler e servir a requisição;
6. Quando o WFS completa o processamento da requisição, um relatório de status é gerado e enviado para o cliente. Caso ocorra algum tipo de erro, este é relatado no documento de resposta.

4.5.2 Operações do *Web Feature Service*

O *Web Feature Service* normalmente disponibiliza operações para criar, apagar, alterar, obter ou pesquisar feições geográficas. Nas subseções seguintes, serão descritas estas operações.

4.5.2.1 *GetCapabilities*

Um WFS precisa estar apto a descrever suas capacidades, ou seja, ele deve informar quais tipos de feições ele pode servir, e quais operações podem ser realizadas sobre esses tipos. As funcionalidades de um WFS podem ser obtidas através da operação *GetCapabilities*, a qual

retorna um documento que normalmente possui quatro principais seções: (1) ***Service section***: contém informações sobre o serviço em si; (2) ***Capabilities section***: onde é especificada a lista de requisições que o WFS pode manipular; (3) ***FeatureType list***: nesta seção, encontra-se a lista de tipos de feições disponíveis no WFS e as operações que podem ser aplicadas sobre cada tipo; (4) ***Filter Capabilities section***: esta é uma seção opcional, se ela existir conterá informações sobre as opções de filtragem oferecidas pelo WFS.

Expressões de Filtro [FEIS03] podem ser utilizadas para selecionar um conjunto específico de feições às quais serão aplicadas às operações. Na seção 4.5.4, são descritos com mais detalhes, os tipos de operadores que podem ser utilizados para requisitar um determinado conjunto de feições geográficas providas por um WFS.

4.5.2.2 DescribeFeatureType

Esta operação habilita o WFS a gerar uma descrição dos tipos de feições que ele pode servir. Em uma requisição pode-se especificar qual feição ou conjunto de feições serão descritas. Caso não tenha sido especificado o nome de uma feição para ser descrita, o documento de resposta conterá a descrição de todas as feições que o WFS pode servir.

4.5.2.3 GetFeature

A operação *GetFeature* permite que sejam recuperadas feições de um WFS. Quando uma requisição *GetFeature* é executada, um documento XML contendo o resultado da execução é retornado ao cliente.

4.5.2.4 Transaction

Uma requisição do tipo *Transaction* pode ser utilizada para criar, alterar ou excluir uma feição no WFS. Ao final da execução da operação, o WFS gera um documento contendo a situação final da transação.

4.5.2.5 LockFeature

Uma operação do tipo *LockFeature* pode ser aplicada a uma ou mais feições durante uma transação que esteja alterando uma ou mais feições no WFS. Com essa operação se torna possível requisitar uma feição para alteração de forma exclusiva, impedindo assim que outro cliente requisite a mesma feição.

Na figura 4.10, pode-se visualizar um diagrama que representa as mensagens que podem ser trocadas entre o WFS e uma aplicação cliente para o processamento de uma requisição. Pode-se observar que para cada tipo de requisição feita pela aplicação cliente, o servidor envia um documento contendo o resultado da execução da requisição.

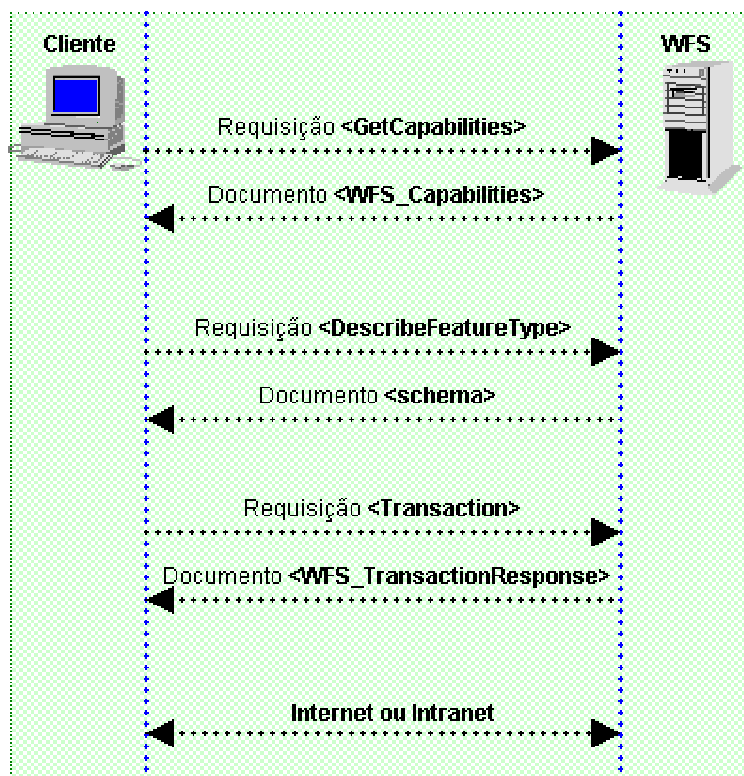


Figura 4.10: Possíveis mensagens trocadas entre uma aplicação cliente e o WFS (Adaptado de [WFS02]).

4.5.3 Classes de WFS

Com base na descrição das operações anteriormente citadas, podem ser definidas duas classes de *Web Feature Services*:

Basic WFS: Essa classe implementa as operações *GetCapabilities*, *DescribeFeatureType* e *GetFeature* e é considerada WFS de somente leitura.

Transaction WFS: Essa classe normalmente suporta todas as operações do *Basic WFS* e, adicionalmente, implementa a operação *Transaction* e opcionalmente pode implementar também a operação *LockFeature*.

Em nosso trabalho, somente será considerada a classe básica, pois serão necessárias somente consultas de leitura em um WFS, a fim de obter informações sobre o serviço, operações disponíveis e sobre as feições que ele pode servir.

4.5.4 Expressões de Filtro

Quando é enviada uma requisição *GetFeature* para um *Web Feature Service*, com intuito de recuperar alguma feição geográfica, é possível estabelecer algumas restrições levando em conta tanto propriedades espaciais como não espaciais. Isso é possível graças à codificação de filtro definida em *OpenGIS® Filter Encoding Implementation Specification* [FEIS03], a qual descreve uma codificação XML da *OGC Common Catalog Query Language (CQL)* [CSS03].

Uma expressão de filtro XML pode ser transformada em uma cláusula *WHERE* de uma instrução *SELECT* do *SQL* [Dat89, Mel93, SQL92] para recuperar dados armazenados em bancos de dados relacionais. Da mesma forma, uma expressão de filtro XML pode ser transformada em uma expressão *Xpath* [XPA03] ou *Xpointer* [XPO03] para recuperar dados armazenados em documentos XML. Em uma expressão de filtro, o elemento ***PropertyName*** é utilizado para codificar o nome de qualquer propriedade de um objeto.

O elemento raiz de uma expressão de filtro é o *<Filter>*. Ele contém a expressão de filtro que é formada através da combinação dos elementos que representam os operadores

espaciais, lógicos e de comparação. Na figura 4.11, é apresentado o fragmento do XML Schema que define o elemento `<Filter>`.

```
...
<xsd:element name="Filter" type="ogc:FilterType"/>
  <xsd:complexType name="FilterType">
    <xsd:choice>
      <xsd:element ref="ogc:spatialOps"/>
      <xsd:element ref="ogc:comparisonOps"/>
      <xsd:element ref="ogc:logicOps"/>
      <xsd:element ref="ogc:FeatureId" maxOccurs="unbounded"/>
    </xsd:choice>
  </xsd:complexType>
...
```

Figura 4.11: Fragmento do XML Schema que define o elemento `<Filter>`.

Na figura 4.11, os elementos `<logicalOps>`, `<spatialOps>` e `<comparasionOps>` são grupos de substituição [SCHE03] para operadores lógicos, espaciais e de comparação, respectivamente. O elemento `<FeatureID>` possibilita que um filtro codifique uma referência para uma ou mais instâncias de feições. Essa estrutura pode ser melhor compreendida através da figura 4.12.

4.5.4.1 Tipos de Operadores

Na especificação *Filter Encoding* [FEIS03], são descritos alguns operadores que podem ser utilizados em expressões de filtro. Estes operadores podem ser: (1) Operadores Espaciais, (2) Operadores Lógicos e (3) Operadores de Comparação. No diagrama da figura 4.12, gerado pela ferramenta *XML Spy* [XSPY03], pode-se visualizar graficamente o fragmento do XML Schema que define o elemento `<Filter>`. A seguir encontra-se a descrição de cada uma destas categorias de operadores.

Operadores Espaciais: Os operadores espaciais determinam se argumentos geométricos satisfazem um determinado relacionamento espacial. Os operadores que compõem este grupo são: *BBox*, *Beyond*, *Contains*, *Crosses*, *DWithin*, *Disjoint*, *Equals*, *Intersects*, *Overlaps*, *Touches* e *Within*.

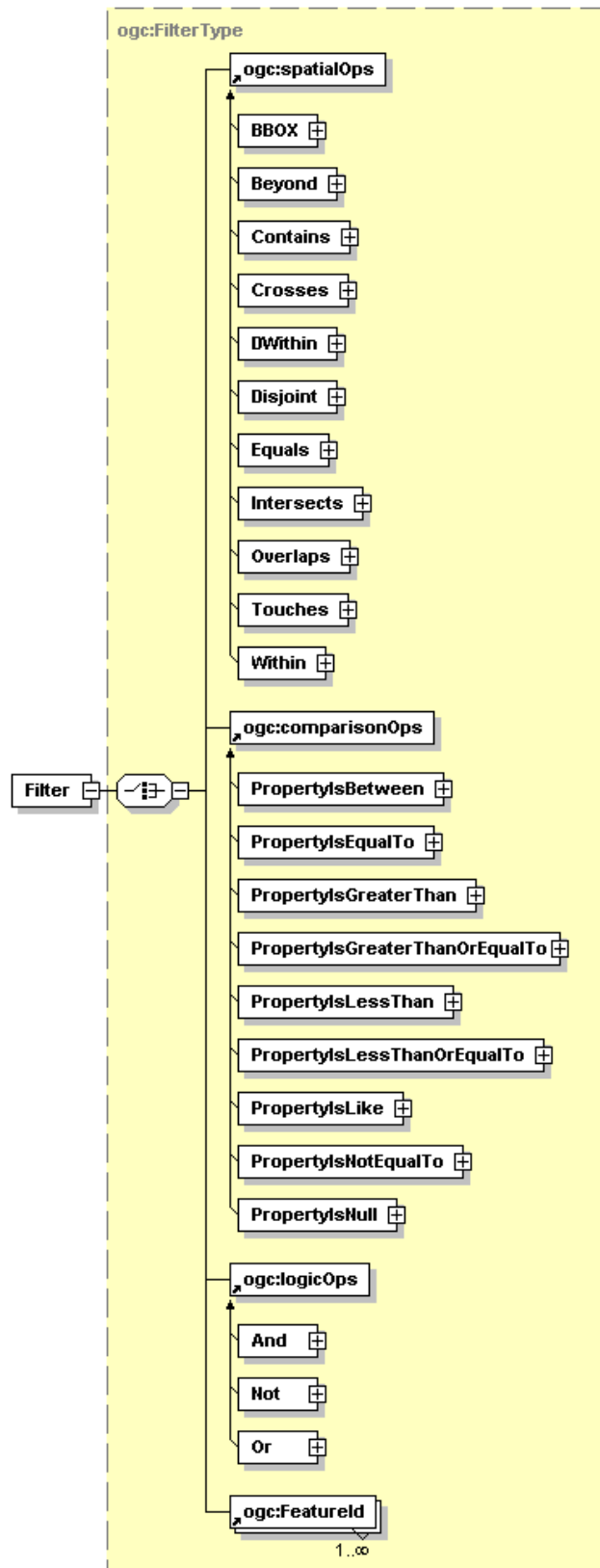


Figura 4.12: Estrutura do elemento `<Filter>`.

Cada um desses operadores possui uma finalidade. Por exemplo, o operador **Overlaps** avalia se uma propriedade geométrica sobrepõe outra. **Bounding Box (BBox)** é utilizado para identificar todas as geometrias que intersectam de alguma forma o retângulo representado por dois pares de coordenadas (Mínimo X, Mínimo Y, Máximo X e Máximo Y). Os operadores **DWithin** e **Beyond** verificam se o valor de uma propriedade geométrica está dentro ou além de uma determinada distância, respectivamente. A semântica desses operadores pode ser conferida em [CSS03]. A semântica dos demais operadores pode ser consultada em [SFS99].

Operadores de Comparação: Em uma requisição enviada para um WFS também podem existir operadores de comparação. Eles são utilizados para efetuar uma comparação matemática entre dois argumentos. Em [CSS03] é definido um conjunto padrão de operadores de comparação (i.e. =, <, >, >=, <=, <>). Além disso, são definidos os elementos: (1) **PropertyIsLike** que é utilizado para comparação de conjuntos de caracteres; (2) **PropertyIsNull** que serve para verificar se um valor é nulo; e (3) **PropertyIsBetween**, usado para checar determinada variação entre dois valores, onde tanto o menor como o maior valor são incluídos.

Operadores Lógicos: Outro grupo de operadores descritos na especificação [FEIS03] é o grupo de operadores lógicos. Os elementos **And**, **Or** e **Not** podem ser utilizados para combinar uma ou mais expressões condicionais, expressões escalares, espaciais ou outras operações lógicas, formando expressões mais complexas.

A especificação *Filter Encoding* [FEIS03] também define os operadores aritméticos, os quais são utilizados para codificar as operações aritméticas fundamentais como adição, subtração, divisão e multiplicação. Estes operadores são binários, aceitam dois argumentos e retornam um único resultado.

Segundo [FEIS03], as capacidades de filtro podem ser divididas em: (1) Capacidades Espaciais e (2) Capacidades Escalares. Capacidades Espaciais incluem a habilidade para filtrar dados espaciais baseados na definição do operador **BBox**, bem como processar os operadores espaciais **Beyond**, **Contains**, **Crosses**, **DWithin**, **Disjoint**, **Equals**, **Intersects**, **Overlaps**, **Touches** e **Within**. Por outro lado, as capacidades escalares se referem à habilidade de processar expressões lógicas, comparações, funções e operações aritméticas.

Na figura 4.13 podemos visualizar um exemplo de requisição *GetFeature*. A parte do código que está em negrito representa uma operação de filtro espacial *BBox*. Neste exemplo, serão listadas todas as cidades que estão localizadas na área definida pelo operador *BBox*.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<wfs:GetFeature outputFormat="GML2"
xmlns:gml="http://www.opengis.net/gml"
xmlns:wfs="http://www.opengis.net/wfs"
xmlns:ogc="http://www.opengis.net/ogc">
  <wfs:Query typeName="Cities">
    <ogc:Filter>
      <ogc:BBOX>
        <ogc:PropertyName>/Cities/location</ogc:PropertyName>
        <gml:Box>
          <gml:coord>
            <gml:X>-8</gml:X>
            <gml:Y>40</gml:Y>
          </gml:coord>
          <gml:coord>
            <gml:X>12</gml:X>
            <gml:Y>56</gml:Y>
          </gml:coord>
        </gml:Box>
      </ogc:BBOX>
    </ogc:Filter>
  </wfs:Query>
</wfs:GetFeature>
```

Figura 4.13: Requisição *GetFeature* com filtro espacial *BBox*.

Outro exemplo de operadores de filtro pode ser visualizado na figura 4.14. Neste exemplo, são selecionados todos os países da Europa que possuem uma área menor que 50000 Km² e estão de acordo com a restrição estabelecida pelo operador *BBox*. Neste exemplo, a parte do código que está em negrito representa uma operação de comparação *PropertyIsLessThan*.

