

# UERJ

Dissertação de Mestrado em Engenharia de Computação  
Área de Concentração Geomática

**ESTUDO DOS PROBLEMAS DE INTEGRAÇÃO SEMÂNTICA  
ENTRE A INFORMAÇÃO GEOGRÁFICA DISTRIBUÍDA E A  
INFORMAÇÃO GEOGRÁFICA LOCAL**

Autor: Ricardo José Leal dos Santos

Orientadora: Prof<sup>a</sup> Margareth Simões Penello Meirelles

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

Março – 2006



Faculdade de Engenharia

# **ESTUDO DOS PROBLEMAS DE INTEGRAÇÃO SEMÂNTICA ENTRE A INFORMAÇÃO GEOGRÁFICA DISTRIBUÍDA E A INFORMAÇÃO GEOGRÁFICA LOCAL**

**Ricardo José Leal dos Santos**

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

Orientador: Professora Margareth Simões Penello Meirelles, D. Sc.

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

Rio de Janeiro

Março – 2006

## FOLHA DE JULGAMENTO

Título: Estudo dos Problemas de Integração Semântica entre a Informação Geográfica Distribuída e Informação Geográfica Local

Candidato: Ricardo José Leal dos Santos

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

Data da defesa: 21/03/2006

Aprovada por:

---

Orientadora: Professora Margareth Simões Penello Meirelles, D. Sc., UERJ.

---

Carlo Emmanoel Tolla de Oliveira, Ph. D., UFRJ

---

Oscar Luiz Monteiro de Farias, D. Sc., UERJ

## DEDICATÓRIA

Aos meus filhos,  
Gabriella e João Vítor  
À minha paixão  
Evelyn,  
À minha mãe,  
Maria Stela,  
Ao amigo  
Paulo Maurício.

## AGRADECIMENTOS

Agradeço principalmente e em primeiro lugar, a Deus, pois sem Ele nada existiria e tudo seria em vão.

À Evelyn, minha namorada, por sua dedicação a mim nos momentos finais deste trabalho cujo incentivo impediu que eu desistisse.

À minha mãe Stela e a meu irmão Sérgio, por estarem sempre ao meu lado quando mais precisava de apoio.

Aos meus filhos Gabriella e João Vitor, pela compreensão nos momentos em que tive de me afastar para dedicar-me a este trabalho.

À minha orientadora Margareth, por manter a paciência quando os meus problemas pessoais pareciam infundáveis.

Ao amigo Paulo Maurício, pelas horas de orientações na Embrapa, apoio e ensinamentos reconhecidamente de um mestre.

A todos que, por algum motivo, não responderam meus *e-mails*, telefonemas e pedidos, porque me deram forças para buscar, sozinho, respostas às minhas dúvidas.

A todos os professores que direta ou indiretamente transmitiram-me seus conhecimentos e experiências.

Aos funcionários administrativos Beth, Rinete, Carlos etc. por trabalharem sempre com boa vontade e amizade.

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

## **ESTUDO DOS PROBLEMAS DE INTEGRAÇÃO SEMÂNTICA ENTRE A INFORMAÇÃO GEOGRÁFICA DISTRIBUÍDA E INFORMAÇÃO GEOGRÁFICA LOCAL**

Ricardo José Leal dos Santos

Março-2006

Orientador: Margareth Simões Penello Meirelles

Programa: Engenharia de Computação - Área de Concentração Geomática

Normalmente os Sistemas de Informações Geográficas (SIG) visam a aquisição e tratamento de dados locais, sendo construídos como sistemas proprietários fechados. A necessidade de distribuição surge posteriormente. Este trabalho busca levantar os problemas que dificultam a integração semântica entre a informação geográfica distribuída e a local. Após extenso estudo, definiu-se que os principais obstáculos para a integração semântica são: o grande volume de dados; os diversos tipos de armazenamento; a diversidade de fontes de informação; os formatos heterogêneos de dados; a dependência da estrutura e localização dos dados; a falta de estruturação dos metadados; a incompatibilidade entre os dados e a diversidade de linguagens de consulta. Estes problemas indicam a necessidade da integração semântica no momento do desenvolvimento do SIG. Para tanto, recomenda-se que a Geographic Markup Language (GML) seja usada não somente como formato de distribuição, mas também como resposta padrão às consultas e armazenamento padrão de documentos locais; que a XQuery estendida com operadores espaciais seja utilizada tanto para consulta GML quanto para informações armazenadas em SGBD; e que a GML seja uma ontologia geográfica que traga maior semântica ao sistema e em conjunto com a Resource Description Framework (RDF) permita o armazenamento e a distribuição da semântica. Assim, o uso dessa recomendação capacitaria um SIG a ser um Centro de Dados.

Palavras-chave: Banco de Dados Espaciais, GML, Interoperabilidade, Integração Semântica

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

# **STUDY OF THE PROBLEMS ON SEMANTIC INTEGRATION AMONG DISTRIBUTED GEOGRAPHIC INFORMATION AND LOCAL GEOGRAPHIC INFORMATION**

Ricardo José Leal dos Santos

March/2006

Advisor: Margareth Simões Penello Meirelles.

Program: Computing Engineering – Geomatic

Normally, Geographic Information System (GIS) aims the acquisition and treatment of local data, being built as a closed proprietary systems. The need of distribution appears later. This work looks for rising the problems that difficult the semantic integration among the distributed geographic information and local geographic information. After a vast study, it was found that the main obstacles for the semantic integration are: large data volume; different types of data storing; different information sources; heterogeneous data formats; lack of metadata structure, dependence of data and different query languages. Those problems indicate the need of semantic integration in the moment of the development of the GIS. So, it is recommended that Geographic Markup Language (GML) be used, not only as a distribution format, but also as an standard result to the queries and a standard storage for local documents; that the use of the XQuery extended with spatial operators be used such for consulting GML as to information stored at DBMS; that GML be a geografic ontology that brings much more semantic to the system and together with Resource Description Framework (RDF) allows the storage and distribution of the semantic. Thus, the use of this recommendation could enable a GIS to be a Data Center.

Keywords: Geospatial Database, GML, Interoperability, Semantic Integration

## ÍNDICE GERAL

<b>DEDICATÓRIA .....</b>	<b>I</b>
<b>AGRADECIMENTOS .....</b>	<b>II</b>
<b>ÍNDICE GERAL.....</b>	<b>V</b>
<b>ÍNDICE DE FIGURAS .....</b>	<b>VII</b>
<b>ÍNDICE DE TABELAS.....</b>	<b>VIII</b>
<b>LISTA DE SIGLAS .....</b>	<b>IX</b>
<b>CAPÍTULO 1 - INTRODUÇÃO .....</b>	<b>1</b>
1.1. INTRODUÇÃO .....