```

<?xml version="1.0" encoding="iso-8859-1"?>
<wfs:GetFeature outputFormat="GML2"
xmlns:gml="http://www.opengis.net/gml"
xmlns:wfs="http://www.opengis.net/wfs"
xmlns:ogc="http://www.opengis.net/ogc">
  <wfs:Query typeName="Europe">
    <wfs:PropertyName>/Europe/Name</wfs:PropertyName>
    <wfs:PropertyName>/Europe/CountryArea</wfs:PropertyName>
    <wfs:PropertyName>/Europe/Border</wfs:PropertyName>
    <ogc:Filter>
      <ogc:And>
        <ogc:BBOX>
          <PropertyName>/Europe/Border</PropertyName>
          <gml:Box>
            <gml:coord>
              <gml:X>1</gml:X>
              <gml:Y>40</gml:Y>
            </gml:coord>
            <gml:coord>
              <gml:X>12</gml:X>
              <gml:Y>56</gml:Y>
            </gml:coord>
          </gml:Box>
        </ogc:BBOX>
        <ogc:PropertyIsLessThan>
          <ogc:PropertyName>/Europe/CountryArea</ogc:PropertyName>
          <ogc:Literal>50000</ogc:Literal>
        </ogc:PropertyIsLessThan>
      </ogc:And>
    </ogc:Filter>
  </wfs:Query>
</wfs:GetFeature>

```

Figura 4.14: Requisição *GetFeature* combinando o operador espacial *BBox* e o operador de comparação *PropertyIsLessThan*.

4.6 Considerações Finais

Neste capítulo, foram apresentados alguns conceitos relacionados às principais tecnologias envolvidas na elaboração do presente trabalho, o qual trata da integração de serviços analíticos e geográficos para suporte à decisão na *Web*. Cada uma delas, exerce uma importante função na pesquisa descrita nesta dissertação. A linguagem XML é bastante utilizada para promover a interoperabilidade entre aplicações, facilidade no intercâmbio de dados e suporte à definição de novos padrões tecnológicos. E neste trabalho, XML é amplamente utilizada para manipulação, processamento e apresentação de dados analíticos e geográficos.

Um dos padrões relacionado com XML é o *XML Schema*, o qual permite a elaboração de esquemas que definem a estrutura de um documento XML, através dos quais se torna possível manipular documentos tendo a garantia de que são válidos e bem formados. Com *XML Schema*, além da estruturação dos elementos, pode-se criar novos tipos de dados que satisfaçam às necessidades de uma determinada aplicação. E *XML Schema* é utilizado aqui para a definição de novos tipos de dados e esquemas para validação de documentos XML, que serão utilizados para manipulação e processamento de dados analíticos e geográficos.

Um padrão de grande importância apresentado neste capítulo, e que está extremamente relacionado com a natureza dos dados geográficos, é o GML. Este padrão foi definido pelo *Open Gis* [OGC03] para permitir a codificação, armazenamento e intercâmbio de dados geográficos no formato XML. XSLT também foi apresentada neste capítulo e neste trabalho é utilizada para realizar transformações em dados codificados em XML. Uma de suas aplicações é a transformação dos dados geográficos, representados em GML, para o formato SVG, fazendo com que estes dados sejam apresentados graficamente ao usuário, na forma de mapas.

A tecnologia de *Web Services* é inserida no contexto deste trabalho para possibilitar a integração dos serviços de processamento analítico (*i.e. XML For Analysis*) e processamento geográfico (*i.e. Web Feature Service*). Os *Web Services* permitem o desenvolvimento de aplicações que substituem a tradicional arquitetura cliente/servidor, possibilitando independência de plataforma e de linguagem de programação, dando suporte ao desenvolvimento de aplicações que trocam informações através da Internet, utilizando padrões abertos como XML, SOAP, HTTP e JAVA, entre outros.

Os conceitos apresentados neste capítulo estão fortemente relacionados com as idéias e implementações que serão descritas nos próximos capítulos. Neles será demonstrado como estas tecnologias foram utilizadas no processo de integração de serviços, bem como será apresentada a descrição detalhada da aplicação de cada uma delas.

Capítulo 5

Modelo ISAG

Neste capítulo, será proposto o modelo ISAG para a integração de serviços analíticos e geográficos para suporte à decisão na *Web*. ISAG se aplica à arquitetura GOLAPA (apresentada no capítulo 3), estando voltado, mais especificamente, para seu componente GOLAPE. Salienta-se que ISAG faz parte desta arquitetura mas é independente da mesma, podendo ser utilizado em outras soluções que visem a integração de serviços analíticos e geográficos.

5.1 Introdução

No processo de engenharia de sistemas, modelos [BRJ99, EN99] são usados para descrever de forma simplificada e gráfica os principais conceitos, atividades, processos e/ou dados envolvidos na construção de um sistema. Com a utilização de modelos, aspectos pertinentes à implementação podem ser abstraídos, possibilitando um melhor entendimento dos principais processos que compõem um sistema.

Neste capítulo, são apresentados os resultados do estudo realizado sobre as especificações voltadas ao processamento multidimensional e geográfico, (*i.e.* XMLA e WFS, discutidas no capítulo 4). Para isso, é apresentado um mapeamento das correspondências existentes entre estas, considerando aspectos como operações de consulta a dados e metadados, forma de comunicação e formato das requisições e das respostas. Em seguida, é introduzido o *GMLA Request Schema*, que é responsável por integrar a sintaxe de consulta dos serviços que implementam as especificações XMLA e WFS. Também é apresentado neste capítulo, o modelo ISAG, o qual é voltado para a integração de serviços analíticos e geográficos na Web. ISAG é utilizado na implementação do componente GOLAPE da arquitetura GOLAPA, apresentada no capítulo 3. Este modelo conta com a existência de uma fonte de metadados, baseada no metamodelo GAM e no esquema GeoMD (ver seção 3.3.1.3). Tanto a especificação do modelo como sua implementação, seguem os passos definidos em [STB03a, STB03b], onde é proposta uma sistematização do ciclo de vida da integração de serviços na *Web*.

O restante deste capítulo está estruturado da seguinte forma: Na seção 5.2, é apresentado um mapeamento das correspondências existentes entre duas especificações que tratam do processamento de dados geográficos e multidimensionais no ambiente Web. A seção 5.3 mostra a especificação de um esquema XML que define os tipos de requisições suportadas pelo modelo de integração. Na seção 5.4, é apresentado o modelo de integração de serviços analíticos e geográficos proposto. Finalmente, na seção 5.5 são tecidas algumas considerações finais sobre este capítulo.

5.2 Estudo das Especificações Utilizadas

Para propor um modelo de integração de serviços analíticos e geográficos, optou-se por seguir especificações padronizadas, as quais são resultantes da união de esforços de grandes organizações, como é o caso do Open GIS [OGC03] e do *XML For Analysis* [XMLA03]. Para prover a padronização do processamento e intercâmbio de dados geográficos e multidimensionais, foram analisadas as especificações *XML For Analysis Specification* [XMLA02] e *Web Feature Service Implementation Specification* [WFS02].

Com a análise detalhada dessas duas especificações, foi realizado um mapeamento entre as correspondências existentes entre serviços que implementam estas especificações, no que se refere ao intercâmbio e processamento de dados. O estudo destes serviços se fez necessário para analisar cautelosamente a forma de funcionamento de cada um deles, bem como estudar as operações oferecidas e o formato dos parâmetros de entrada e saída. Com o estudo realizado, chegou-se ao mapeamento apresentado na figura 5.1, o qual apresenta as correspondências existentes entre o serviço analítico e o geográfico.

	WFS	XMLA
Consulta a Dados	<ul style="list-style-type: none"> • GetFeature • Transaction • LockFeature 	<ul style="list-style-type: none"> • Execute <ul style="list-style-type: none"> ○ MDX ○ mdXML
Consulta a Metadados	<ul style="list-style-type: none"> • GetCapabilities • DescribeFeatureType 	<ul style="list-style-type: none"> • Discover
Comunicação	<ul style="list-style-type: none"> • HTTP – GET, POST 	<ul style="list-style-type: none"> • SOAP/HTTP
Requisição	<ul style="list-style-type: none"> • XML – Expressões de Filtro 	<ul style="list-style-type: none"> • XML – Envelope SOAP
Retorno	<ul style="list-style-type: none"> • GML2 – XML • W3C XML Schema 	<ul style="list-style-type: none"> • SOAP – XML • W3C XML Schema

Figura 5.1: Correspondências entre os serviços WFS e XMLA.

As correspondências existentes entre o serviço analítico e o geográfico contemplam, basicamente, cinco aspectos: 1) **Consulta a Dados**, 2) **Consulta a Metadados**, 3) **Comunicação**, 4) **Requisição** e 5) **Retorno**, os quais podem ser descritos da seguinte forma:

1. **Consulta a Dados**: Como pode ser visualizado na figura 5.1, para realizar consulta a dados, o serviço WFS tem à disposição as operações *GetFeature*, *Transaction* e *LockFeature* (seção 4.5.2). Com estas operações o serviço geográfico WFS tem capacidade para pesquisar, alterar e excluir feições geográficas em uma fonte de dados geográficos. No contexto do presente trabalho, somente foi utilizada a operação *GetFeature*, pelo fato de não serem permitidas alterações ou exclusões no *Data Warehouse* Geográfico. Por sua vez, para consulta a dados, o serviço analítico XMLA dispõe do método *Execute* (seção 4.4.1.2) para realizar consultas em uma base de dados multidimensionais, através da utilização de expressões MDX [MDX03a] ou mdXML [MDXML03].
2. **Consulta a Metadados**: Quanto à consulta a metadados, o serviço geográfico WFS dispõe das operações *GetCapabilities* e *DescribeFeatureType* (ver seção 4.5.2). Com estas operações, pode-se obter informações referentes aos tipos de feições existentes em uma determinada fonte de dados geográficos, ou referentes a uma certa feição, bem saber como quais operações podem ser realizadas sobre estas feições. Por sua vez, para consultar metadados, o serviço analítico XMLA dispõe do método *Discover* (ver seção 4.4.1.1) para recuperar informações sobre as fontes de dados disponíveis, aspectos referentes aos cubos de dados multidimensionais, suas dimensões, hierarquias, entre outras informações.
3. **Comunicação**: No que se refere à comunicação entre uma aplicação cliente e o serviço geográfico WFS, esta é realizada através do protocolo HTTP, com a utilização de requisições GET e POST [HTTP03]. Já a comunicação entre uma aplicação cliente e o serviço analítico XMLA é realizada conforme o padrão dos *Web Services* (ver seção 4.3). Dessa forma, o envio e o recebimento de requisições é realizado através de envelopes SOAP [SOAP03].

4. **Requisição:** A correspondência existente entre os serviços WFS e XMLA, no que se relaciona à forma com que são realizadas as requisições, pode ser descrita como segue: As requisições contendo consultas geográficas e consultas a metadados, são enviadas ao serviço geográfico WFS através do protocolo HTTP codificadas em documentos XML que contêm expressões de filtro (ver seção 4.5.4). Com estas expressões de filtro, torna-se possível aplicar operações espaciais sobre os dados geográficos. De forma semelhante, também baseado em XML, as requisições ao serviço XMLA são realizadas através do envio de envelopes SOAP, padrão definido pelos *Web Services* para a troca de informações.

5. **Retorno:** Quanto ao tipo de retorno dos serviços em questão (*i.e.* XMLA e WFS), existe uma grande semelhança, pois como os dois são fortemente baseados em XML, a resposta destes serviços é sempre um documento XML. No serviço geográfico, quando executada uma consulta a dados, o retorno é dado em GML⁸. Na consulta a metadados, na maioria das vezes, a resposta é um *XML Schema* que descreve a feição geográfica ou conjunto de feições. Da mesma forma, no serviço analítico XMLA, o retorno para consulta a dados é um documento XML, que possui um *XML Schema* embutido para definir sua estrutura. Para consulta a metadados, a resposta também é baseada em XML e *XML Schema*.

Como pode ser analisado nos itens anteriores, que se encontram representados na figura 5.1, existem grandes semelhanças na forma em que os serviços (*i.e.* XMLA e WFS) manipulam os dados. Como estes dois serviços possuem seu processamento altamente baseado em XML, a utilização da tecnologia de *Web Services* pode ser aplicada com grandes vantagens no provimento de um processamento geográfico-multidimensional, proveniente da integração entre eles.

O mapeamento apresentado na figura 5.1, foi de grande utilidade para projetar o modelo de integração dos serviços (seção 5.4) e implementá-lo (seção 6.3), devido ao mesmo explicitar as correspondências existentes entre as operações, formato de dados e outros aspectos dos serviços XMLA e WFS.

⁸ Nas versões atuais do WFS a resposta é baseada na segunda versão da GML (*i.e.* GML2 [GML03a]).

5.3 GMLA Request Schema

Outro passo importante para o desenvolvimento do modelo de integração de serviços analíticos e geográficos, foi a definição do *GMLA Request Schema*. Trata-se de um esquema XML que define os tipos de requisições que podem ser enviadas ao serviço GMLA WS (*GMLA Web Service*) (ver seção 6.2), o qual implementa o modelo proposto neste capítulo.

Neste trabalho, a implementação do modelo de integração de serviços foi realizada utilizando *Web Services*, onde, o GMLA WS recebe um envelope SOAP contendo a requisição que ele irá processar. A aplicação cliente elabora esta requisição no formato de um documento XML. Para este documento ser empacotado em um envelope SOAP e posteriormente enviado ao GMLA WS, ele precisa ser válido e bem formado.

Para definir a estrutura do documento e realizar a validação de uma requisição GMLA, foi desenvolvido o *GMLA Request Schema*, o qual valida as consultas que serão enviadas ao GMLA WS, possibilitando, dessa forma, que o GMLA WS possa extrair os parâmetros da requisição para posterior processamento, sem que ocorram erros de natureza sintática. O conjunto de elementos definidos pelo *GMLA Request Schema* engloba todos os parâmetros necessários para a realização de uma consulta multidimensional, geográfica ou ainda geográfica-multidimensional.

Executando de forma isolada, o serviço analítico (*i.e.* XMLA - ver seção 4.4) e o geográfico (*i.e.* WFS - ver seção 4.5), possuem sintaxe de consulta diferente. As consultas ao XMLA são realizadas através do envio e recebimento de envelopes SOAP (*i.e.* padrão *Web Service*). Por sua vez, as consultas ao WFS são realizadas através do envio e recebimento de requisições HTTP (*i.e.* GET, POST), utilizando as expressões de filtro (seção 4.5.4) [FEIS03]. O ponto comum entre estes dois tipos de consultas é que elas são codificadas em documentos XML. Dessa forma, *GMLA Request Schema* herda características dessas duas tecnologias, preservando os elementos definidos para cada um dos casos, e apenas acrescentando novos elementos de acordo com a necessidade de representar novas informações.

No diagrama apresentado na figura 5.2, gerado pela ferramenta *XML Spy* [XSPY03], pode ser visualizada a estrutura do *GMLA Request Schema*. Este esquema define os tipos de requisições que podem ser enviadas ao GMLA WS, bem como a estrutura de cada uma delas. O

GMLA WS realiza a integração dos serviços analítico (*i.e.* XMLA) e geográfico (*i.e.* WFS). Dessa forma, foram definidos pelo *GMLA Request Schema* três tipos de requisições: MD, GEO e GEOMD.

Uma requisição MD é aquela que contém somente parâmetros analíticos, para execução de uma consulta multidimensional. Quando uma requisição desse tipo é recebida pelo GMLA WS, o serviço analítico XMLA será requisitado, produzindo assim o seu resultado. Por sua vez, uma consulta do tipo GEO é aquela que contém apenas parâmetros geográficos, onde o GMLA WS acionará o serviço WFS para que este processe a mesma. Por outro lado, uma requisição do tipo GEOMD é aquela contendo parâmetros tanto analíticos quanto geográficos, onde os dois serviços (*i.e.* XMLA e WFS) serão consultados a fim de produzirem um resultado que satisfaça à requisição. Uma requisição GEOMD pode ser classificada em: 1) GEOMD de Mapeamento, quando é realizada uma consulta analítica e se deseja visualizar os dados que possuem correspondência geográfica em um mapa, e 2) GEOMD de Integração, quando é enviada ao GMLA WS uma requisição contendo parâmetros analíticos, juntamente com parâmetros geográficos que contêm algum tipo de operador espacial, aritmético, lógico ou de comparação, que irá restringir a consulta. Maiores detalhes pertinentes à forma de processamento dessas requisições são dados na seção 6.8 deste documento.

O elemento principal do *GMLA Request Schema* é o *GmlaRequest* (ver figura 5.2). Este elemento aparece em todo documento que contém uma requisição GMLA. Como subelemento do *GmlaRequest*, existe um elemento *MD* (para o caso de uma consulta apenas multidimensional), ou um elemento *GEO* (em caso de um consulta apenas geográfica) ou ainda um elemento *GEOMD* (em caso de uma consulta de mapeamento ou integração).

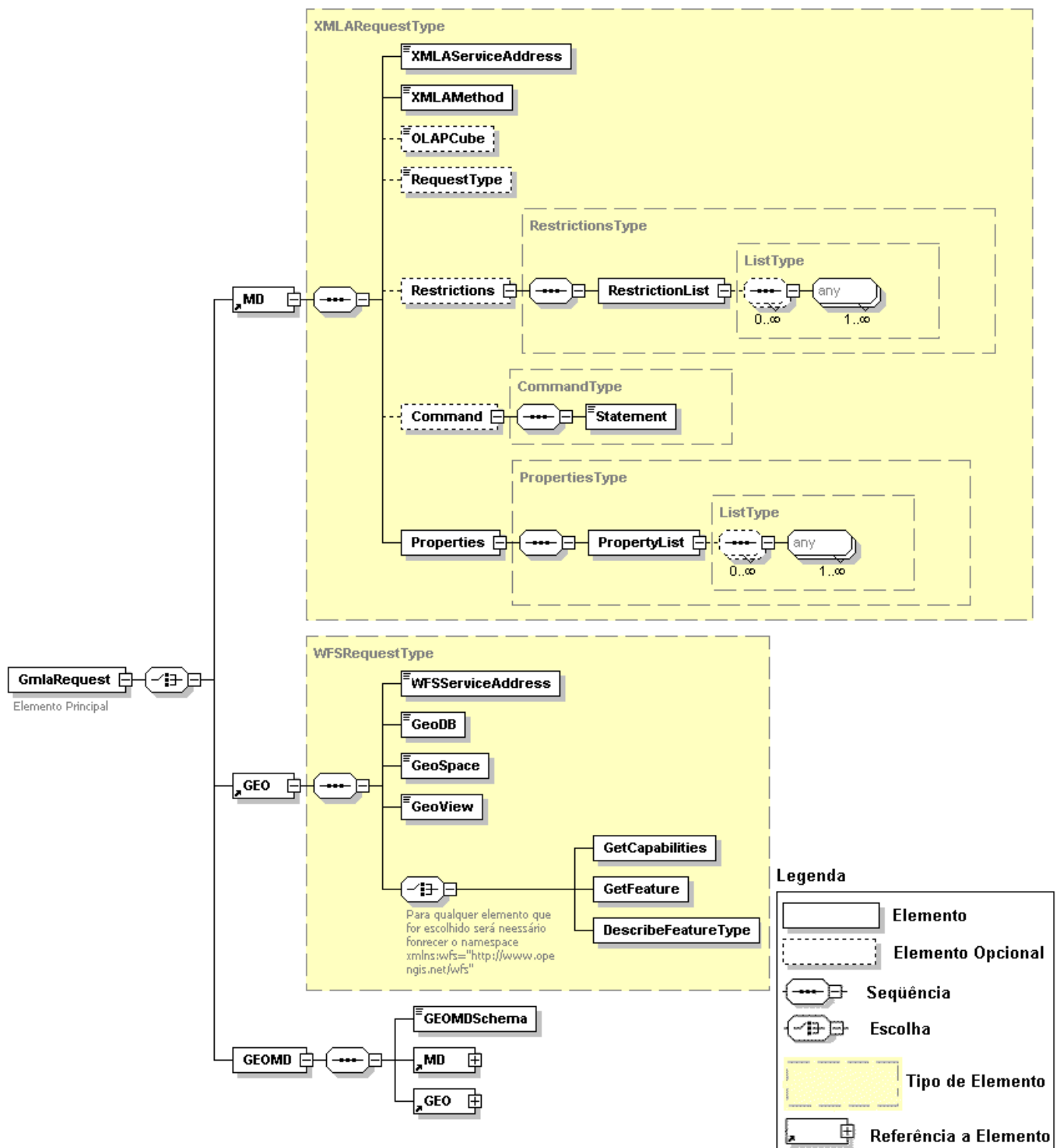


Figura 5.2: Estrutura do GMLA Request Schema.

Os elementos *MD* e *GEO* possuem subelementos que representam os parâmetros necessários para que o GMLA WS possa identificar o tipo da requisição e acionar o serviço

correspondente, produzindo assim, o resultado da consulta. Por sua vez, o elemento *GEOMD*, contém um elemento chamado *MD*, outro chamado *GEO* e também o elemento *GEOMDSchema* que contém o nome do esquema de integração utilizado na consulta. Este esquema corresponde a uma instância do metamodelo *GeoMD* (seção 3.3.1.3), o qual é responsável por representar as correspondências existentes entre os dados analíticos e geográficos armazenados no *Data Warehouse Geográfico* (e.g. correspondências entre um cubo de dados multidimensionais e um banco de dados geográficos ou entre uma dimensão de um cubo e uma área geográfica).

As tabelas 5.1, 5.2 e 5.3 demonstram a estrutura dos elementos *MD*, *GEO* e *GEOMD* respectivamente, apresentando seus tipos e subelementos. Nas tabelas 5.1 e 5.2, a coluna *Origem* indica se o elemento foi herdado da sintaxe de consulta de um dos serviços (i.e. XMLA ou WFS), ou se é um elemento novo, criado para representar novas informações pertinentes ao *GMLA Request Schema*. Com isso, fica preservada a sintaxe de consulta dos dois serviços, incluindo um fator adicional, para possibilitar a realização de consultas integradas (i.e. geográfica-multidimensional), envolvendo a sintaxe de consulta de ambos os serviços.

Tabela 5.1: Estrutura do Elemento *MD*.

Elemento MD		
Subelemento	Tipo	Origem
<i>XMLAServiceAddress</i>	<i>String</i>	Novo
<i>XMLAMethod</i>	<i>String</i>	Novo
<i>OLAPCube</i>	<i>String</i>	Novo
<i>RequestType</i>	<i>String</i>	XMLA
<i>Restrictions</i>	<i>RestrictionsType</i>	XMLA
<i>Command</i>	<i>CommandType</i>	XMLA
<i>Properties</i>	<i>PropertiesType</i>	XMLA

Uma breve descrição sobre cada um dos subelementos de *MD* é dada a seguir:

- *XMLAServiceAddress*: É um elemento obrigatório e não pode ser vazio. Representa o endereço do serviço analítico XMLA.
- *XMLAMethod*: Elemento obrigatório e também não pode ser vazio. Armazena o nome do método que será executado pelo serviço XMLA. *Execute* para obtenção de dados e *Discover* para metadados (ver seção 4.4.1).

- *OLAPCube*: Contém o nome do cubo de dados multidimensionais, sobre o qual será executada a consulta. É opcional, sendo utilizado somente nos casos em que o método a ser executado (*XMLAMethod*) for o *Execute*.
- *RequestType*: É opcional, sendo utilizado somente quando o método a ser executado (*XMLAMethod*) for o *Discover*. É utilizado para determinar a estrutura do conteúdo retornado (ver seção 4.4.1.1).
- *Restrictions*: Elemento opcional, sendo utilizado somente quando o método a ser executado é o *Discover*. Possui um subelemento chamado *RestrictionList*, do tipo *ListType*, que pode conter uma lista de elementos representando as restrições que serão aplicadas aos dados retornados (ver seção 4.4.1.1).
- *Command*: É também opcional, sendo utilizado somente nas ocasiões em que o método *Execute* é executado (ver seção 4.4.1.2). Possui um subelemento chamado *Statement*, do tipo *String* que irá conter o comando MDX [MDX03a] ou mdXML [MDXML03] que será executado para recuperar dados de um cubo multidimensional.
- *Properties*: É obrigatório, sendo utilizado tanto pelo método *Execute* quanto pelo *Discover*, para controlar aspectos relacionados à conexão e ao formato dos dados retornados. Possui um subelemento chamado *PropertyList*, do tipo *ListType*, o qual poderá conter uma lista de subelementos representando estas propriedades (ver seção 4.4.1).

Tabela 5.2: Estrutura do Elemento *GEO*.

Elemento <i>GEO</i>		
Subelemento	Tipo	Origem
<i>WFSServiceAddress</i>	<i>String</i>	Novo
<i>GeoDB</i>	<i>String</i>	Novo
<i>GeoSpace</i>	<i>String</i>	Novo
<i>GeoView</i>	<i>String</i>	Novo
<i>GetCapabilities</i> , <i>GetFeature</i> ou <i>DescribeFeatureType</i>	<i>Filter Encoding</i> (ver seção 4.5.4)	WFS

Os subelementos do elemento *GEO* são descritos como segue:

- *WFSServiceAddress*: É obrigatório e não pode ser vazio. Representa o endereço do serviço geográfico WFS.
- *GeoDB*: É obrigatório e não pode ser vazio. Contém o nome do banco de dados geográficos sobre o qual será realizada a consulta geográfica.
- *GeoSpace*: É obrigatório e não pode ser vazio. Ele identifica qual espaço geográfico do banco de dados (*GeoDB*) será consultado.
- *GeoView*: Elemento obrigatório, não podendo ser vazio. Representa a visão geográfica sobre a qual será executada a consulta.
- *GetCapabilities*, *GetFeature* ou *DescribeFeatureType*: Um elemento *GEO* deverá conter um destes três subelementos. O elemento *GetCapabilities* é utilizado para requisitar ao serviço WFS, quais feições ele pode servir e quais operações podem ser realizadas sobre estas feições. O elemento *GetFeature* é utilizado quando se deseja obter uma feição ou conjunto de feições do serviço WFS, através da execução de uma consulta contendo expressões de filtro (ver seção 4.5.4). Por sua vez, o elemento *DescribeFeatureType* é utilizado para requisitar informações que descrevem uma feição ou conjunto de feições servidas pelo serviço WFS.

Tabela 5.3: Estrutura do elemento *GEOMD*.

Elemento <i>GEOMD</i>	
Subelemento	Tipo
<i>GEOMDSchema</i>	<i>String</i>
<i>MD</i>	<i>XMLRequestType</i>
<i>GEO</i>	<i>WFSRequestType</i>

A seguir se encontra a descrição de cada um dos subelementos do elemento *GEOMD*.

- *GEOMDSchema*: É obrigatório e não pode ser vazio. Contém o nome do esquema de integração ao qual a consulta está relacionada. Um

GEOMDSchema representa o nome de uma instância do esquema *GeoMD* (ver seção 3.3.1.3).

- *MD*: Elemento obrigatório e não pode ser vazio. Sua estrutura está descrita na tabela 5.1.
- *GEO*: É obrigatório e não pode ser vazio. Sua estrutura está descrita na tabela 5.2.

5.4 O Modelo de Integração ISAG

Após ter definido os tipos de consultas que o modelo ISAG [STF+03a] comportaria, com a especificação do esquema *GMLA Request*, o próximo passo foi desenvolver o modelo de integração de serviços analíticos e geográficos. Para a especificação do ISAG, foi utilizada a sintaxe de diagramas de atividades da linguagem de modelagem UML [BRJ99], devido ao fato da UML ser um padrão bastante conhecido e amplamente utilizado na especificação de modelos.

O modelo ISAG considera a necessidade de integrar serviços analíticos e geográficos, de forma que seja implementado um mecanismo de integração que possibilite a junção do processamento destes dois serviços. Dessa forma, o mecanismo de integração de serviços desenvolvido com base no modelo apresentado a seguir, terá capacidades de processamento tanto de dados analíticos como de dados geográficos, promovendo uma abstração da complexidade de consultar os dois serviços.

Outro fato que o modelo considera é a existência de uma base de metadados, a qual contém informações sobre quais dados analíticos possuem correspondências geográficas, bem como informações sobre como recuperar essas correspondências. Essas informações são muito úteis no momento da realização de consultas que requerem processamento para integrar ou mapear resultados.

Na figura 5.3, é apresentado o nível mais alto de abstração do modelo de integração, o qual é composto por um conjunto de atividades, as quais são desempenhadas de forma seqüencial, onde o resultado da execução de cada atividade é utilizado pela atividade seguinte, até que o processamento seja completado e o estado final seja alcançado. Cada atividade do

modelo ISAG possui subatividades, as quais precisam ser desempenhadas para que a atividade principal seja concluída. Dessa forma, cada atividade apresentada na figura 5.3 possui uma expansão, as quais serão descritas nesta seção.

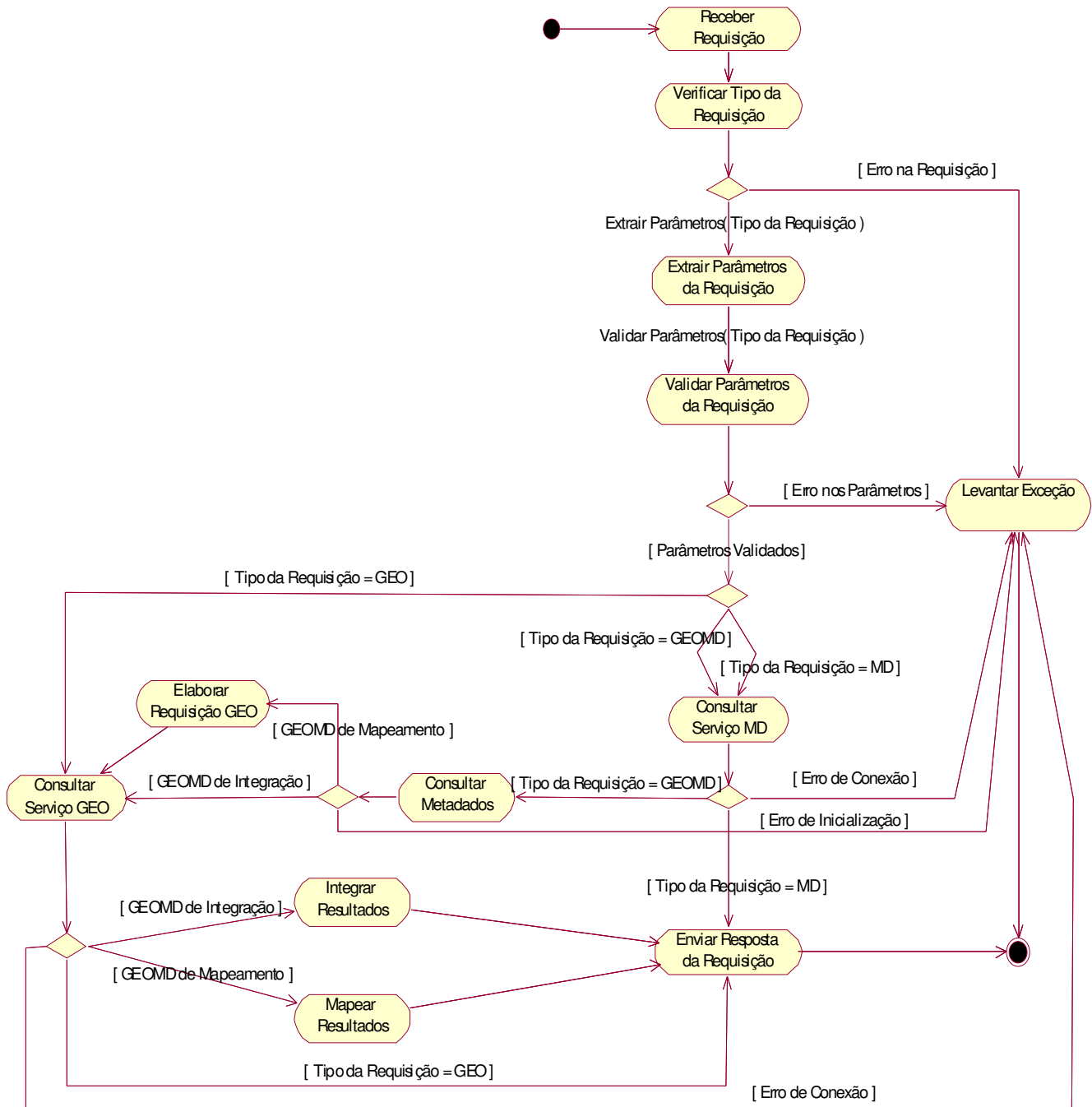


Figura 5.3: Modelo de Integração de Serviços Analíticos e Geográficos (ISAG).

As expressões entre colchetes no modelo (“[” “]”), indicam que a transição entre atividades só será realizada se determinada condição for satisfeita (i.e. “[Tipo da Requisição = MD]” indica que somente ocorrerá um evento que provocará a transição para a atividade **Consultar Serviço MD** se a requisição for do tipo MD). Por sua vez, a utilização de parênteses na transição entre atividades (e.g. *Extrair Parâmetros (Tipo da Requisição)* e *Validar Parâmetros (Tipo da Requisição)*), indicam que a atividade para a qual está ocorrendo a transição, receberá como parâmetro o argumento que está entre parênteses. No caso do exemplo citado o parâmetro será *Tipo da Requisição*.

Como pode ser visualizado na figura 5.3, a primeira atividade do modelo é a **Receber Requisição**. As atividades do modelo sempre consideram que a requisição recebida está de acordo com o esquema *GMLA Request*, dessa forma, as atividades são executadas considerando que a requisição sendo processada é válida e bem formada.

A próxima atividade representada no modelo é a **Verificar Tipo da Requisição**. Nesta atividade, é realizado um processamento que identifica qual o tipo de requisição que foi recebida. O tipo da requisição pode ser classificado em MD, GEO ou GEOMD, que são os tipos definidos pelo esquema *GMLA Request* apresentado na seção 5.3. O tipo da requisição é então armazenado para servir como parâmetro para as atividades seguintes. No caso da ocorrência de algum tipo de erro durante a execução dessa atividade, um evento é gerado e acontece a transição para a atividade **Levantar Exceção**, que é encarregada de informar o erro ocorrido para a aplicação cliente que enviou a requisição, e em seguida o processamento é encerrado. Caso nenhum erro ocorra, é realizada a transição para a atividade **Extrair Parâmetros da Requisição**, através da ocorrência do evento *Extrair Parâmetros*, que tem como parâmetro o tipo da requisição, identificado na atividade anterior.

A atividade **Extrair Parâmetros da Requisição** recebe como parâmetro o tipo da requisição, e de acordo com esse tipo, suas subatividades são encarregadas de extrair os parâmetros passados na requisição. Esses parâmetros podem ser entendidos como o conteúdo dos elementos MD, GEO e GEOMD do esquema *GMLA Request* (ver seção 5.3). Por exemplo, no caso de uma requisição MD, os parâmetros recuperados podem incluir o endereço do serviço analítico, o nome do método que será executado, o nome do cubo de dados que será consultado, as restrições da consulta, entre outros. Por sua vez, em caso de uma requisição GEO, os parâmetros a serem recuperados serão o endereço do serviço geográfico, o nome do banco de

dados geográficos, do espaço geográfico e da visão geográfica onde a consulta será realizada, e também a operação que foi submetida.

O diagrama UML que representa a expansão da atividade *Extrair Parâmetros da Requisição* é apresentado na figura 5.4. A primeira subatividade é *Extrair Nome do Esquema de Integração*. No modelo ISAG, o esquema de integração corresponde ao esquema GeoMD descrito na seção 3.3.1.3. Os símbolos em forma de barras horizontais representam o início e final de um processamento paralelo. No caso do modelo ISAG, isso não indica que dois processos serão executados ao mesmo tempo, mas sim, indica que a atividade *Extrair Parâmetros da Requisição* só será encerrada após a realização das atividades *Extrair Parâmetros MD* e *Extrair Parâmetros GEO*.

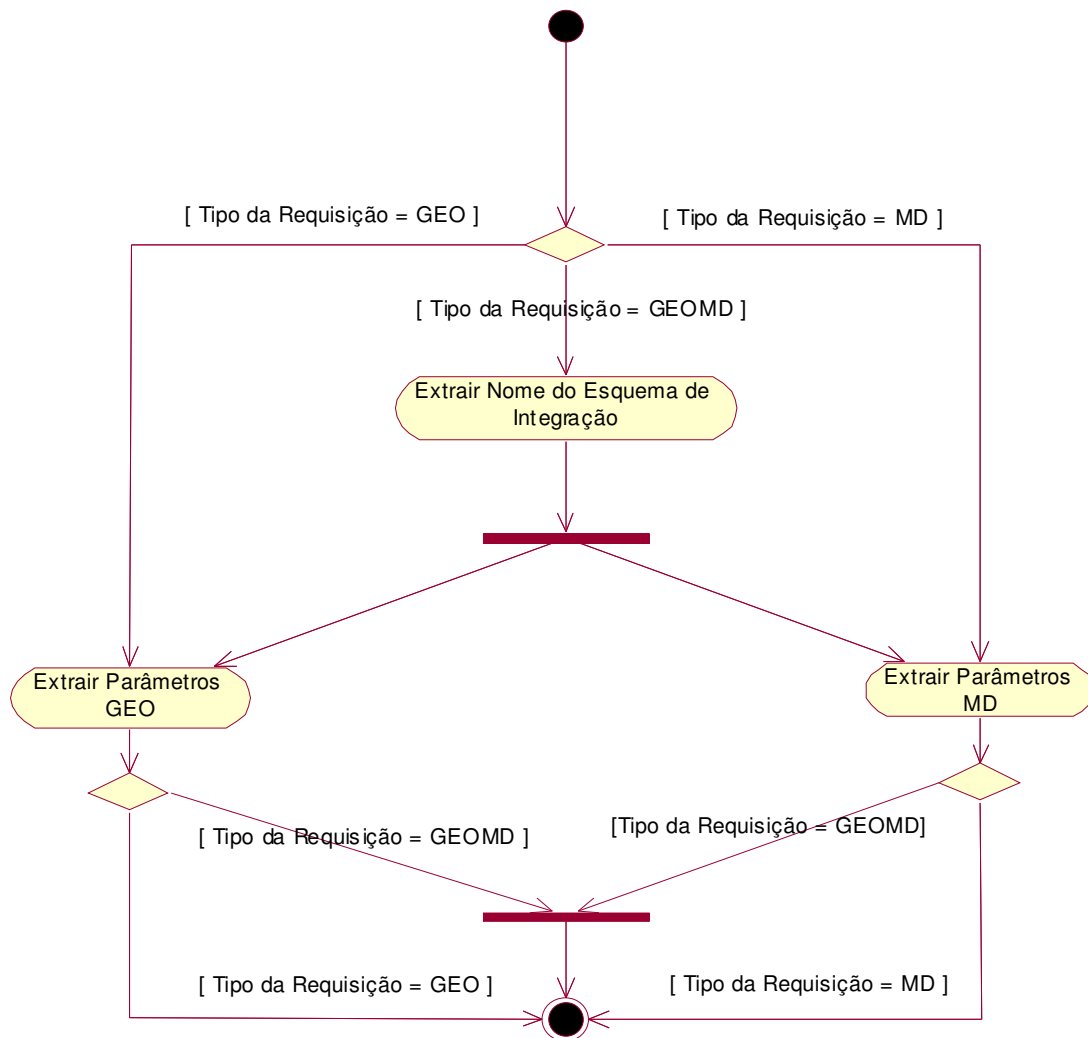


Figura 5.4: Expansão da atividade *Extrair Parâmetros da Requisição*.

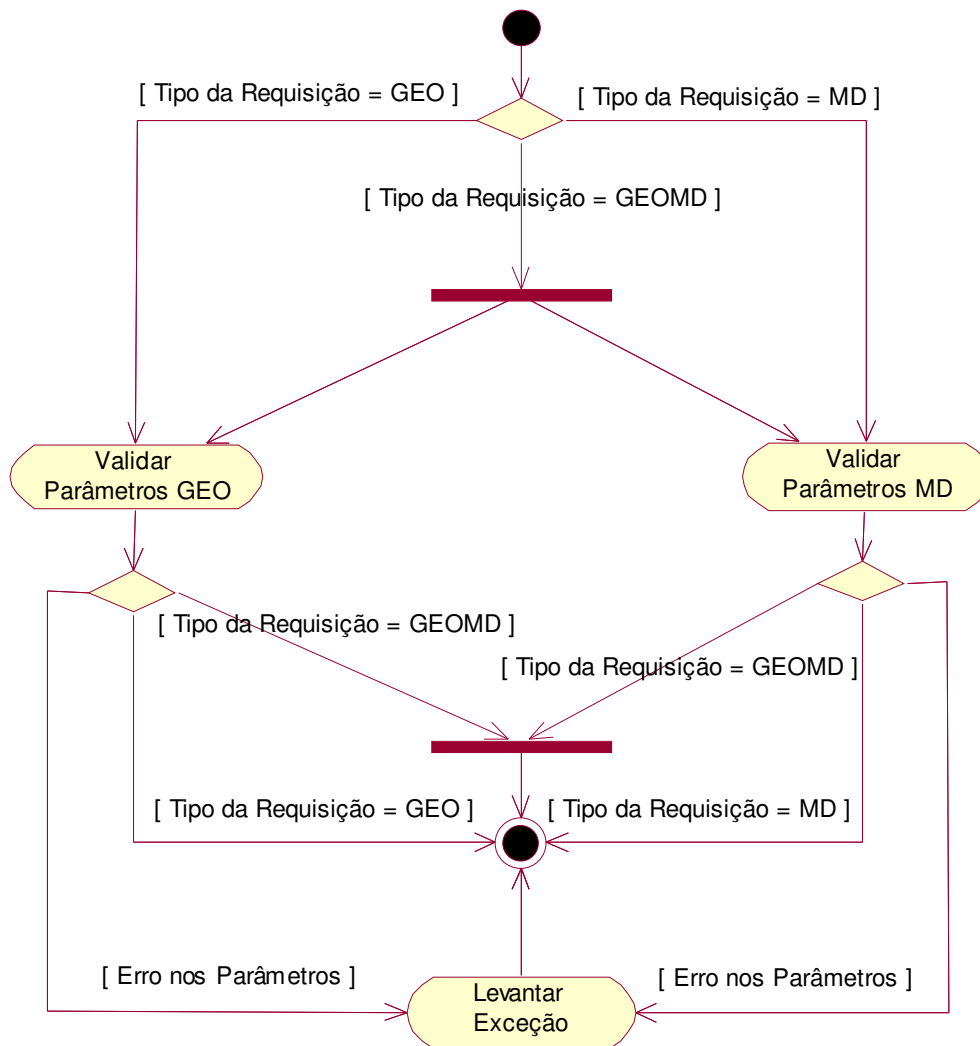


Figura 5.5: Expansão da atividade *Validar Parâmetros da Requisição*.

Outra atividade importante do modelo é a *Validar Parâmetros da Requisição* (ver figura 5.5). Esta atividade representa o processamento para validar os parâmetros que foram passados na requisição, os quais foram identificados na atividade anterior. Em caso de uma requisição MD por exemplo, um dos testes realizados é para verificar se o nome do cubo de dados contido no elemento *OLAPCube* é o mesmo que está sendo referenciado na instrução de consulta do elemento *Command*. Além disso, em uma requisição MD, outro processamento realizado é para identificar, de acordo com o nome do método passado em *XMLMethod*, se os demais parâmetros estão de acordo com a sintaxe de consulta do serviço analítico. É nesta atividade que é verificado se uma requisição do tipo GEOMD é de mapeamento ou não. Uma requisição é GEOMD de mapeamento se os parâmetros MD e GEO forem válidos e não for encontrada nenhuma operação espacial, lógica ou de comparação no elemento GEO. Isso indica

que devem ser recuperadas automaticamente as correspondências geográficas dos membros resultantes da consulta analítica. Em caso de erro em algum dos parâmetros ocorre um evento que identifica a transição para a atividade **Levantar Exceção**, e o processamento é encerrado.

Com a execução das atividades listadas acima, foi possível determinar o tipo da requisição que será processada e todos os parâmetros necessários para o processamento foram recuperados e validados. Os tipos de requisições foram definidos pelo esquema *GMLA Request* e caso: (i) o tipo da requisição seja MD ou GEOMD, a próxima atividade a ser executada é **Consultar Serviço MD**, (ii) o tipo for GEO, a próxima atividade será **Consultar Serviço GEO**.

A atividade **Consultar Serviço MD** é responsável por recuperar os parâmetros necessários para processar a requisição MD, estabelecer conexão com o serviço analítico e enviá-lo a consulta. Caso ocorra algum erro na conexão o processamento é cancelado. O diagrama representando a expansão da atividade **Consultar Serviço MD** pode ser visualizado na figura 5.6.

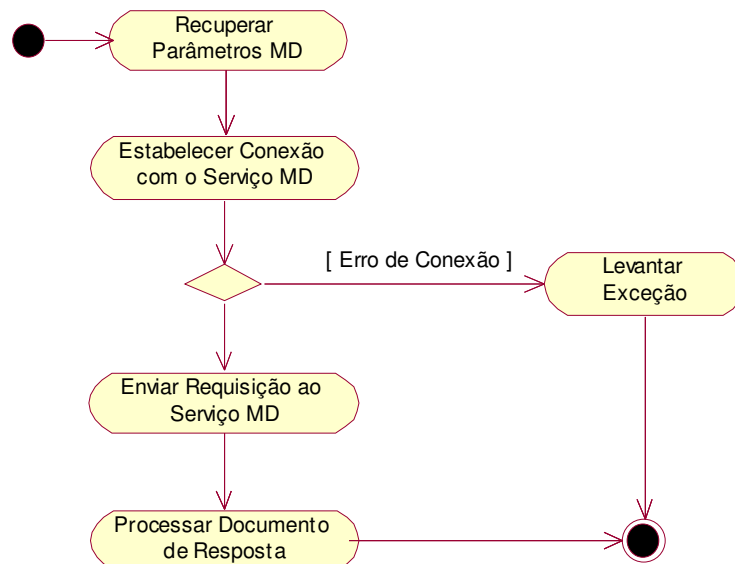


Figura 5.6: Expansão da atividade **Consultar Serviço MD**.

Por outro lado, se o tipo da requisição for GEO, a próxima atividade a ser executada é a **Consultar Serviço GEO** (ver figura 5.7). O processamento executado durante esta atividade envolve a recuperação dos parâmetros geográficos que fazem parte da requisição, o estabelecimento da conexão com o serviço geográfico e o envio da consulta para o mesmo. Nesta

atividade, podem acontecer três eventos, os quais possibilitam a transição para as atividades posteriores. Se o tipo da requisição for GEO, um evento ocasionará a transição para a atividade **Enviar Resposta da Requisição**. Se o tipo for GEOMD de mapeamento, a próxima atividade a ser executada será **Mapear Resultados**, e por último, se o tipo da requisição for GEOMD de integração, a atividade posterior será **Integrar Resultados**.

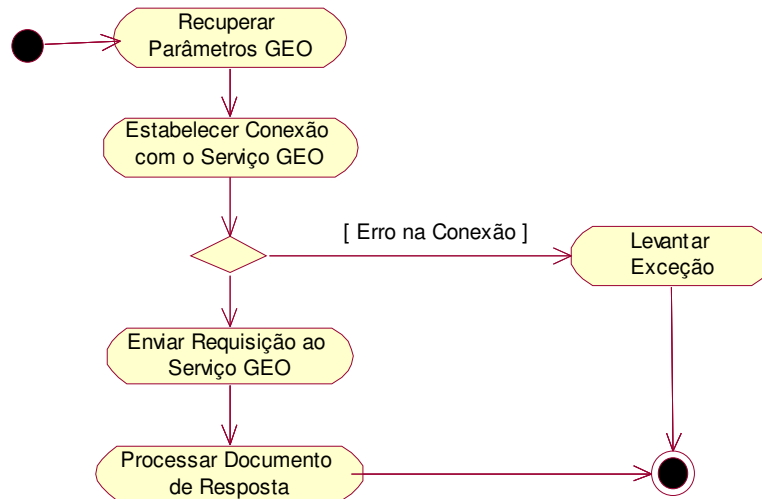


Figura 5.7: Expansão da atividade **Consultar Serviço GEO**.

O fato de consultar primeiro o serviço analítico em uma requisição GEOMD se deve a uma decisão de projeto. Um dos motivos para essa decisão é que foi definido que o modelo permitiria a realização de consultas GEOMD de mapeamento, onde os dados resultantes de uma consulta analítica, que possuem correspondência geográfica, poderiam ser visualizados graficamente em um mapa. Dessa forma, constatou-se que a melhor estratégia a adotar era consultar o serviço analítico e em seguida, através de uma consulta à base de metadados da integração, formular uma consulta que recuperasse as correspondências geográficas dos dados em questão. Outro motivo para a consultar primeiramente o serviço analítico é que o custo de processamento para a recuperação de dados geográficos é provavelmente maior. Deste modo, em uma requisição do tipo GEOMD de integração, a restrição sempre será realizada pelo lado geográfico, consultando primeiro o serviço analítico e depois o geográfico, e através de uma operação de produto cartesiano, são mantidos somente os dados analíticos que satisfizerem a restrição imposta na requisição geográfica.

Na seqüência das atividades que compõem o modelo ISAG, em caso de uma consulta do tipo GEOMD, após consultar o serviço analítico, é realizada uma consulta à base de metadados de integração, o que é representado no modelo pela atividade **Consultar Metadados** (ver figura 5.8). Essa consulta aos metadados tem a finalidade de identificar quais dados analíticos, que resultaram da consulta ao serviço analítico, possuem alguma correspondência geográfica. Estas informações são úteis tanto em uma requisição do tipo GEOMD de mapeamento quanto em uma requisição GEOMD de integração. No caso de uma consulta GEOMD de mapeamento, o acesso aos metadados indica qual dado analítico possui correspondência geográfica, bem como qual é o dado geográfico correspondente. Essas informações são úteis para formular a requisição geográfica que fornecerá o mapeamento esperado. Para o caso de uma consulta GEOMD de integração, o acesso aos metadados permite saber quais dados analíticos possuem correspondência geográfica, possibilitando, dessa forma, que a restrição contida na requisição geográfica seja aplicada.

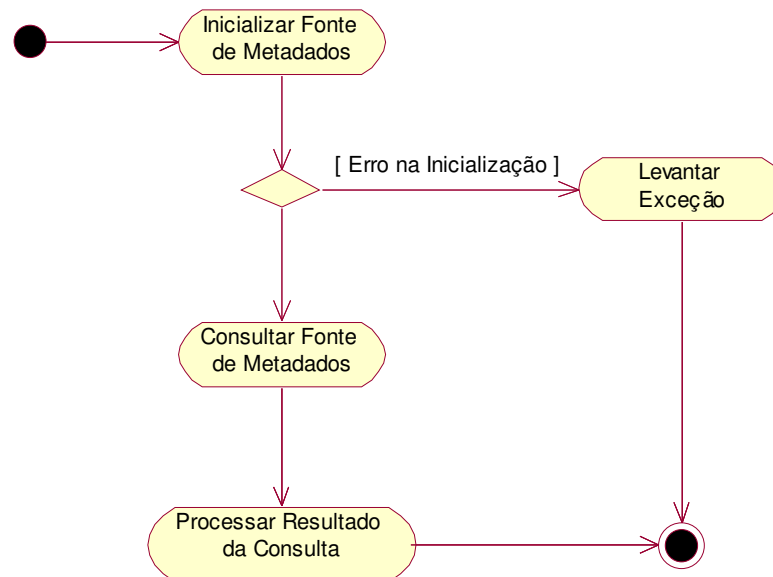


Figura 5.8: Expansão da atividade **Consultar Metadados**.

Após a atividade **Consultar Metadados** existe outro processo de decisão (ver figura 5.3). No caso do tipo da requisição ser GEOMD de integração, a próxima atividade a ser executada é **Consultar Serviço GEO** (ver figura 5.7), descrita anteriormente. Por outro lado, se o tipo da requisição for GEOMD de mapeamento, a atividade a ser executada será **Elaborar Requisição GEO**. A elaboração da requisição geográfica consiste em utilizar as informações fornecidas pela atividade **Consultar Metadados**, para montar uma consulta que será enviada ao

serviço geográfico. O objetivo é obter as correspondências geográficas dos dados analíticos recuperados pela atividade *Consultar Serviço MD* (ver figura 5.6).

Se o tipo da requisição que está sendo processada é GEOMD de integração, com o encerramento da atividade *Consultar Serviço GEO*, a próxima atividade a ser executada será *Integrar Resultados*. Nesta atividade, são recuperados os resultados das consultas aos serviços analítico e geográfico. A expansão desta atividade pode ser visualizada no diagrama da figura 5.9. Como pode ser analisado, após a recuperação de ambos os resultados, uma subatividade denominada *Aplicar Restrição Geográfica* é executada. Esta atividade tem a finalidade de eliminar os dados analíticos que não satisfazem a restrição espacial, aritmética ou de comparação contida na requisição. Em seguida, um documento contendo o resultado da integração é montado para ser enviado para a aplicação cliente que fez a requisição. Neste ponto também é encerrada a conexão com a base de metadados, a qual foi inicializada anteriormente.

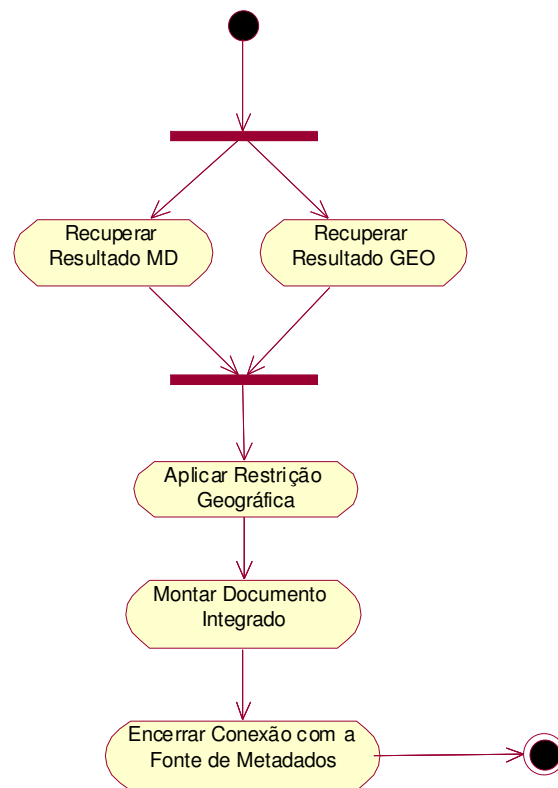


Figura 5.9: Expansão da atividade *Integrar Resultados*.

No caso de uma requisição ser do tipo GEOMD de mapeamento, após o término da atividade *Consultar Serviço GEO*, a próxima atividade a ser executada é a *Mapear Resultados*.

O diagrama que representa a expansão desta atividade é apresentado na figura 5.10. Basicamente, as subatividades de **Mapear Resultados** recuperam os resultados das consultas aos serviços analítico e geográfico, e identificam as correspondências existentes entre os dados analíticos e geográficos para montar o documento que irá conter a representação espacial dos dados analíticos. Finalmente, antes do encerramento dessa atividade, a conexão com a fonte de metadados, inicializada pela atividade **Consultar Metadados** é encerrada.

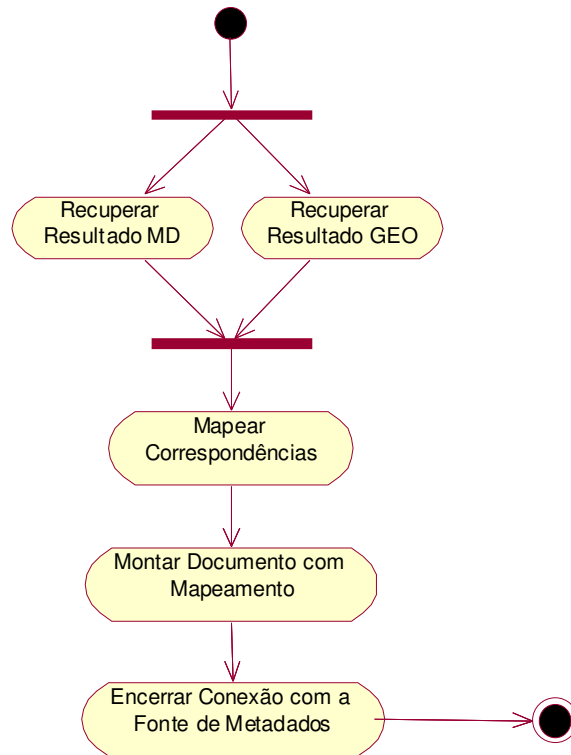


Figura 5.10: Expansão da atividade **Mapear Resultados**.

A última atividade do modelo a ser executada é a **Enviar Resposta da Requisição**. Esta atividade recupera o documento que contém o resultado do processamento da requisição e o envia para a aplicação cliente que realizou a consulta.

5.5 Considerações Finais

O mapeamento das correspondências entre as especificações do XMLA e do WFS, apresentado no item 5.2, foi de grande utilidade para a proposição do modelo ISAG, uma vez que apresenta em forma detalhada todo o funcionamento das implementações dessas especificações. Por sua vez, o esquema que define os tipos de requisições, apresentado na seção 5.2, é independente do

modelo de integração apresentado neste capítulo. Por estender a sintaxe de consulta proposta por duas especificações que são padrões de fato para o processamento geográfico e analítico via *Web*, o *GMLA Request Schema* pode ser adaptado e utilizado no desenvolvimento de outras soluções que visam a integração de serviços analíticos e geográficos que estejam baseados em XMLA e WFS.

O modelo de Integração de Serviços Analíticos e Geográficos (ISAG) [STF+03a] apresentado neste capítulo, foi validado através de uma implementação, e o resultado final foi um serviço *Web* que disponibiliza operações para a realização de consultas analíticas, geográficas ou ainda geográficas-multidimensionais. Os aspectos relacionados a essa implementação serão discutidos no próximo capítulo desta dissertação.

Embora a implementação do modelo, discutida no capítulo 6, esteja voltada para a arquitetura GOLAPA, a utilização do modelo é independente da mesma, podendo ser utilizado em outras soluções que visem a integração de processamento analítico e geográfico. Inclusive, como será discutido com maiores detalhes no capítulo 7, o modelo aqui apresentado poderia ser utilizado nas soluções propostas por alguns dos trabalhos relacionados, apresentados no capítulo 3, com a vantagem de focar na utilização de padrões abertos, extensíveis e independentes de plataforma.

Capítulo 6

GMLA Web Service

Neste capítulo, serão descritos alguns detalhes referentes à implementação do Modelo ISAG. Esta implementação resultou em um serviço denominado *GMLA Web Service*, que disponibiliza operações para realização de consultas a dados analíticos e/ou geográficos.

6.1 Introdução

Com intuito de validar as idéias apresentadas no capítulo anterior, neste capítulo, serão descritos alguns aspectos pertinentes à implementação do serviço *GMLA Web Service* [STF+03a, STF+03b], o qual é responsável por prover a integração dos processamentos analítico e geográfico, disponibilizados pelos serviços que implementam as especificações XMLA e WFS.

Este capítulo descreve aspectos relacionados à implementação do Modelo ISAG, apresentado no capítulo 5. São descritos os componentes da arquitetura do *GMLA Web Service* (GMLA WS), o relacionamento entre cada um deles e a forma com que foram implementados. São apresentados também, alguns exemplos de consultas processadas pelo GMLA WS, seguindo a estrutura definida pelo *GMLA Request Schema* (ver seção 5.3).

Neste capítulo, são utilizados conceitos pertinentes a todas as tecnologias relacionadas no capítulo 4. As implementações descritas neste capítulo também possibilitam um melhor entendimento de trabalhos relacionados, apresentados no capítulo 3, como a arquitetura GOLAPA, e o *GMLA Schema*. Além disso, serve para exemplificar e possibilitar um melhor entendimento de alguns dos conceitos listados no capítulo 2.

O restante deste capítulo está organizado da seguinte forma: Na seção 6.2, é apresentada a arquitetura do *GMLA Web Service*, incluindo seus principais componentes. Na seção 6.3, aspectos pertinentes à implementação do modelo ISAG são relatados. A seção 6.4 descreve a implementação das especificações XMLA e WFS. Por sua vez, a seção 6.5 detalha a implementação da fonte de metadados da integração. Na seção 6.6, é apresentada a forma de implementação do DWG como base de dados analíticos e geográficos, utilizada para validar as implementações do GMLA WS. A seção 6.7 apresenta a implementação de uma aplicação cliente para acesso ao serviço GMLA WS. Na seção 6.8, é descrito a forma com que são realizadas as consultas ao GMLA WS. Finalmente, na seção 6.9, são apresentadas algumas considerações finais sobre o presente capítulo.

6.2 Arquitetura do GMLA Web Service

O modelo ISAG, proposto no capítulo anterior, foi implementado utilizando tecnologias abertas como XML, JAVA e *Web Services*. Na figura 6.1, podem ser visualizados os principais componentes que resultaram na arquitetura que compõe o GMLA Web Service (GMLA WS). Dentre os principais componentes podem ser citados o serviço analítico (*SERVIÇO XMLA*), o serviço geográfico (*SERVIÇO WFS*), a base de metadados (*FONTE DE METADADOS*), a base de dados analíticos e geográficos (*DATA WAREHOUSE GEOGRÁFICO*) e o cliente que acessa o GMLA WS (*CLIENTE GMLA*). De certa forma, a arquitetura apresentada na figura 6.1 pode ser considerada uma instância da arquitetura GOLAPA, descrita no capítulo 3 (seção 3.3.1). Equivalências entre estas duas arquiteturas são dadas na tabela 6.1. Cada um dos componentes da figura 6.1, são descritos a seguir.

Tabela 6.1: Correspondências entre GOLAPA e a arquitetura do GMLA WS.

	Arquitetura GOLAPA	Arquitetura do GMLA WS
Componente	GOLAPE	GMLA WEB SERVICE
	GOLAPI	CLIENTE GMLA
	METADATA	FONTE DE METADADOS
	GDW	DATA WAREHOUSE GEOGRÁFICO
	GEOGRAPHICAL SERVICE	SERVIÇO WFS
	ANALYTICAL SERVICE	SERVIÇO XMLA

O *GMLA WEB SERVICE*, localizado no centro da figura 6.1, implementa o modelo ISAG. Para isso foi utilizada a linguagem de programação JAVA e a tecnologia dos *Web Services*. Este componente, denominado de GMLA WS, disponibiliza uma interface WSDL [WSDL01] que contém a descrição do serviço oferecido. Dessa forma, aplicações clientes podem ser desenvolvidas em qualquer linguagem de programação ou sistema operacional, e podem requisitar ao GMLA WS três tipos de consultas (*i.e.* MD, GEO e GEOMD), conforme definido pelo esquema *GMLA Request*. Para processar a requisição recebida, o GMLA WS tem a sua disposição um serviço analítico, o qual é voltado para o processamento analítico de dados multidimensionais, e um serviço geográfico, que por sua vez, é capaz de realizar operações geográficas sobre os dados armazenados em um *Data Warehouse Geográfico* (ver seção 2.6). Em caso de uma consulta de integração ou mapeamento (*i.e.* subdivisões de uma consulta GEOMD - tipo de requisição definida pelo *GMLA Request Schema*), o serviço GMLA WS pode acessar a fonte de metadados para verificar se um determinado dado analítico possui correspondência geográfica, bem como obter informações para recuperar os dados geográficos

que representam essa correspondência. Ao final do processamento, o GMLA WS irá devolver como resposta um documento XML contendo o resultado da consulta. Este documento é validado pelo *GMLA Schema* [FST+03a] (ver seção 3.3.1.2), o qual foi especificado para definir a estrutura de um documento que contenha dados analíticos e geográficos. O documento de resposta é então processado pela aplicação cliente, de forma que o resultado é apresentado graficamente através da utilização de SVG [SVG03] (ver seção 4.2.5) para desenhar o mapa com as geometrias das feições geográficas envolvidas na consulta e HTML para apresentar a tabela contendo os dados analíticos. Nas subseções a seguir, é descrito com maiores detalhes, o processamento dos tipos de requisições que podem ser enviadas ao GMLA WS.

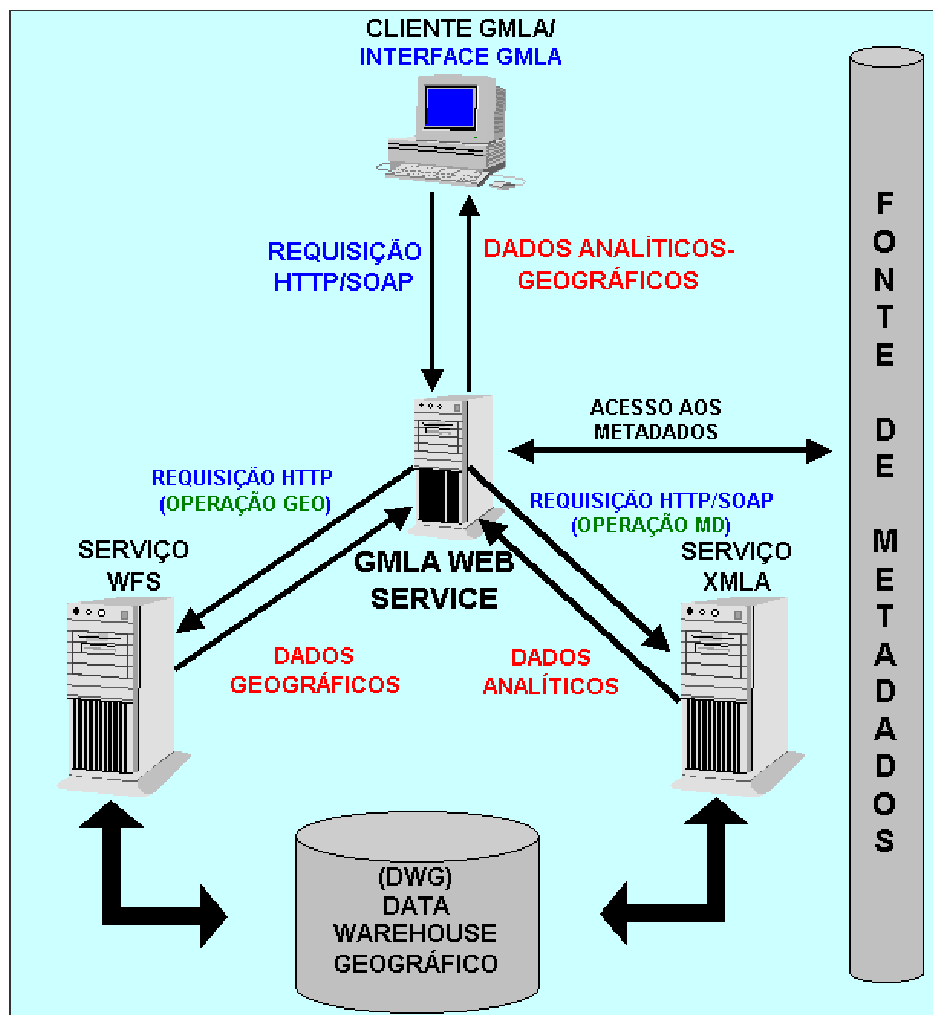


Figura 6.1: Arquitetura do GMLA WS.

6.3 Implementação do Modelo ISAG

O modelo ISAG foi instanciado, e sua implementação resultou na definição do serviço GMLA WS, como mecanismo de integração de serviços analíticos e geográficos. A figura 6.2 apresenta, utilizando a notação UML, a relação das classes JAVA que compõem o serviço GMLA WS. Nos parágrafos a seguir, as principais classes GMLA WS serão descritas com maiores detalhes, o que permitirá identificar quais classes são responsáveis pela execução das atividades previstas pelo modelo ISAG.

Na definição de um *Web Service* em JAVA, normalmente é descrita uma interface que define os principais métodos que serão disponibilizados pelo serviço. É através dessa interface que é gerado o documento WSDL (ver seção 4.3.1.4), o qual contém a descrição do serviço, formato dos parâmetros de entrada e saída, bem como a localização do serviço. No caso do GMLA WS, essa função é desempenhada pela interface *GMLAInterface*, como apresentado na figura 6.2. A interface do GMLA WS define apenas um método, o *GmlaRequest*, que recebe uma *string* como parâmetro de entrada, contendo a requisição enviada pela aplicação cliente, e retorna outra *string* contendo o resultado do processamento. No GMLA WS, essa interface é implementada pela classe *GMLAImplementation*, que será descrita nos próximos parágrafos.

Como todo *Web Service*, o GMLA WS também necessita de um mecanismo que publique o serviço em um determinado endereço, para que, a partir deste ponto, as aplicações clientes possam estabelecer uma conexão com o serviço para a troca de informações. Essa função é desempenhada pela classe *GMLAServer*, que utiliza a ferramenta GLUE [GLUE03] para publicar o serviço e gerenciar o envio/recebimento de informações codificadas de acordo com o padrão SOAP (ver seção 4.3.1.2).

Uma das classes mais importantes do GMLA WS é a *GMLAImplementation*. Esta classe é responsável pela execução da maioria das atividades definidas pelo modelo ISAG. Entre as funções desempenhadas pela classe *GMLAImplementation* estão: o recebimento das requisições, a verificação do tipo das requisições, a extração e validação dos parâmetros, a consulta ao serviço analítico e ao geográfico, o acesso à fonte de metadados, a integração ou mapeamento dos resultados, o levantamento de exceções no caso da ocorrência de algum erro, e, por fim, o envio do resultado do processamento para a aplicação cliente. Para desempenhar todas

estas atividades, esta classe requer o auxílio das demais classes listadas no diagrama da figura 6.2, as quais são descritas a seguir.

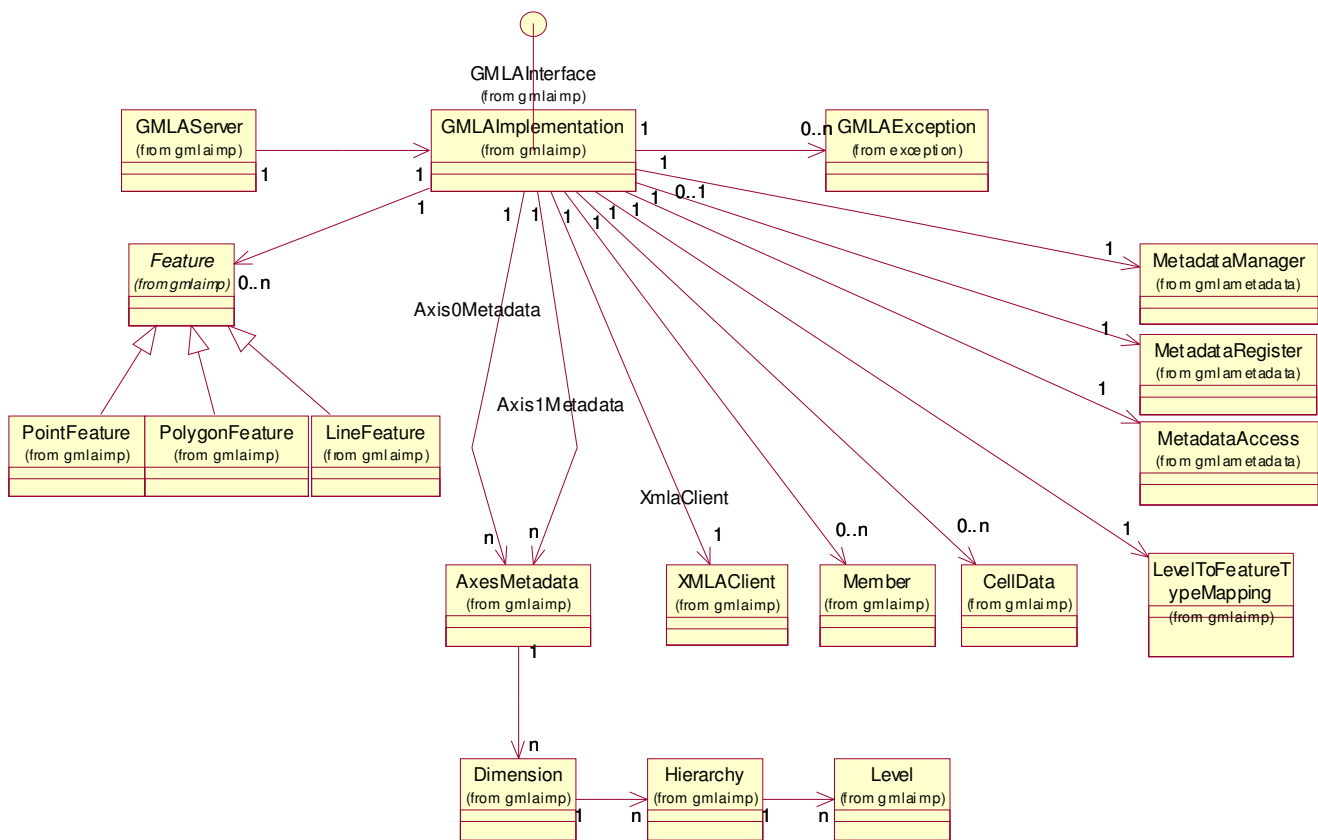


Figura 6.2: Diagrama de Classes do GMLA WS.

A classe *GMLAException* é utilizada para representar os possíveis erros que possam ocorrer durante o processamento de uma requisição. Assim que um erro ocorre, é criado um objeto dessa classe, contendo informações sobre a causa da exceção. Estas informações, especificando a causa do erro, são enviadas como resposta para a aplicação cliente.

Para manipulação das feições geográficas, resultantes das consultas ao serviço geográfico, o GMLA WS faz uso da classe abstrata *Feature*. Como pode ser visualizado no diagrama da figura 6.2, a classe *Feature* possui três especializações: 1) *PointFeature* – que é utilizada para representar geometrias pontuais, tais como a localização de lojas em um determinado mapa; 2) *PolygonFeature* – utilizada para representar uma feição geográfica em forma de polígono, como por exemplo, a geometria de uma cidade ou estado; e *LineFeature* –

que é utilizada para representar feições geográficas em forma de linha, tais como a geometria de um rio ou de uma rua.

Para realizar a comunicação com o serviço analítico, identificado na figura 6.1 como *SERVIÇO XMLA*, o GMLA WS utiliza a classe *XMLAClient*. Esta classe usa os pacotes fornecidos pelo Java WSDP (*Java™ Web Services Developer Pack 1.2*) [WSDP03], o qual é voltado para a manipulação e processamento de documentos XML, para criar os envelopes SOAP contendo as requisições que serão enviadas a este serviço (*i.e.* XMLA).

Quanto ao tratamento dos dados analíticos, resultantes das consultas ao serviço XMLA, o GMLA WS tem à disposição as classes *Member* e *CellData*. Com instâncias da classe *Member*, o GMLA WS manipula os dados analíticos, pertencentes a um determinado cubo de dados. Para isso, a classe *Member* possui atributos para identificar a dimensão, hierarquia e nível ao qual o dado pertence. Por sua vez, a classe *CellData* é utilizada para armazenar informações sobre as medidas de um cubo de dados, contidas no resultado processado pelo serviço XMLA.

Ainda no que se refere ao tratamento dos dados analíticos, quatro outras classes são consideradas de grande importância. As classes *AxesMetadata*, *Dimension*, *Hierarchy* e *Level* são utilizadas pelo GMLA WS para armazenar informações sobre os metadados dos eixos de um resultado enviado pelo XMLA. Todo resultado analítico possui dois eixos (*i.e.* *Eixo 0* e *Eixo 1*). Analisando o resultado analítico como uma tabela, o *Eixo 0* corresponde às colunas e o *Eixo 1* representa as linhas. Uma instância da classe *AxesMetadata*, utiliza vetores para armazenar informações sobre as dimensões do cubo de dados em questão, as quais são representadas por uma instância da classe *Dimension*. Por sua vez, uma instância da classe *Dimension* utiliza vetores para armazenar as informações sobre as hierarquias desta dimensão, as quais são representadas por instâncias da classe *Hierarchy*. As instâncias da classe *Hierarchy*, por sua vez, utilizam vetores para armazenar informações sobre os níveis de uma hierarquia, que são representados por instâncias da classe *Level*. Finalmente, as instâncias da classe *Level*, utilizam vetores para armazenar os identificadores de todos os membros analíticos envolvidos no processamento.

As classes *Dimension*, *Hierarchy* e *Level* possuem um atributo do tipo *boolean* chamado *GeographicalCorrespondence*, o qual é verdadeiro se o objeto em questão possui alguma correspondência geográfica, e falso, caso contrário. A atribuição do valor ao atributo

GeographicalCorrespondence é realizada através do acesso à fonte de metadados da integração, descrita na seção 6.5, podendo ser resumida da seguinte maneira: Para cada eixo resultante da consulta analítica (*i.e. Eixo 0 e Eixo 1*), são realizadas consultas de acesso à fonte de metadados, com intuito de descobrir se a dimensão em questão possui alguma correspondência geográfica. Se a resposta for verdadeira, o atributo *GeographicalCorrespondence* daquela dimensão é atualizado para verdadeiro, e a consulta continua para saber se as hierarquias daquela dimensão possuem alguma correspondência geográfica. Caso o retorno da consulta seja verdadeiro, o atributo *GeographicalCorrespondence* da hierarquia também é atualizado para verdadeiro e a consulta continua para identificar se os níveis daquela hierarquia possuem alguma correspondência geográfica. Este acesso à base de metadados, é provido pelas classes *MetadataManager* e *MetadataAccess*, que serão descritas a seguir. Com este conjunto de informações sobre os metadados dos eixos do resultado analítico, o GMLA WS está apto a recuperar os dados geográficos que satisfaçam consultas do tipo GEOMD, de mapeamento e de integração.

Para manipulação da base de metadados, o GMLA WS conta com as classes *MetadataRegister*, *MetadataManager* e *MetadataAccess*. A classe *MetadataRegister* é utilizada para inserir as informações na base de metadados, e portanto, é utilizada somente nos casos em que os metadados necessitam de alguma modificação, exclusão ou inserção. A classe *MetadataManager* é utilizada como gerenciador da base de metadados, desempenhando funções como inicialização e encerramento da conexão com a base. Enquanto que a classe *MetadataAccess* disponibiliza métodos para a realização das consultas aos metadados. Entre estes métodos, estão o *DimHasGeoCorrespondence*, *HieHasGeoCorrespondence* e *LevHasGeoCorrespondence* que consultam a base de metadados para identificar, respectivamente, se uma determinada dimensão, hierarquia ou nível possui correspondência geográfica. Estes métodos retornam verdadeiro para o caso da existência de uma correspondência geográfica, no caso contrário retornam falso. Outro método disponibilizado pela classe *MetadataAccess* é o *LevGeoCorrespondence*, que retorna o identificador da feição geográfica que corresponde a um determinado nível do resultado analítico. Maiores detalhes sobre a implementação da base de metadados são descritos na seção 6.5.

Outra classe que auxilia o GMLA WS no processamento das requisições é a *LevelToFeatureTypeMapping*. Esta classe é útil no processamento de requisições do tipo GEOMD de mapeamento (ver seção 5.3), para que, a partir dos níveis resultantes de um

resultado analítico, possam ser montados os temas contendo as feições geográficas correspondentes.

6.4 A Implementação das Especificações XMLA e WFS

Para implementar e validar o modelo ISAG, fez-se necessário o desenvolvimento de um Serviço Web que interagisse com os dois serviços (*i.e.* analítico e geográfico), possibilitando a integração no processamento. Como implementações para as especificações XMLA e WFS, apresentadas no capítulo 4, foram utilizados o *XML For Analysis Provider* (serviço analítico) [MSXA03] e o *Deegree Web Feature Service* (serviço geográfico) [DWFS03]. As subseções a seguir apresentam maiores informações sobre essas duas implementações.

6.4.1 Especificação XMLA

XML For Analysis Provider [MSXA03] foi desenvolvido pela *Microsoft* e implementa todas as funcionalidades contidas na especificação XMLA [XMLA02]. Por ser desenvolvido de acordo com o padrão dos *Web Services*, disponibiliza uma interface WSDL que permite que aplicações clientes sejam desenvolvidas independentes de plataforma ou linguagem de programação.

6.4.2 Especificação WFS

Embora existam outras implementações da especificação WFS, como por exemplo o *GeoServer* [GWFS03], optou-se pela utilização do *Deegree Web Feature Service* [DWFS03]. Por ser uma implementação de domínio público, ela possui uma limitação, pois somente oferece o operador espacial *BBOX* (ver seção 4.5.4.1). Entretanto, essa limitação não impossibilita a utilização desse serviço para a implementação do mecanismo de integração. Isso se deve ao fato de que a utilização dos demais operadores espaciais segue a mesma forma do *BBOX*, ou seja, para todo serviço implementado de acordo com a especificação WFS a entrada é uma expressão de filtro [FEIS03] (ver seção 4.5.4) e a resposta é um conjunto de feições geográficas no formato GML [GML03a] (ver seção 4.2.4), sendo que somente a forma de processamento interno das restrições espaciais aplicadas às feições geográficas é diferente para cada operador. Dessa forma, no momento em forem implementados outros operadores espaciais no *Deegree Web Feature*

Service, o GMLA WS irá incorporar automaticamente estas operações, sem haver a necessidade de alterações no modelo de integração de serviços nem mesmo no mecanismo de integração.

6.5 Fonte de Metadados

Como pode ser visualizado na figura 6.1, um dos componentes da arquitetura do GMLA WS é a fonte de metadados. Ela é baseada no modelo GAM e no esquema GeoMD e seu modelo de implementação (*i.e.* GeoMDM), definidos originalmente em [Fid03], e apresentados de forma resumida na seção 3.3.1.3 deste documento.

A função do componente *FONTE DE METADADOS* é armazenar informações sobre os dados analíticos e geográficos disponíveis no *Data Warehouse* Geográfico. A partir dos metadados é possível identificar as correspondências existentes entre os dados analíticos e geográficos, bem como é possível obter informações que possibilitam a recuperação destas correspondências. A fonte de metadados é indispensável para o processamento de requisições GEOMD (ver seção 5.3), seja ela de mapeamento ou de integração, devido a esse tipo de consulta necessitar de informações tais como: saber se um determinado dado analítico possui alguma correspondência geográfica, qual o dado geográfico correspondente e como proceder para recuperá-lo.

A implementação da fonte de metadados poderia ter sido realizada utilizando-se uma base de dados relacional, onde os metadados seriam armazenados em tabelas e a recuperação de informações seria feita através de consultas SQL. Entretanto, como toda arquitetura GOLAPA está baseada em padrões abertos e extensíveis, optou-se pela utilização do padrão MOF (*MetaObject Facility*) [MOF02]. MOF é uma especificação criada pela OMG (*Object Management Group*) [OMG03] na tentativa de oferecer uma forma padrão para a definição e representação de metamodelos. MOF suporta qualquer forma de metadados que pode ser descrita utilizando técnicas de modelagem orientada a objetos e, através do padrão XMI (*XML Metadata Interchange*) [XMI02], possibilita o intercâmbio de modelos de metadados entre aplicações que são baseadas na especificação MOF, utilizando a tecnologia XML .

Outro padrão utilizado na implementação da fonte de metadados foi o JMI (*Java Metadata Interface*) [JMI02], o qual corresponde ao mapeamento de MOF para Java. Entre

outras funcionalidades, o JMI possibilita a criação, armazenamento, consulta e intercâmbio de metadados.

Com as tecnologias descritas nos parágrafos anteriores (*i.e.* MOF, XMI e JMI), é possível que um modelo de metadados, descrito através da linguagem de modelagem UML e que esteja conforme a especificação MOF, possa ser exportado para o formato XMI. Tendo disponível o documento XMI, descrevendo a estrutura do modelo de metadados, através da utilização de ferramentas como o *NetBeans* [NBe03], pode-se gerar automaticamente as interfaces JMI que possibilitam a manipulação dos metadados em um repositório. Como repositório, o *NetBeans* oferece o MDR (*MetaData Repository*), que também é implementado seguindo a especificação MOF.

No caso do serviço GMLA WS descrito neste capítulo, a fonte de metadados foi implementada utilizando MOF, XMI e JMI, fazendo uso do repositório MDR, que pode ser desintegrado da ferramenta *NetBeans* e utilizado de forma independente. Vale salientar que poderiam ser utilizados outros repositórios que são implementados de acordo com a especificação MOF. Um exemplo que pode ser citado é o *dMOF* [DMOF03]. Entretanto, optou-se pela utilização do MDR simplesmente pelo fato dele ser um projeto livre, de domínio público e por suprir as necessidades da implementação do modelo de integração proposto nesta dissertação.

6.6 A Base de Dados Analíticos e Geográficos

Como demonstrado na figura 6.1, que apresenta a arquitetura do GMLA WS, um dos componentes é o *Data Warehouse Geográfico* (DWG). O esquema conceitual de um DWG é definido por [Fid03]. O esquema é voltado para utilização na camada (I) da arquitetura GOLAPA (ver seção 3.3.1), é independente da mesma e pode ser empregado em outras soluções que necessitem disponibilizar dados analíticos e geográficos para suporte à decisão. A implementação do DWG foi baseada no *GeoDWFrame* (ver seção 3.3.1.4). Entretanto, como ainda não dispomos da implementação de uma ferramenta de ETL geográfico para efetuar a carga dos dados no DWG, como prevê a arquitetura GOLAPA, a carga foi realizada manualmente. A seguir é descrita a origem dos dados analíticos e geográficos.

Para representar os dados analíticos foi utilizada a base de dados de exemplo da *Microsoft*, o *FoodMart 2000*, o qual é disponibilizado juntamente com o CD de instalação do *Microsoft SQL Server 2000* [MSQL03]. A base *FoodMart* disponibiliza dados de uma grande cadeia de supermercados, com vendas nos Estados Unidos, México e Canadá. Ela contém alguns cubos de dados analíticos para apresentação dos dados relativos às vendas (*Sales*), recursos humanos (*HR*), orçamento de despesas (*Budget*), estoque (*Warehouse*), entre outros. Estes cubos OLAP compartilham algumas dimensões como produto (*Product*), categoria (*Category*), tempo (*Time*), loja (*Store*) e departamento (*Department*).

Na figura 6.3, é possível visualizar a estrutura da base de dados *FoodMart*. Nesta figura, é possível identificar os cubos de dados analíticos, juntamente com as dimensões que compõem estes cubos. A estrutura apresentada na figura 6.3 é resultado do uso da ferramenta *Analysis Manager*, também disponível no CD de instalação do *Microsoft SQL Server 2000*.

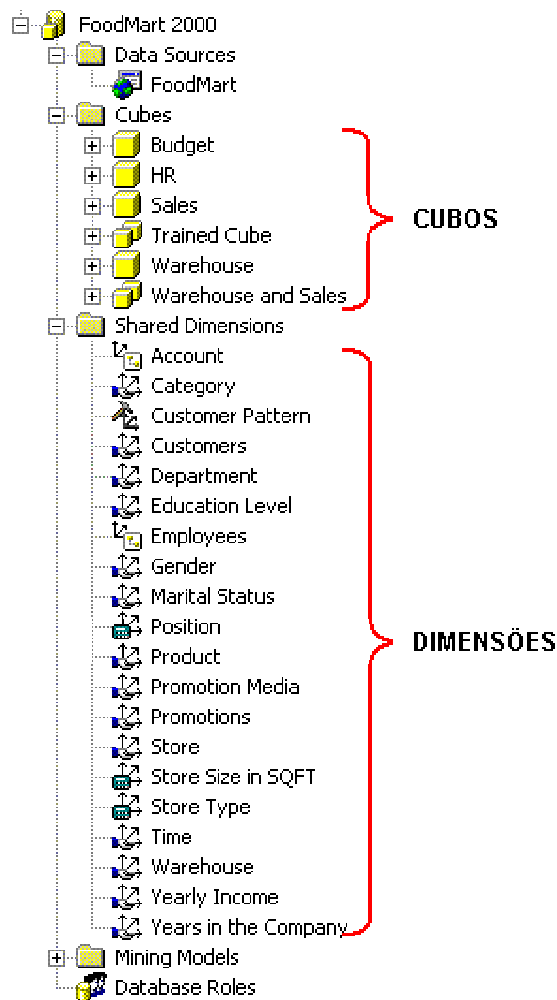


Figura 6.3: Estrutura da base de dados *FoodMart 2000*.

Para representar as dimensões que contêm os dados geográficos no DWG, foram utilizados dados que descrevem a geometria dos países, estados e cidades onde são realizadas as vendas registradas no *FoodMart 2000*. Estas geometrias são armazenadas no DWG no formato GML (ver seção 4.2.4), em tabelas do *SQL Server*, da mesma forma que os dados analíticos. Para a construção do DWG foram utilizados os conceitos definidos pelo arcabouço GeoDWFrame [Fid03] (ver seção 3.3.1.4), onde os dados analíticos estão agrupados em dimensões convencionais, e os geográficos em uma dimensão híbrida conjunta., a qual é relacionada com as dimensões primitivas que contêm as geometrias dos dados geográficos. Essas geometrias em GML foram geradas a partir da ferramenta *Viewer/Converter*. Esta ferramenta encontra-se disponibilizada de forma aberta pelo projeto *Deegree* [DWFS03], e transforma para GML, dados geométricos que estão em formatos proprietários, como por exemplo os *shape-files* (i.e. arquivo com extensão .SHP) do *ArcView* [ESRI03]. Outra forma de gerar dados GML é com a utilização de uma extensão do *ArcView* chamada *ToWKT*, a qual é disponibilizada juntamente com o projeto *GeoServer* [GWFS03]. Neste trabalho, foi utilizada a ferramenta *Viewer/Converter* para gerar as geometrias em GML, simplesmente pelo fato dela pertencer ao mesmo projeto do serviço geográfico (i.e. *Deegree Web Feature Service* [DWFS03]).

Nesta seção, relatamos como foi construído o DWG, que disponibiliza uma base com dados analíticos e geográficos, para que o GMLA WS utilize-a para recuperar os dados requeridos pelas consultas a ele enviadas. Salienta-se que todo esse processo deverá ser automatizado pelos componentes presentes na camada (B) da arquitetura GOLAPA. Apesar da presente dissertação estar voltada especificamente para a camada (II) de GOLAPA, a implementação do DWG, com a carga manual dos dados, se fez necessária para testar as funcionalidades oferecidas pelo GMLA WS.

6.7 O Cliente GMLA

Como aplicação cliente, foi desenvolvido um programa em JAVA, que envia as requisições para o serviço GMLA WS. Esta aplicação pode ser vista como a interface da arquitetura GOLAPA, apresentada na seção 3.3.1 dessa dissertação. As funcionalidades são um pouco limitadas devido a esta interface ser caracterizada como um simples protótipo para a visualização dos resultados gerados pelo GMLA WS.

Como a comunicação segue o padrão dos *Web Services*, a aplicação cliente monta um envelope SOAP [SOAP03] contendo a requisição previamente validada pelo *GMLA Request Schema*, estabelece a conexão com o serviço GMLA WS e envia o envelope. Quando recebe o resultado do processamento, os dados são visualizados graficamente em mapas (para os dados geográficos) e tabelas (para os dados analíticos).

Para a visualização dos dados geográficos é utilizada a tecnologia SVG. A aplicação recupera os dados geográficos, codificados em GML e realiza um processamento, utilizando um *parser* Java, a fim de transformá-los para o formato SVG. Quanto aos dados analíticos, é realizado um processamento com XSLT para transformá-los para o formato HTML e exibi-los em uma tabela.

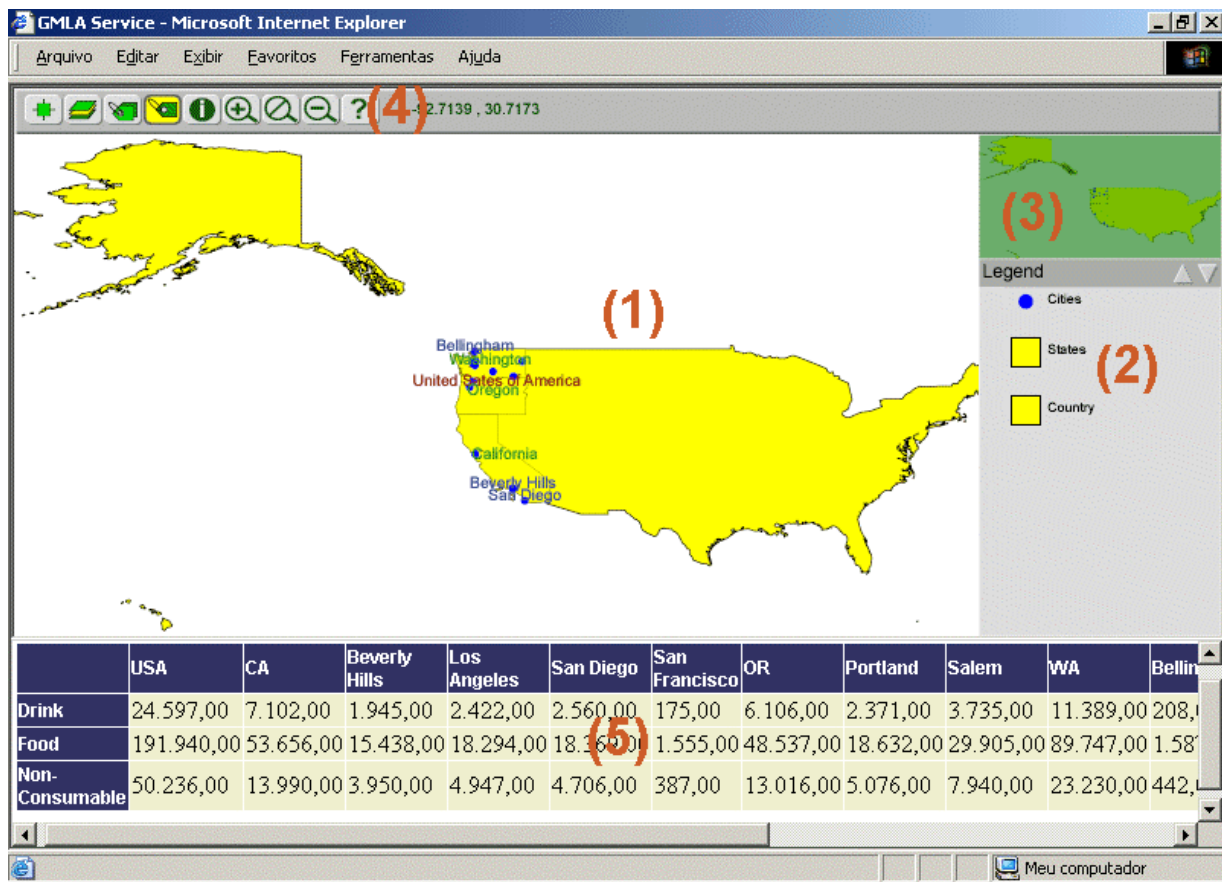











Figura 6.4: Interface do Cliente GMLA.

A interface gráfica da aplicação cliente pode ser dividida em cinco principais áreas (ver figura 6.4), as quais podem ser descritas da seguinte maneira:

- (1) – Esta é uma das principais áreas da interface. É nesta área que o mapa resultante de uma consulta GEO ou GEOMD é desenhado.
- (2) – Nesta área se encontra a legenda, a qual contém a relação de temas resultantes da consulta.
- (3) – Esta área permite a utilização do operador Pan, o qual possibilita selecionar uma determinada posição do mapa, arrastando a imagem em alguma direção, para ser visualizada na área (1).
- (4) – Na área (4) se encontra uma barra de ferramentas, contendo alguns botões que têm suas funcionalidades descritas na tabela 6.2.
- (5) – A área (5) se encarrega da visualização gráfica dos dados analíticos presentes no resultado de uma consulta GEOMD ou MD. Em caso de uma consulta GEO, esta área não irá aparecer. Por outro lado, em caso de uma consulta MD, somente esta área será visualizada.

Tabela 6.2: Barra de ferramentas da interface do Cliente GMLA.

Botão	Funcionalidade
	Habilita/Desabilita o operador Pan, o qual possibilita arrastar a imagem em alguma direção, dentro da área de visualização.
	Habilita/Desabilita a legenda da interface, a qual permite o controle da visualização dos temas resultantes de uma consulta GEO ou GEOMD.
	Exibe/Ocultar os nomes de todas as feições de todos os temas em questão.
	Exibe/Ocultar os nomes de feições selecionadas individualmente pelo usuário. Basta um clique sobre a feição para ocultar ou exibir seu nome.
	Exibe informações sobre uma feição selecionada pelo usuário. Com um simples clique sobre a feição, é exibida em uma janela contendo as informações sobre a feição selecionada.
	Mais Zoom, aproxima a imagem com alta resolução.
	Adapta o tamanho do mapa de acordo com o tamanho da área de exibição.
	Menos Zoom, afasta a imagem.
	Exibe informações sobre a interface.

É importante ressaltar que o módulo da interface gráfica que manipula os dados geográficos, que são transformados para o formato SVG, faz parte do projeto *GeoServer*

[GWFS03]. Este projeto é distribuído sob os termos de licença GNU (*General Public License*) [GNU03], pode ser baixado e modificado livremente. Dessa forma, a interface foi modificada a fim de receber os dados que foram transformados de GML para SVG pela aplicação cliente, e em seguida, exibir os resultados em forma de mapas. A interface obtida foi desenvolvida utilizando *Java Script* e SVG. Ela disponibiliza algumas funcionalidades que permitem a manipulação das informações geográficas exibidas, as quais são listadas na tabela 6.2. A interface produzida pode ser exibida em um navegador da Internet que processe SVG. No caso das implementações aqui descritas, foi utilizado o navegador *Microsoft Internet Explorer* [MIE03], equipado com o visualizador SVG *SVGViewer* da Adobe [SVG03].

Salienta-se que poderiam ser utilizadas outras tecnologias para a apresentação gráfica dos dados geográficos, como por exemplo o VML (*Vector Markup Language*) [VML98]. Entretanto, optou-se por SVG [SVG03]. por ser uma especificação que tende a se tornar um padrão para disponibilização de dados vetoriais na Internet. Outro motivo para a escolha de SVG se deve ao fato da interface utilizada estar baseada nessa tecnologia e estar à disposição para a livre utilização. Isto minimizou o tempo de trabalho, até porque, o desenvolvimento de uma interface gráfica mais elaborada, é objeto de estudo de outra dissertação de mestrado ligada ao projeto GOLAPA e ainda em andamento.

6.8 Tipos de Requisições Processadas pelo GMLA WS

Nesta seção, serão apresentados alguns exemplos de requisições que podem ser enviadas ao GMLA WS. Alguns exemplos de requisições do tipo MD, GEO e GEOMD serão listados, incluindo a visualização gráfica dos seus resultados. A descrição detalhada de cada um dos tipos de requisição processada pelo GMLA WS foi apresentada na seção 5.3.

6.8.1 Requisições do Tipo MD

Nesta seção, é descrito como ocorre a execução de uma requisição do tipo MD no GMLA *Web Service*. Uma requisição MD é aquela que contém somente parâmetros analíticos, ou seja, somente o serviço XMLA é consultado pelo GMLA WS. Dessa forma, será considerado uma requisição analítica que possibilite ao GMLA WS listar o total das vendas de produtos, classificando pela categoria do produto e pelo estado onde as vendas foram efetuadas. Para isto, é necessário enviar ao GMLA WS uma solicitação do tipo MD. O GMLA WS recebe esta

consulta, extrai os parâmetros necessários e em seguida, requisita o serviço XMLA, o qual é responsável pela execução das consultas analíticas à base de dados. Nesta consulta, necessariamente existirá um comando MDX (*Multidimensional Expressions*) [MDX03a, MDX03b], que será executado pelo serviço XMLA. No caso do exemplo citado, onde pretende-se listar os totais de vendas dos produtos, o comando MDX correspondente pode ser visualizado na figura 6.5

```
SELECT
    NON EMPTY(DRILLDOWNLEVEL([Store].[Store Country].members)) ON COLUMNS,
    NON EMPTY([Product].[Product Family].members) ON ROWS
FROM Sales
```

Figura 6.5: Exemplo de comando MDX.

```
<?xml version="1.0" encoding="UTF-8"?>
<GmlaRequest xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="GMLARequest.xsd">
  <MD>
    <XMLAServiceAddress>http://localhost/xmla/msxisapi.dll</XMLAServiceAddress>
    <OLAPCube>Sales</OLAPCube>
    <XMLAMethod>Execute</XMLAMethod>
    <Command>
      <Statement>
        SELECT
          NON EMPTY(DRILLDOWNLEVEL([Store].[Store Country].members))
          ON COLUMNS,
          NON EMPTY([Product].[Product Family].members) ON ROWS
        FROM Sales
      </Statement>
    </Command>
    <Properties>
      <PropertyList>
        <DataSourceInfo>Provider=MSOLAP;Data Source=local </DataSourceInfo>
        <Catalog>FoodMart 2000</Catalog>
        <Format>Multidimensional</Format>
        <AxisFormat>TupleFormat</AxisFormat>
      </PropertyList>
    </Properties>
  </MD>
</GmlaRequest>
```

Figura 6.6: Exemplo de requisição MD.

Conforme a sintaxe da linguagem de consulta MDX [MDX03a, MDX03b], o comando apresentado na figura 6.5, irá executar uma consulta ao cubo vendas (*Sales*), de forma que no eixo colunas (*ON COLUMNS*), sejam listados os membros da dimensão loja (*Store*). Por outro lado, no eixo linhas (*ON ROWS*), serão listados os membros da dimensão produto

(*Product*). A indicação *NON EMPTY* possibilita que somente membros com valores sejam visualizados. A indicação *DRILLDOWNLEVEL*, permite a execução do operador analítico *Drill-Down* (ver seção 2.4.1). Com isso torna-se possível um maior detalhamento do resultado, possibilitando a agregação das vendas por estados.

As requisições enviadas ao serviço GMLA WS devem obedecer à estrutura estabelecida pelo *GMLARequest Schema*, apresentado na seção 5.3, o qual foi criado para definir a estrutura dos documentos XML contendo as requisições enviadas ao GMLA WS. Na figura 6.6, é apresentado o documento XML contendo o exemplo de consulta MD discutido anteriormente.

Toda requisição enviada ao serviço GMLA WS é processada para fins de: (1) classificação (*i.e.* verificar se seu tipo é MD, GEO ou GEOMD), (2) verificação (*i.e.* no caso de uma consulta MD ou GEOMD, o GMLA WS verifica se o nome do cubo a ser consultado, expresso com o elemento *OLAPCube*, é o mesmo que está sendo utilizado no comando MDX do elemento *Command*), e (3) extração dos demais parâmetros necessários para o processamento da consulta. No caso do exemplo apresentado na figura 6.6, a requisição é classificada como MD. Assim, o GMLA WS envia uma requisição ao serviço XMLA, que após executá-la, devolve ao GMLA WS, um documento XML contendo o resultado da consulta.

Quando o GMLA recebe a resposta do XMLA, esta é processada e enviada para o cliente que efetuou a requisição, o qual fica responsável pela representação gráfica dos dados resultante da consulta. Salienta-se que o documento resultante enviado para a aplicação cliente segue a estrutura definida pelo *GMLA Schema* (ver seção 3.3.1.2). A figura 6.7 apresenta o resultado da consulta analítica da figura 6.6, o qual foi previamente processado pela aplicação cliente para que os dados fossem demonstrados graficamente.

	USA	CA	OR	WA
Drink	24.597,00	7.102,00	6.106,00	11.389,00
Food	191.940,00	53.656,00	48.537,00	89.747,00
Non-Consumable	50.236,00	13.990,00	13.016,00	23.230,00

Figura 6.7: Resultado gráfico da consulta MD expressa na figura 6.6.

Outro exemplo de consulta analítica enviada ao GMLA WS é o apresentado na figura 6.8. Neste caso, o comando MDX lista as 10 primeiras lojas com maior valor de vendas, agrupando os totais das outras lojas em uma única linha denominada “*Outras Cidades*”. No eixo coluna são listadas as medidas do cubo vendas (*Sales*), entre elas o preço de custo (*Store Cost*), preço de venda (*Store Sales*) e o lucro líquido (*Store Sales Net*).

```
WITH SET DezMajoresVendas AS
'TOPCOUNT([Store].[Store City].MEMBERS, 10, [Sales Count])'
MEMBER [Store].[Outras Cidades] AS '([Store].[All Stores], Measures.CURRENTMEMBER)
- SUM(DezMajoresVendas, Measures.CURRENTMEMBER)'
SELECT {Measures.MEMBERS} ON COLUMNS, {DezMajoresVendas, [Store].[Outras
Cidades]} ON ROWS
FROM [Sales]
```

Figura 6.8: Exemplo de comando MDX para listar as dez primeiras lojas com o maior valor de vendas.

Se substituirmos o comando MDX, contido no elemento <*Statement*>, da requisição apresentada na figura 6.6 pelo comando da figura 6.8, e enviarmos ao GMLA WS, o resultado gráfico final será semelhante ao que pode ser visualizado na figura 6.9.

	Unit Sales	Store Cost	Store Sales	Sales Count	Store Sales Net
Salem	41.580,00	34.823,56	R\$ 87.218,28	13347	52.394,72
Tacoma	35.257,00	29.959,28	R\$ 74.843,96	11184	44.884,68
Portland	26.079,00	21.948,94	R\$ 55.058,79	8264	33.109,85
Los Angeles	25.663,00	21.771,54	R\$ 54.545,28	8207	32.773,74
San Diego	25.635,00	21.713,53	R\$ 54.431,14	8095	32.717,61
Seattle	25.011,00	20.956,80	R\$ 52.644,07	7956	31.687,27
Bremerton	24.576,00	21.121,96	R\$ 52.896,30	7876	31.774,34
Spokane	23.591,00	19.795,49	R\$ 49.634,46	7397	29.838,97
Beverly Hills	21.333,00	18.266,44	R\$ 45.750,24	6815	27.483,80
Yakima	11.491,00	9.713,81	R\$ 24.329,23	3652	14.615,42
Outras Cidades	6.557,00	5.555,87	13886,3800000001	4044	8.330,51

Figura 6.9: Resultado gráfico da consulta apresentada na figura 6.8.

6.8.2 Requisições do Tipo GEO

Nesta seção, é descrito detalhadamente como ocorre a execução de uma consulta do tipo GEO no serviço GMLA WS. Uma consulta do tipo GEO é aquela que requer somente consulta a dados geográficos. Dessa forma, somente o serviço WFS é requisitado.

Para demonstrar a execução de uma consulta geográfica no GMLA, será utilizado o operador espacial *BBOX* (seção 4.5.4.1), o qual recupera um conjunto de feições geográficas, contidas dentro da área retangular formada pelos pares de coordenadas enviadas na requisição. Na figura 6.10, pode ser visualizado o documento XML contendo um exemplo de consulta geográfica que pode ser enviada ao serviço GMLA WS. Este documento XML também é validado pelo *GMLARequest Schema*, que é o responsável pela definição da estrutura dos documentos contendo as requisições submetidas ao GMLA WS.

```
<GmlaRequest xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="D:\GMLARequest.xsd">
  <GEO>
    <WFSServiceAddress>http://localhost:8080/deegreewfs/deegreewfs</WFSServiceAddress>
    <GeoDB>UsaGeoDB</GeoDB>
    <GeoSpace>USAGeoSpace</GeoSpace>
    <GeoView>USAGeoView</GeoView>
    <GetFeature>
      <wfs:GetFeature
outputFormat="GML2" xmlns:gml="http://www.opengis.net/gml"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:ogc="http://www.opengis.net/ogc">
        <wfs:Query typeName="Country"> </wfs:Query>
        <wfs:Query typeName="States">
          <ogc:Filter>
            <ogc:BBOX>
              <wfs:PropertyName>/States/Border</wfs:PropertyName>
              <gml:Box>
                <gml:coord>
                  <gml:X>-124.73276978</gml:X>
                  <gml:Y>48.99993133</gml:Y>
                </gml:coord>
                <gml:coord>
                  <gml:X>-116.91913243</gml:X>
                  <gml:Y>45.54309283</gml:Y>
                </gml:coord>
              </gml:Box>
            </ogc:BBOX>
          </ogc:Filter>
        </wfs:Query>
      </wfs:GetFeature>
    </GetFeature>
  </GEO>
</GmlaRequest>
```

Figura 6.10: Exemplo de consulta geográfica (GEO).

Os parâmetros *GeoDB*, *USAGeoSpace* e *USAGeoView*, presentes nas requisições GEO e GEOMD, correspondem, respectivamente, ao banco de dados geográfico, espaço geográfico e visão geográfica envolvidos na consulta. Estes conceitos são definidos pelo modelo GAM e pelo esquema GeoMD (ver seção 3.3.1.3) para representar as correspondências dos metadados pertencentes aos ambientes de processamento analítico e geográfico. Os demais parâmetros das requisições enviadas ao GMLA WS são descritos na seção 5.3 deste documento.

A primeira atividade que o GMLA WS executa após receber uma nova requisição, é fazer a classificação da mesma. No caso da consulta apresentada na figura 6.10, ela é classificada como geográfica (GEO), portanto, o GMLA WS acessa somente o serviço WFS, que é o responsável por processar as consultas geográficas. Em seguida, o GMLA WS extrai os parâmetros necessários e envia a consulta ao WFS.

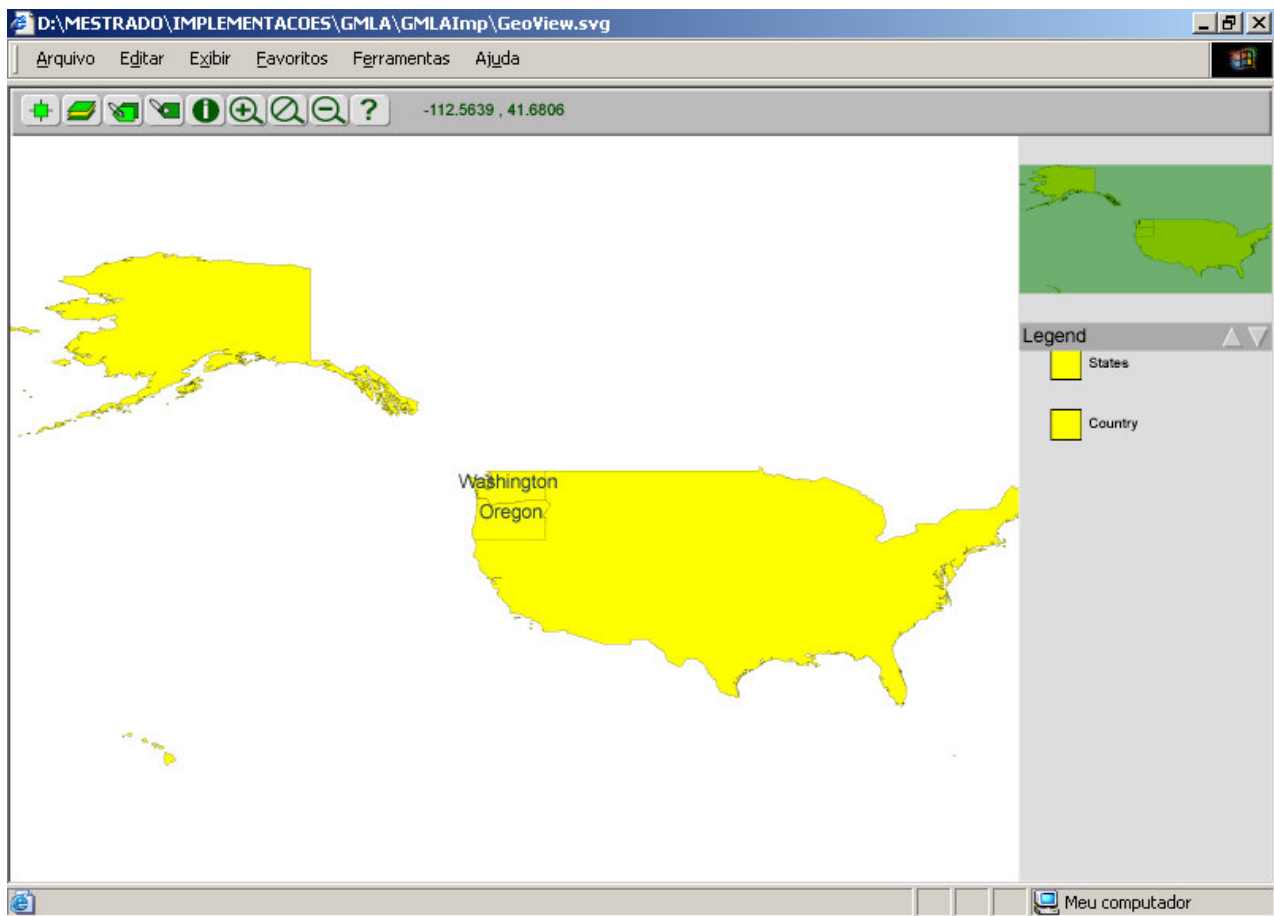


Figura 6.11: Feições geográficas resultantes da consulta GEO apresentada na figura 6.10.

De posse do resultado da consulta processada pelo WFS, o GMLA codifica os dados resultantes em um envelope SOAP, e em seguida, envia o resultado para a aplicação cliente

responsável pela submissão da requisição. A aplicação cliente processa os dados contidos no documento recebido como resposta, validando o mesmo com o esquema *GMLA Schema* (seção 3.3.1.2) e em seguida, apresenta graficamente esses dados. A visualização gráfica correspondente à consulta geográfica apresentada na figura 6.10, pode ser observada na figura 6.11. Nesta figura, pode-se observar que as feições geográficas que satisfizeram ao operador espacial *BBOX*, foram as feições correspondentes às geometrias dos estados *Washington* e *Oregon*, pertencentes ao tema geográfico que representa os Estados Unidos e seus estados.

Outro exemplo de consulta geográfica que pode ser submetida ao serviço GMLA WS é apresentada na figura 6.12. Nesta requisição, não está sendo utilizado nenhum operador espacial, sendo listadas, sem nenhuma restrição, todas as feições geográficas pertencentes aos temas país (*Country*), estados (*States*) e cidades (*Cities*). O resultado gráfico desta consulta pode ser visualizado na figura 6.13.

```
<GmlaRequest xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="D:\ GMLARequest.xsd">
  <GEO>
    <WFSServiceAddress>http://localhost:8080/deegreewfs/deegreewfs</WFSServiceAddress>
    <GeoDB>UsaGeoDB</GeoDB>
    <GeoSpace>USAGeoSpace</GeoSpace>
    <GeoView>USAGeoView</GeoView>
    <GetFeature>
      <wfs:GetFeature outputFormat="GML2"
xmlns:gml="http://www.opengis.net/gml"
xmlns:wfs="http://www.opengis.net/wfs"
xmlns:ogc="http://www.opengis.net/ogc">
        <wfs:Query typeName="Country"/>
        <wfs:Query typeName="States"/>
        <wfs:Query typeName="Cities"/>
      </wfs:GetFeature>
    </GetFeature>
  </GEO>
</GmlaRequest>
```

Figura 6.12: Consulta GEO para recuperar os temas País, Estados e Cidades.


```

<GmlaRequest xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="D:\GMLARequest.xsd">
  <GEOMD>
    <GEOMDSchema>GOLAPAGMDM001</GEOMDSchema>
    <MD>
      <XMLAServiceAddress>http://localhost/xmla/msxisapi.dll</XMLAServiceAddress>
      <XMLAMethod>Execute</XMLAMethod>
      <OLAPCube>Sales</OLAPCube>
      <Command>
        <Statement>
          SELECT NON EMPTY(DRILLDOWNLEVEL(DRILLDOWNLEVEL([Store].[Store
Country].members)))
          ON COLUMNS,NON EMPTY([Product].[Product Family].members) ON ROWS
          FROM Sales
        </Statement>
      </Command>
      <Properties>
        <PropertyList>
          <DataSourceInfo>Provider=MSOLAP;Data Source=local </DataSourceInfo>
          <Catalog>Foodmart 2000</Catalog>
          <Format>Multidimensional</Format>
          <AxisFormat>TupleFormat</AxisFormat>
        </PropertyList>
      </Properties>
    </MD>
    <GEO>
      <WFSServiceAddress>http://localhost:8080/deegreewfs/deegreewfs</WFSServiceAddress>
      <GeoDB>USAGeoDB</GeoDB>
      <GeoSpace>USAGeoSpace</GeoSpace>
      <GeoView>USAGeoView</GeoView>
      <GetFeature> </GetFeature>
    </GEO>
  </GEOMD>
</GmlaRequest>

```

Figura 6.14: Exemplo de requisição GEOMD de mapeamento.

Como definido pelo *GMLA Request Schema*, apresentado na seção 5.3, uma requisição do tipo GEOMD pode ser considerada de mapeamento ou de integração. Uma requisição GEOMD de mapeamento é executada quando se deseja realizar uma consulta analítica e visualizar os dados que possuem alguma correspondência geográfica em um mapa. Para isso, é necessário informar na consulta, qual será o esquema de integração utilizado, e os demais parâmetros presentes em toda requisição analítica. Também deverão ser fornecidos o nome do banco de dados geográficos, o nome do espaço geográfico e o nome da visão geográfica.

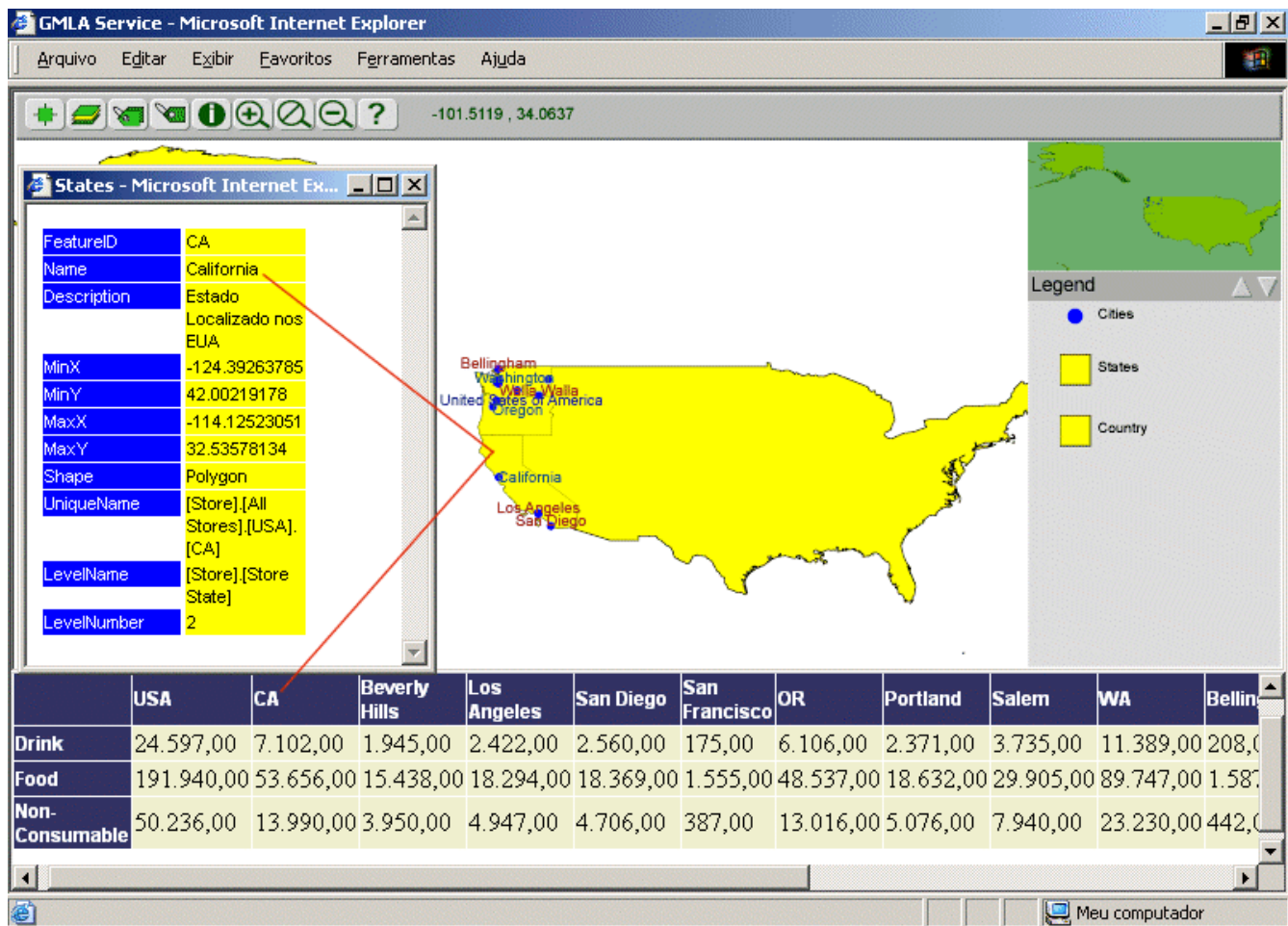


Figura 6.15: Visualização gráfica do resultado de uma consulta GEOMD de mapeamento.

A figura 6.14 apresenta um exemplo de requisição GEOMD de mapeamento, onde, os parâmetros analíticos consistem na relação dos totais de vendas de produtos, classificando o resultado por categoria do produto e pela cidade, estado e país onde as vendas foram efetuadas. Através do acesso à fonte de metadados de integração, descrita na seção 6.5, o GMLA WS processa a requisição, recuperando da base de dados analíticos e geográficos, as feições geográficas que correspondem aos dados analíticos envolvidos na consulta. Como em toda requisição processada pelo GMLA WS, o documento resultante, estruturado de acordo com o *GMLA Schema*, é enviado como resposta para aplicação cliente, a qual é responsável pela apresentação gráfica dos resultados. Na figura 6.15 é possível visualizar o resultado gráfico da requisição GEOMD apresentada na figura 6.14.

Como pode ser visualizado na figura 6.15, a interface gráfica da aplicação cliente apresenta os dados geográficos em forma de mapa, na parte superior da tela, e os dados analíticos em forma de tabela na parte inferior. As correspondências entre os dados analíticos e geográficos podem ser visualizadas claramente nesta figura. Por exemplo, na tabela contendo os dados analíticos (*i.e.* totais das vendas dos produtos por categoria, distribuídos pelas cidades e estados onde ocorreram as vendas), estão apresentados os números correspondentes às vendas no estado *CA* (Califórnia), e, por sua vez, a geometria que representa este estado é apresentada no mapa da parte superior da tela.

As requisições do tipo GEOMD de integração, como já foi descrito na seção 5.3, são aquelas que possuem tanto parâmetros analíticos como geográficos. Normalmente, neste tipo de requisição, é realizada uma consulta analítica, semelhante à apresentada na figura 6.6, e como parâmetro geográfico é enviado algum operador espacial, como o apresentado na figura 6.10, para restringir o resultado da consulta.

Como exemplo de consulta GEOMD de integração, podemos citar a requisição MD apresentada na figura 6.16. Nesta, da mesma forma que a requisição apresentada na figura 6.6, serão listados os totais de vendas de produtos, classificando por categoria e por estado onde as vendas foram efetuadas. Como restrição geográfica, será aplicado o operador *BBOX* (ver seção 4.5.4.1). Dessa forma, a listagem dos totais de vendas dos produtos fica restrita aos estados que satisfizerem o operador espacial *BBOX* e a visualização gráfica do documento de resposta é apresentada na figura 6.17. Em uma requisição GEOMD de integração, o resultado final é obtido através da interseção das respostas do serviço analítico e do serviço geográfico. Dessa forma, só são enviados para a aplicação cliente, os dados que satisfizerem a consulta analítica, expressa no elemento *MD*, e a restrição geográfica presente no elemento *GEO* da requisição.

```

<GmlaRequest xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="D:\GMLARequest.xsd">
  <GEOMD>
    <GEOMDSchema>GOLAPAGMDM001</GEOMDSchema>
    <MD>
      <XMLAServiceAddress>http://localhost/xmla/msxisapi.dll</XMLAServiceAddress>
      <XMLAMethod>Execute</XMLAMethod>
      <OLAPCube>Sales</OLAPCube>
      <Command>
        <Statement>
          SELECT NON EMPTY(DRILLDOWNLEVEL([Store].[Store Country].members))
          ON COLUMNS,NON EMPTY([Product].[Product Family].members) ON ROWS
          FROM Sales
        </Statement>
      </Command>
      <Properties>
        <PropertyList>
          <DataSourceInfo>Provider=MSOLAP;Data Source=local </DataSourceInfo>
          <Catalog>Foodmart 2000</Catalog>
          <Format>Multidimensional</Format>
          <AxisFormat>TupleFormat</AxisFormat>
        </PropertyList>
      </Properties>
    </MD>
    <GEO>
      <WFSServiceAddress>http://localhost:8080/deegreewfs/deegreewfs</WFSServiceAddress>
      <GeoDB>UsaGeoDB</GeoDB>
      <GeoSpace>USAGeoSpace</GeoSpace>
      <GeoView>USAGeoView</GeoView>
      <GetFeature>
        <wfs:GetFeature outputFormat="GML2"
xmlns:gml="http://www.opengis.net/gml"
xmlns:wfs="http://www.opengis.net/wfs"
xmlns:ogc="http://www.opengis.net/ogc">
          <wfs:Query typeName="Country"> </wfs:Query>
          <wfs:Query typeName="States">
            <ogc:Filter>
              <ogc:BBOX>
                <wfs:PropertyName>/States/Border</wfs:PropertyName>
                <gml:Box>
                  <gml:coord>
                    <gml:X> -124.73276978</gml:X>
                    <gml:Y> 48.99993133</gml:Y>
                  </gml:coord>
                  <gml:coord>
                    <gml:X> -117.2084</gml:X>
                    <gml:Y> 43.0081</gml:Y>
                  </gml:coord>
                </gml:Box>
              </ogc:BBOX>
            </ogc:Filter>
          </wfs:Query>
        </wfs:GetFeature>
      </GetFeature>
    </GEO>
  </GEOMD>
</GmlaRequest>

```

Figura 6.16: Exemplo de requisição GEOMD de Integração.

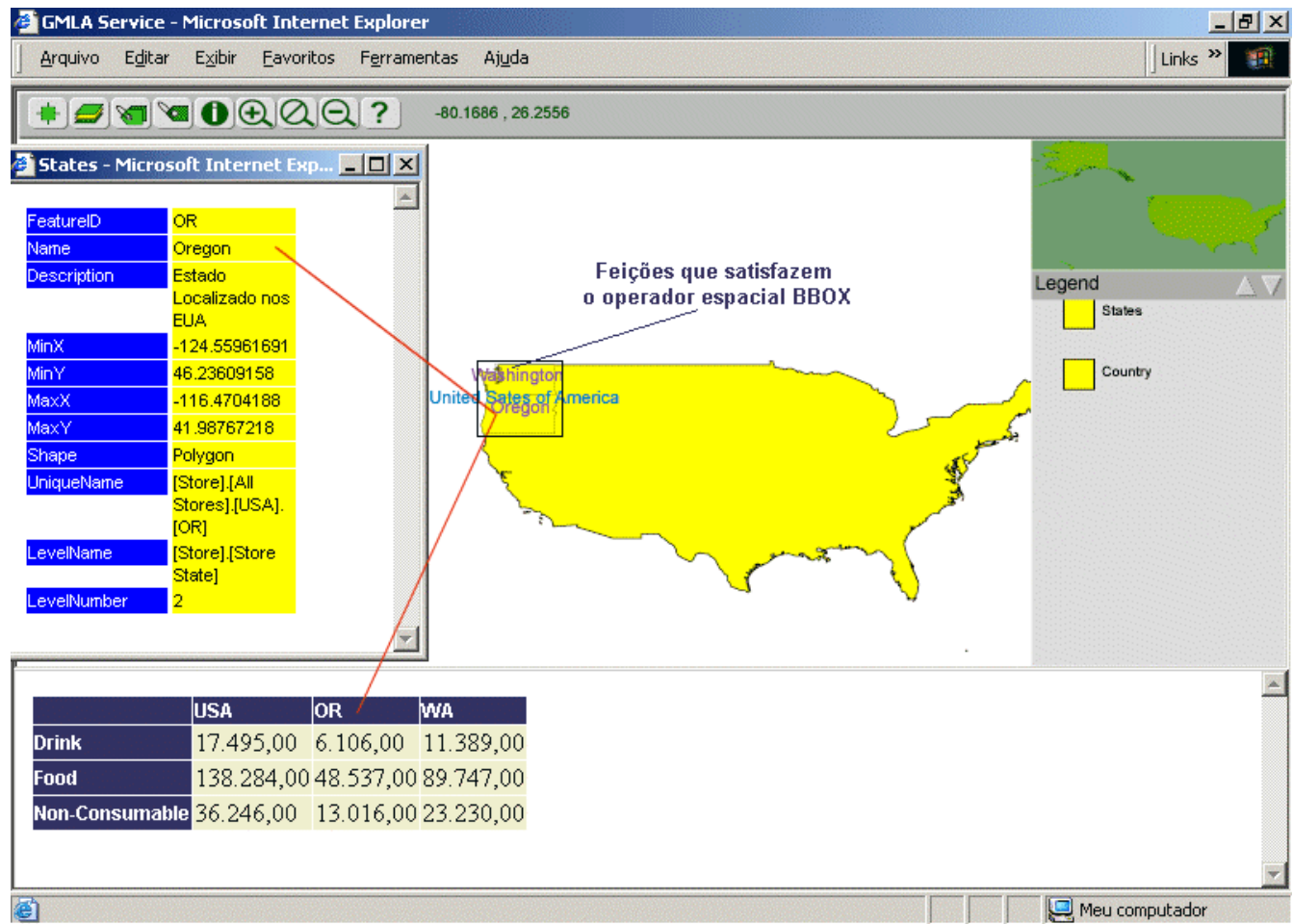


Figura 6.17: Visualização gráfica do resultado de uma consulta GEOMD de Integração.

Os documentos XML contendo o resultado das requisições apresentadas neste capítulo podem ser baixados da página do projeto GOLAPA. Estas instâncias se encontram no endereço <http://www.cin.ufpe.br/~golapa/gmlaws/responses>. Por sua vez, os documentos contendo as consultas aqui apresentadas e também outras requisições se encontram em <http://www.cin.ufpe.br/~golapa/gmlaws/requests>. Para baixar o *GMLA Request Schema* e o *GMLA Schema*, basta acessar o endereço <http://www.cin.ufpe.br/~golapa/schemas>.

6.9 Considerações Finais

Como pôde ser analisado nas seções que compõem este capítulo, o modelo de integração de serviços analíticos e geográficos, e demais idéias apresentadas no capítulo 5, puderam ser validadas. Visto que, foi disponibilizado um serviço capaz de integrar o processamento analítico e geográfico oferecido por dois serviços que antes trabalhavam de forma separada.

O serviço geográfico disponibiliza somente um operador espacial (*i.e.* *BBOX*). Este fato não prejudica, em nenhum momento, a definição do GMLA WS como serviço para processamento analítico e geográfico, uma vez que, para os demais operadores espaciais, a forma de tratamento das requisições e resultados obtidos seriam semelhantes. Ou seja, em todos os casos, a consulta sempre será expressa através de um documento XML e a resposta sempre será um conjunto de feições geográficas descritas em GML.

O fato de ser utilizado uma base de dados analíticos e geográficos que não contenha dados de aplicações reais, também não prejudica a validação das idéias apresentadas nesta dissertação, devido ao fato que a forma de processamento dos dados seria a mesma. A única vantagem adquirida pelo fato de se ter à disposição uma base de dados de maior porte, seria a possibilidade de realizar uma maior variedade de consultas envolvendo os tipos de requisições processadas pelo GMLA WS.

Neste capítulo, também pode-se observar como se deu a aplicação das tecnologias apresentadas no capítulo 4, no desenvolvimento do mecanismo de integração de serviços analíticos e geográficos. Isso esclarece a função de cada uma destas tecnologias dentro do contexto da presente dissertação.

Capítulo 7

Conclusões e Trabalhos Futuros

Este capítulo tem como objetivo apresentar algumas considerações finais sobre os principais tópicos abordados nesta dissertação, incluindo as contribuições alcançadas e indicações para trabalhos futuros.

7.1 Considerações Finais

O principal objetivo deste trabalho foi alcançado com a definição do modelo ISAG. Este modelo visa a integração de serviços analíticos e geográficos para suporte à decisão na *Web*, baseando-se em tecnologias abertas e extensíveis como XML, Java e *Web Services*.

O componente GOLAPE da referida arquitetura foi implementado com o modelo ISAG e a definição do serviço GMLA WS. Por sua vez, o componente GOLAPI, foi representado com o desenvolvimento de um protótipo, que possibilita a visualização gráfica dos resultados das requisições processadas pelo GMLA WS. Outro componente da arquitetura GOLAPA implementado foi o *Metadata*, através da utilização do repositório MDR, conforme descrito na seção 6.5.

Embora as soluções apresentadas nesta dissertação estejam voltadas para a arquitetura GOLAPA, vale salientar que são independentes da mesma. Por serem baseados em tecnologias abertas e extensíveis, tanto o modelo de integração como a maioria das implementações realizadas, podem ser aplicadas a outras pesquisas voltadas ao processamento analítico e geográfico para suporte à decisão. O modelo ISAG, apresentado no capítulo 5, pode ser aplicado em alguns dos trabalhos relacionados apresentados na seção 3.2 com a vantagem de utilizar somente tecnologias abertas e padrões emergentes. Por exemplo, o modelo ISAG poderia ser aplicado aos projetos *GOAL* e *SIGOLAP*.

7.2 Contribuições

Pode-se destacar como principais contribuições desta pesquisa:

- Apresentação do mapeamento das correspondências existentes entre as tecnologias para processamento analítico-multidimensional (*i.e. XML For Analysis*) e geográfico (*i.e. Web Feature Service*). O mapeamento leva em consideração aspectos como sintaxe de consulta a dados e metadados, forma de comunicação e formato para intercâmbio dos dados;
- Definição do *GMLA Request Schema* que integra a sintaxe de consulta dos serviços XMLA e WFS. Com a especificação deste esquema, foram definidos os tipos de consultas que podem ser enviadas ao serviço GMLA WS. Com isso, possibilitou-se a utilização de operadores analíticos e espaciais em um único documento XML, contendo a requisição que será processada pelo GMLA WS;
- Especificação do modelo ISAG, o qual visa a integração de serviços analíticos e geográficos na *Web*. Com este modelo, é possível agrupar em um único ambiente, o poder de processamento analítico-multidimensional, disponibilizado pelas implementações da especificação *XML For Analysis* e as habilidades oferecidas pelos serviços que implementam a especificação *Web Feature Service*, para manipulação de feições geográficas através da aplicação de operações espaciais;
- Validação do modelo ISAG através da implementação do serviço GMLA WS. Este serviço é o mecanismo de integração que possibilita o processamento de requisições contendo operadores analíticos e espaciais. O processamento destas requisições implica no acesso à base de dados analíticos e geográficos, com posterior tratamento do resultado da consulta, para enviar para a aplicação cliente responsável pela solicitação;

- Desenvolvimento de um protótipo da interface gráfica para demonstração gráfica dos dados analíticos em forma de tabelas, e dos dados geográficos em forma de mapas.

7.3 Trabalhos Futuros

Esta dissertação apresentou um modelo de integração de serviços analíticos e geográficos (ISAG), o qual foi validado pela implementação de um *Web Service* que disponibiliza operações para consultas a uma base de dados analíticos e geográficos, proporcionando um suporte considerável para o processo de tomada de decisões estratégicas. Embora os objetivos tenham sido alcançados, ainda existem temas de pesquisas ligados a esta dissertação que podem ser explorados:

- Definição de uma interface gráfica mais elaborada, uma vez que a nossa interface tem como objetivo apenas validar os resultados produzidos pelo GMLA WS. A interface gráfica a ser construída deveria permitir uma maior interação com o usuário, possibilitando que, por exemplo, consultas fossem realizadas a partir da manipulação dos objetos gráficos apresentados na tela;
- Para otimização das consultas ao GMLA WS, seria interessante analisar a possibilidade de armazenar, de forma pré-processada, os mapas com maior frequência de utilização, o que pode diminuir a carga das consultas ao GMLA WS;
- Reestruturação do *GMLA Schema* (ver seção 3.3.1.2) e do GMLA WS (ver seção 6.2) para suportar a versão mais recente de GML. Pois, como descrito na seção 4.2.4, foi utilizada a versão mais estável de GML (*i.e.* versão 2.1.2);
- Especificação de uma linguagem de consulta multidimensional-geográfica, a qual permita uma maior integração no envio de requisições ao GMLA WS. Atualmente, a sintaxe de consulta está baseada no *GMLA Request Schema* (ver seção 5.3), a qual herda as sintaxes definidas para consultas ao XMLA e ao WFS. Entretanto, os parâmetros analíticos e geográficos ainda continuam

separados, e a definição de uma sintaxe única, que possibilite a definição de requisições contendo tanto operadores analíticos como espaciais, seria interessante.

- Complementar a implementação do GMLA WS a fim de que ele suporte a realização de consultas que envolvam mais de duas dimensões geográficas.
- Analisar a possibilidade do tratamento de dados raster, visto que neste trabalho somente foram considerados dados vetoriais pelo fato da versão GML utilizada só oferecer suporte para este tipo de dado.
- Realizar um estudo de viabilidade da implementação de outros operadores espaciais no serviço geográfico utilizado (*i.e. Degree Web Feature Service*), já que este é de domínio público e pode ser alterado livremente. Dessa forma, poderiam ser apresentados os resultados do GMLA WS com outras operações espaciais além do *BBOX*.
- Analisar a possibilidade de realizar alterações no modelo de integração de serviços e no GMLA WS para permitir o processamento de requisições do tipo GEOMD de integração que consulte primeiro o serviço geográfico. Isto possivelmente melhoraria o desempenho do sistema e possibilitaria a realização de operações que não foram previstas pela versão atual do protótipo (*e.g. junções espaciais*).

Referências Bibliográficas

- [AI80] Steven L. Alter, *Decision Support Systems : Current Practice and Continuing Challenges*, Addison-Wesley, 1980.
- [AMG03] *AutoDesk MapguideServer*
<http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=331041>
(último acesso em dezembro de 2003)
- [AF03] Francisco José de A. Alves e Elizabeth S. Furtado, *Sistema de Educação a Distância Baseada em Serviços Web*
<http://www.abed.org.br/congresso2002/trabalhos/texto46.htm> (último acesso em dezembro de 2003)
- [BM03] Ilse M. Beuren e Luciano W. Martins, *Sistema de Informações Executivas: Suas Características e Reflexões sobre sua Aplicação no Processo de Gestão*
http://www.eac.fea.usp.br/cadernos/completos/cad26/Revista_26_part_1.pdf
(último acesso em novembro de 2003)
- [Boh04] Juan Euclides Bohorquez - Oracle DBA, *Aproximación Metodológica de un Spatial Data Warehouse*
http://gis.esri.com/library/userconf/latinproc00/colombia/spatial_data.pdf
(último acesso em março de 2004)
- [BR03] S. Brobst, J. Rarey, *Five Stages of Data Warehouse Decision Support Evolution*,
<http://dssresources.com/subscriber/password/papers/features/brobst&rarey01062003.html>, 2003, (último acesso em dezembro de 2003)

- [BRJ99] G. Booch, J. Rumbaugh, and I. Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley, 1999.
- [BRM01] Y. Bédard, S. Rivest, P. Marchand, *Toward Better Support for Spatial Decision Making: Defining the Characteristics of SOLAP*, Geomatica Vpl. 55, N° 4, 2001.
- [BSG00] T. Barclay, R. D. Slutz, J. Gray, *TerraServer: A Spatial Data Warehouse*. Proceedings of the 2000 ACM-SIGMOD Conference, 2000.
- [CCH+96] G. Câmara, M. A. Casanova, A. S. Hemerly, G. C. Magalhães, C. M. B. Medeiros. *Anatomia de Sistemas de Informação Geográfica*. 10ª Escola de Computação, 1996. <http://www.dpi.inpe.br/geopro/livros/anatomia.pdf> (último acesso em dezembro de 2003)
- [CD96] S. Chaudhuri, U. Dayal. *Decision support, Data Warehouse, and OLAP*, Tutorials of the Twenty-Second international Conf. On Very Large Data Base, Bombay, 1996.
- [CF03] M. L. Campos, A. V. R. Filho. *Data warehouse Tutorial*, <http://genesis.nce.ufrj.br/dataaware/tutorial/tutorial.html> (último acesso em novembro de 2003)
- [Chr96] N. Chrisman. *Exploring Geographic Information Systems*. John Wiley and Sons, 1996.
- [Cla97] Keith C. Clarke. *Getting Started with Geographic Information Systems*, Prentice Hall, USA, 1997.
- [CSS03] *OpenGIS® Catalog Services Specification*, <http://test.opengis.org/docs/02-087r3.pdf> (último acesso em novembro de 2003)
- [CT97] Luca Cabibbo and Riccardo Torlone, *Querying Multidimensional Databases*, Sixth Int. Workshop on Database Programming Languages, 1997.

- [Dat89] C. J. Date, *A Guide to the SQL Standard (Second Edition)*. Addison-Wesley, 1989.
- [DEL03] *Borland Delphi Official Home Page*, <http://www.borland.com/> (último acesso em novembro de 2003)
- [Dem97] Michael N. Demers. *Fundamentals of Geographic Information Systems*, Wiley, USA, 1997.
- [DMOF03] *dMOF 1.1, An OMG Meta Object Facility Implementation* <http://www.dstc.edu.au/Products/CORBA/MOF/> (último acesso em dezembro de 2003)
- [DTD98] W3C Consortium. *Data Modeling Report Prepared for: W3C XML Specification DTD (“XMLspec”)*, 1998. <http://www.w3.org/XML/1998/06/xmlspec-report-19980910.htm#AEN27> (último acesso em dezembro de 2003)
- [DWFS03] *Deegree Web Feature Service*, <http://deegree.sourceforge.net/> (ultimo acesso em novembro de 2003)
- [EN99] R.Elmasri e S. B. Navathe. *Fundamentals of Database Systems*. Addison-Wesley, 3rd edition, 1999.
- [ESRI03] *ArcView Official Home Page* <http://www.esri.com/> (último acesso em dezembro de 2003)
- [FCT01] A.C. Ferreira, M. L. Campos, A. Tanaka. *An Architecture for Spatial and Dimensional Analysis Integration*. Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics (SCI 2001), volume XIV computer Science and Engineering, 2001.
- [FEIS03] *OpenGIS[®] Filter Encoding Implementation Specification*, <http://www.opengis.org/techno/specs/02-059.pdf> (último acesso em novembro de 2003)

- [Fer02] A.C. Ferreira. *Um Modelo para Suporte à Integração de Análises Multidimensionais e Espaciais*. Dissertação de Mestrado - UFRJ, 2002.
- [FH01] Lixin Fu and Joachim Hammer, *CUBIST: A New Approach to Speeding Up OLAP Queries in Data Cubes*, University of Florida, Gainesville, FL, Technical Report TR01-007, 2001.
- [FI96] J. L. Filho e C. Iochpe. *Introdução a Sistemas de Informação Geográfica com Ênfase em Banco de Dados*. XV Jornada de Atualização em Informática - XVI Congresso da SBC, 1996.
<http://www.dpi.ufv.br/~jugurta/publica.html> (último acesso em dezembro de 2003)
- [Fid03] Robson do Nascimento Fidalgo. *Integração de Processamento Analítico com Sistemas de Informações Geográficas para Suporte à Decisão em Ambientes de Data Warehouse Geográfico* – Proposta de Tese de Doutorado, Cin/UFPE, 2003.
- [FME03] *Feature Manipulation Engine*, <http://www.safe.com/products/fme/> (último acesso em novembro de 2003)
- [FPR+00] F. Ferri, E. Pourabbas, M. Rafanelli, F.L. Ricci, *Extending Geographic Databases for a Query Language to Support Queries Involving Statistical Data*, Proceedings of the 12th International Conference on Scientific and Statistical Database Management (SSDBM'00), 2000.
- [FST+03a] R. N. Fidalgo, J. Silva, V. C. Times, F. F. Souza, R. S. M. Barros. *GMLA: A XML Schema for Integration and Exchange of Multidimensional-Geographical Data*. Simpósio Brasileiro de Geoinformática - GEOINFO 2003, <http://www.geoinfo.info/Anais/geoinfo2003-48.pdf> (último acesso em dezembro de 2003)
- [FST+03b] R. N. Fidalgo, J. Silva, V. C. Times, F. F. Souza. *GeoDWFrame: A Framework to Guide the Design of Geographical Dimensional Schemas*. Artigo Submetido para a 6th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2004)

- [FTS01] R. N. Fidalgo, V. C. Times, F. F. Souza. ***GOLAPA: Uma Arquitetura Aberta e Extensível para Integração entre SIG e OLAP.*** Simpósio Brasileiro de Geoinformática - GEOINFO 2001, <http://www.ic.unicamp.br/~cmbm/geoinfo/papers/149robson.pdf>, (último acesso em dezembro de 2003)
- [GBL+96] J. Gray, A. Bosworth, A. Layman, and H. Pirahesh. ***Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals.*** Proceedings of the Twelfth International Conference on Data Engineering - 12th ICDE, 1996.
- [GHP03] ***GOLAPA Home Page***, <http://www.cin.ufpe.br/~golapa/> (último acesso em novembro de 2003)
- [GLUE03] ***GLUE, Official Home Page***, <http://www.theminelectric.com/glue/> (último acesso em novembro de 2003)
- [GML03a] ***OpenGIS[®] Geography Markup Language (GML) 2.0 Implementation Specification*** <http://www.opengis.org/techno/documents/02-023r4.pdf> (último acesso em dezembro de 2003)
- [GML03b] OGC, ***Geography Markup Language (GML) Implementation Specification, version 3.0.0,*** <http://www.opengis.org/techno/implementation.htm> (último acesso em dezembro de 2003)
- [GNU03] ***GNU General Public License***, <http://www.gnu.org/licenses/gpl.html> (último acesso em novembro de 2003)
- [Goa03] ***Home Page Oficial do Projeto GOAL***, <http://krizik.felk.cvut.cz/goal/> (último acesso em dezembro de 2003)
- [Gon99] M. L. Gonzales. ***Spatial OLAP: Conquering Geography.*** DB2 Magazine White paper, 1999, http://www.db2mag.com/db_area/archives/1999/q1/99sp_gonz.shtml (último acesso em Janeiro de 2003).

- [GWFS03] *The GeoServer Project Official Home Page*, <http://geoserver.sourceforge.net/html/index.php> (último acesso em novembro de 2003)
- [Har03] Leopoldo E.Q.Hartard, *Sistemas de Inteligência de Negócios*, <http://www.sergiomaturana.com/SAG/CLASE03LQ.pdf> (último acesso em novembro de 2003)
- [HK00] J. Han, M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann, 2000.
- [HKS97] J. Han, K. Koperski, and N. Stefanovic. *GeoMiner: A System Prototype for Spatial Data Mining*. Proceedings of the 1997 ACM-SIGMOD Conference, 1997.
- [HKS00] J. Han, K. Koperski, N. Stefanovic, *Object-Based Selective Materialization for Efficient Implementation of Spatial Data Cubes*, IEEE Transactions on Knowledge and Data Engineering, 2000.
- [Hol01] HOLZER. Steven, *Desvendando XML*, Ed. Campus, RJ, 2001.
- [HSK98] J. Han, N. Stefanovic, K. Koperski. *Selective materialization: An Efficient Method for Spatial Data Cube Construction*. Proceedings of the 1998 Pacific-Asia Conf. Knowledge Discovery and Data Mining (PAKDD'98), 1998.
- [HTTP03] *Hypertext Transfer Protocol*, <http://www.w3.org/Protocols/> (último acesso em dezembro de 2003)
- [IBM03] *IBM Official Home Page*, <http://www.ibm.com/> (último acesso em novembro de 2003)
- [Inm97] W. H. Inmon. *Como construir um Data Warehouse*. Editora Campus, 1997.
- [JAV03] *Java Official Home Page*, <http://java.sun.com>, (último acesso em Março de 2004)

- [JDBC03] *JDBC technology*, <http://java.sun.com/products/jdbc/> (último acesso em novembro de 2003)
- [JMI02] Sun Corp. *Java Metadata Interface (JMI) Specification Api 1.0*, 2002, <http://java.sun.com/products/jmi/download.html> (último acesso em dezembro de 2003)
- [Kam98] V. Kamp, *Database System Support for Spatio-Temporal Aspects in Scientific Applications*, International Database Engineering and Applications Symposium (IDEAS), 1998.
- [Kim96] R. Kimball. *The Data warehouse Lifecycle Toolkit*. 1996, John Wiley & Sons, Inc.
- [Kim01] R. Kimball. *Spatial Enabling Your Data Warehouse*. Intelligent Enterprise White Paper. 2001, <http://www.intelligententerprise.com/010101/webhouse.shtml>. (último acesso em agosto de 2003)
- [KMM00] Z. Kouba, K. Matousek, P. Miksovsky. *On Data Warehouse and GIS integration*. Proceedings of the 11th International Conference on Database and Expert Systems Applications (DEXA), 2000.
- [KMM01] Z. Kouba, P. Mikšovský, K. Matousek. *On Geographical On-Line Analytical Processing (GOLAP)*. Proc. of World Multiconference on Systemics, Cybernetics and Informatics (ISAS-SCIs), 2001.
- [MA03] *Microsoft Access Official Home Page* <http://www.microsoft.com/office/access/default.asp> (último acesso em novembro de 2003)
- [Mal00] Efrem G. Mallach, *Decision Support and Data Warehouse systems*, McGraw-hill, 2000.
- [MDX03a] *Introduction to Multidimensional Expressions (MDX)* <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnolap/html/intromdx.asp> (último acesso em novembro de 2003).

- [MDX03b] *Introduction to Multidimensional Expressions (MDX) Comparison of SQL and MDX*. Analysis Service Documentation, Microsoft Corporation, 2000.
- [MDXML03] *Access to Intelligence: The New OLAP APIs* - http://www.intelligententerprise.com/cgi-bin/printable.cgi?file=../020917/515feat2_1.shtml (último acesso em dezembro de 2003)
- [Me193] J. Melton, A.R. Simon, *Understanding the New SQL: A Complete Guide*, Morgan Kaufmann, 1993.
- [Men02] A. Mendes. *Arquitetura de Software: Desenvolvimento Orientado para a Arquitetura*. Ed. Campus, 2002.
- [MGR91] David J. Maguire, Michael F. Goodchild, David W. Rhind., *Geographical Information Systems Principles and Applications*, Volume 1, Longman, 1991.
- [MIE03] *Microsoft Internet Explorer Official Home Page*, <http://www.microsoft.com/windows/ie/default.asp> (Último acesso em dezembro de 2003).
- [MI03] *MapInfo Professional*, <http://www.mapinfo.com/> (Último acesso em dezembro de 2003).
- [MOF02] *OMG - Object Management Group. MetaObject Facility (MOF) Specification 1.4*, 2002.
- [MS00] K. Matousek, L. Svoboda. *Extending GIS by Data Warehouse*, Proceedings of International Carpathian Control Conference, 2000.
- [MSQL03] *Microsoft SQL Server Official Home Page*, <http://www.microsoft.com/sql/> (último acesso em dezembro de 2003)
- [MSXA03] *Microsoft XML For Analysis Provider*, <http://www.microsoft.com/downloads/details.aspx?FamilyId=0E93B1EE->

- [BE73-4D15-9C37-6CC2DCC2AC33&displaylang=en](http://www.microsoft.com/downloads/details.aspx?familyid=3144B72B-B4F2-46DA-B4B6-C5D7485F2B42&displaylang=en) (último acesso em novembro de 2003)
- [MSXML03] **MSXML 4.0 Service Pack 2 (Microsoft XML Core Services)**
<http://www.microsoft.com/downloads/details.aspx?familyid=3144B72B-B4F2-46DA-B4B6-C5D7485F2B42&displaylang=en> (último acesso em dezembro de 2003)
- [Mur95] Lisa D. Murphy, **Geographic Information Systems: Are They Decision Support Systems?**, Proceedings of the 28th Annual Hawaii International Conference on System Sciences, 1995.
- [NBe03] **NetBeans Official Home Page**, <http://www.netbeans.org> (último acesso em novembro de 2003)
- [NFC01] Marcos C. Neves, Corina C. Freitas E Gilberto Câmara, **Mineração de Dados em Grandes Bancos de Dados Geográficos**, Instituto Nacional de Pesquisas Espaciais (INPE), Relatório Técnico, 2001.
- [NNa03] **Netscape Navigator Official Home Page**, <http://www.netscape.com/> (Último acesso em novembro de 2003)
- [ODBC03] **Microsoft ODBC Official Home Page**,
<http://www.microsoft.com/data/odbc/default.htm> (último acesso em novembro de 2003)
- [OGC01] OpenGIS Consortium. **The OpenGIS Abstract Specification Topic 1: Feature Geometry (ISO 19107 Spatial Schema)**, Version 5, 2001.
- [OGC03] **Open GIS Consortium Inc.**, <http://www.opengis.org/> (último acesso em novembro de 2003)
- [OGC99] OpenGIS Consortium. **The Abstract OpenGIS Specification**. OpenGIS Document Number 99-100. Jun., 1999.
<http://www.opengis.org/techno/specs.htm> (Último acesso em dezembro de 2003)

- [OGC99a] OpenGIS Consortium. *The OpenGIS Abstract Specification - Topic 0: Abstract Specification Overview*, Version 4, 1999.
- [OGC99b] OpenGIS Consortium. *The OpenGIS Abstract Specification Topic 10: Feature Collections*, Version 4, 1999
- [OGC99c] OpenGIS Consortium. *The OpenGIS Abstract Specification Topic 6: The Coverage Type and its Subtypes*, Version 6, 1999.
- [OGC99d] OpenGIS Consortium. *The OpenGIS Abstract Specification Topic 5: Features*, Version 4, 1999.
- [OLE03] *Microsoft OLE DB*, <http://www.microsoft.com/data/oledb/default.htm> (último acesso em dezembro de 2003)
- [OMG03] *Object Management Group Home Page*, www.omg.org (último acesso em dezembro de 2003)
- [PKZ+01] D. Papadias, P. Kalnis, J. Zhang, Y. Tao, *Efficient OLAP Operations in Spatial Data Warehouses*, Proceedings of the 7th International Symposium on Spatial and Temporal Databases, (SSTD), 2001.
- [Pow98] Daniel J. Power, *What is a Decision Support Systems?*, <http://dssresources.com/papers/whatisadss/index.html>, 1998, (último acesso em Setembro de 2003)
- [Pow00] Daniel J. Power, *Supporting Decision-Makers: An Expanded Framework*, <http://dssresources.com/papers/supportingdm/sld001.htm>, 2000, (último acesso em dezembro de 2003)
- [Pow02] Daniel J. Power, *Decision Support Systems Web Tour*, <http://dssresources.com/tour/index.html>, 2002, (último acesso em dezembro de 2003)
- [PR02] E. Pourabbas, M. Rafanelli, *A Pictorial Query Language for Querying Geographic Databases Using Positional and OLAP Operators*, SIGMOD Record, Vol. 31, N° 2, June 2002.

- [PT01] T. B. Pedersen, N. Tryfona, *Pre-aggregation in Spatial Data Warehouses*, Proceedings of the Seventh International Symposium on Spatial and Temporal Databases, 2001.
- [PTK+02] D. Papadias, Y. Tao, P. Kalnis, J. Zhang, *Indexing Spatio-Temporal Data Warehouses*, Proceedings of the IEEE International Conference on Data Engineering (ICDE), 2002.
- [PZ04] S. Prasher, X. Zhou, *Multiresolution Amalgamation: Dynamic Spatial Data Cube Generation*, Proceedings of the Fifteenth Australian Database Conference (ADC2004), 2004.
- [Rao96] Sudir G. Rao, *Providing Better Support for a Class of Decision Support Queries*. ACM-SIGMOD Int'l Conf. on Management of Data, 1996.
- [RG93] S. A. Roberts and M. N. Gahegean. *An Object –Oriented Geographic Information System Shell*, Information and Software Technology, 35(10), 1993.
- [RGH+91] S. A. Roberts, M.N. Gahegean, J.Hogg, B. Boyle. *Application of Object-Oriented Databases to Geographic Information Systems*. Information and Software Technology, 33(1), 1991.
- [Ril03] Fábio Rilston Silva Paim, *Uma Metodologia para Análise de Requisitos em Sistemas Data Warehouse* - Dissertação de Mestrado, Cin/UFPE, 2003.
- [SC98] E.C.S. Silva e M.L.M. Campos. *Integração de Sistemas de Informação Geográficas e Ferramentas OLAP*, Congresso para Usuários de Geoprocessamento da América Latina (GISBrasil98), 1998.
- [SCHE03] *World Wide Web Consortium, XML Schema*
<http://www.w3.org/XML/Schema> (último acesso em dezembro de 2003)
- [SFS99] *OpenGIS® Simple Features Specification for SQL*, Open GIS Consortium Inc. Maio de 1999, Revision 1.1.

- [STB03a] J. Silva, V. C. Times, R. S. M. Barros. *Sistematizando o Ciclo de Vida da Integração de Serviços na Web*. 2nd I2TS'2003 - International Information and Telecommunication Technologies Symposium, 2003.
- [STB03b] J. Silva, V. C. Times, R. S. M. Barros. *Sistematização da Integração de Serviços na Web*. IDEAS'2004 - 7º Workshop Iberoamericano de Engenharia de Requisitos e Ambientes de Software, 2004.
- [STF+03a] J. Silva, V. C. Times, R. N. Fidalgo, R. S. M. Barros. *Towards a Web Service for Geographic-Multidimensional Processing*. Artigo Submetido para a ICWS'2004 - IEEE International Conference on Web Services.
- [STF+03b] J. Silva, V. C. Times, R. N. Fidalgo, R. S. M. Barros. *GMLA WS: A Multidimensional-Geographic Web Service*. Artigo Submetido para a ECOWS'04 - The European Conference on Web Services (ECOWS'04).
- [SGML03] *Standard Generalized Markup Language (SGML)*, <http://xml.coverpages.org/sgml.html> (Último acesso em dezembro de 2003)
- [SLT+00a] S. Shekhar, C.T. Lu, X. Tan, S. Chawla, *Map Cube: A Visualization Tool for Spatial Data Warehouses* - Proceedings of NSF workshop on Data Mining in GIS, 1999.
- [SLT+00b] S. Shekhar, C.T. Lu, X. Tan, S. Chawla, *Map Cube: A Visualization Tool for Spatial Data Warehouses*, as Chapter of Geographic Data Mining and Knowledge Discovery. Harvey J. Miller and Jiawei Han (eds.), Taylor and Francis, 2001, ISBN 0-415-23369-0. <http://www.cs.umn.edu/Research/shashi-group/Mapcube/mapcube.html> (último acesso em dezembro de 2003)
- [SM03] Sugumaran, R. and J. Meyer, *Building a Web-Based Spatial Decision Support System (WEBSDSS) for Environmental Planning and Management*, 2003, <http://dssresources.com/>, (último acesso em novembro de 2003)

- [SOAP03] World Wide Web Consortium, **SOAP**, <http://www.w3c.org/TR/soap12-part0/>, <http://www.w3c.org/TR/soap12-part1/>, <http://www.w3c.org/TR/soap12-part2/> (último acesso em novembro de 2003)
- [Spr80] R. H. Sprague. *A Framework for the Development of Decision Support Systems*. Management Information Systems Quarterly, vol. 4, n° 2, 1980.
- [SQL92] *Structured Query Language. Database Language SQL*, ISO/IEC 9075:1992
- [Ste97] N. Stefanovic. *Design and Implementation of On-line Analytical Processing (OLAP) of Spatial Data*. M.Sc. Thesis, Simon Fraser University, Canada, 1997.
<http://citeseer.nj.nec.com/stefanovic97design.html> (último acesso em dezembro de 2003)
- [SVG03] *Scalable Vector Graphics (SVG)*. <http://www.w3.org/Graphics/SVG/>, W3C Recommendation, (Último acesso em dezembro de 2003)
- [SVG03] *Adobe SVGviewer*, <http://www.adobe.com/svg> (último acesso em Setembro de 2003)
- [Tho97] E. Thomsen. *OLAP Solutions – Building Multidimensional Information Systems*. John Wiley, 1997.
- [UDDI03] *Universal Description, Discovery and Integration (UDDI)*
<http://www.uddi.org/> (último acesso em dezembro de 2003)
- [VB03] *Visual Basic Official Home Page*, <http://msdn.microsoft.com/vbasic/> (último acesso em novembro de 2003)
- [VC03] *Visual C++ Official Home Page*, <http://msdn.microsoft.com/visualc/> (último acesso em novembro de 2003)

- [VML98] **W3C Vector Markup Language (VML)**,
<http://www.w3.org/TR/1998/NOTE-VML-19980513> (último acesso em novembro de 2003)
- [W3C03] *World Wide Web Consortium (W3C)* <http://www.w3.org> (último acesso em novembro de 2003)
- [WFS02] **OpenGIS[®] Web Feature Service Implementation Specification**, Open GIS Consortium Inc. 19-September-2002, Version: 1.0.0.
- [WMS03] **OpenGIS[®] Web Map Service Implementation Specification**
<http://www.opengis.org/techno/specs/01-068r3.pdf> (último acesso em novembro de 2003)
- [Wor95] M. Worboys. **GIS: A computing Perspective**. Taylor & Francis, 1995.
- [WSA02] **Web Services Architecture** <http://www.w3.org/TR/2002/WD-ws-arch-20021114/> (último acesso em dezembro de 2003)
- [WSCA01] **Web Services Conceptual Architecture (WSCA 1.0)**, By Heather Kreger IBM Software Group, 2001.
<http://www.ibm.com/software/solutions/webservices/pdf/WSCA.pdf> (último acesso em dezembro de 2003)
- [WSDL01] *World Wide Web Consortium. Web Services Description Language (WSDL)*, <http://www.w3.org/TR/2001/NOTE-wsdl-20010315> (último acesso em novembro de 2003).
- [WSDP03] **Java[™] Web Services Developer Pack 1.2**,
<http://java.sun.com/webservices/downloads/webservicespack.html> ,(último acesso em novembro de 2003)
- [XC03] **Xalan-C++**, <http://xml.apache.org/xalan-c/> (último acesso em outubro de 2003)
- [XJ03] **Xalan-Java**, <http://xml.apache.org/xalan-j/> (último acesso em outubro de 2003).

- [XLI03] **XML Linking Language (XLink) Version 1.0**, <http://www.w3.org/TR/xlink/>, W3C Recommendation , (Último acesso em novembro de 2003)
- [XMI02] **OMG Org. XML Metadata Interchange (XMI) Specification Version 1.2**. Especificação, 2002. <http://www.omg.org/technology/documents/formal/xmi.htm>.(último acesso em novembro de 2003)
- [XML00] **W3C Extensible Markup Language (XML) 1.0 (Second Edition)**. W3C Recommendation, 2000. <http://www.w3.org/TR/REC-xml>, (último acesso em novembro de 2003)
- [XML03a] **Extensible Markup Language (XML) Official Home Page**. <http://www.w3.org/XML/> (último acesso em dezembro de 2003)
- [XML03b] **XML.ORG Home Page**, <http://www.xml.org/> (último acesso em dezembro de 2003)
- [XMLA02] **XML For Analysis Specification**, Version 1.1, Microsoft Corporation & Hyperion Solutions Corporation Updated: 11/20/2002.
- [XMLA03] **XML For Analysis Home Page**, <http://www.xmla.org> (último acesso em dezembro de 2003)
- [XNa03] **XML Namespaces** - <http://www.w3.org/TR/1999/REC-xml-names-19990114/> - (último acesso em dezembro de 2003)
- [XPA03] **XML Path Language (XPath) Version 2.0**, <http://www.w3.org/TR/xpath20/>, (último acesso em novembro de 2003)
- [XPO03] **XML Pointer Language (XPointer)**, <http://www.w3.org/TR/xptr/> (último acesso em novembro de 2003).
- [XSL03] **W3C, Extensible Stylesheet Language (XSL) Version 1.0**, <http://www.w3.org/TR/xsl/> (ultimo acesso em novembro de 2003)

- [XSLFO03] *Using XSL FO with XEP 3.0*,
<http://www.renderx.com/Tests/doc/html/tutorial.html> (último acesso em novembro de 2003)
- [XSLT03] *World Wide Web Consortium, XSL Transformations (XSLT)*,
<http://www.w3.org/TR/xslt> (último acesso em novembro de 2003)
- [XSPY03] *XML Spy Editor*, <http://www.xmlspy.com> (último acesso em novembro de 2003)
- [XT03] *XT - An implementation of XSLT in Java*,
<http://www.blz.com/xt/index.html> (último acesso em outubro de 2003)
- [Ylu00] Damires Yluska de Souza Fernandes, *GeoVisual Interface – Uma Interface para Consultas Visuais em Bancos de Dados Geográficos*, Dissertação de Mestrado, CIn-UFPE, 2000.
- [ZFG01] H. B. Zghal, S. Faiz, H. B. Ghézala, *Exploration Techniques of the Spatial Data Warehouses: Overview and Application to Incendiary Domain*, Proceedings of the ACS/IEEE International Conference on Computer Systems and Applications (AICCSA'01), 2001.
- [ZLR+03a] L. Zhang, Y. Li, F. Rao, X. Yu, Y. Chen, D. Liu, *An Approach to Enabling Spatial OLAP by Aggregating on Spatial Hierarchy*, International Conference on Data Warehousing and Knowledge Discovery (DaWaK), 2003.
- [ZLR+03b] L. Zhang, Y. Li, F. Rao, X. Yu, Y. Chen, *Spatial Hierarchy and OLAP-Favored Search in Spatial Data Warehouse*, ACM Sixth International Workshop on Data Warehousing and OLAP (DOLAP 2003), 2003.

Dissertação de Mestrado apresentada por **Joel da Silva** à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título , “**Integrando Serviços Analíticos e Geográficos para Suporte à Decisão na Web**”, orientada pela **Profa. Valéria Cesário Times** e aprovada pela Banca Examinadora formada pelos professores:



Profa. Ana Carolina Brandão Salgado
Centro de Informática / UFPE



Prof. Claudio de Souza Baptista
Departamento Sistemas e Computação / UFCG



Profa. Valeria Cesário Times
Centro de Informática / UFPE

Visto e permitida a impressão.
Recife, 15 de março de 2004.



Prof. JAELSON FREIRE BRELAZ DE CASTRO
Coordenador da Pós-Graduação em Ciência da Computação do
Centro de Informática da Universidade Federal de Pernambuco.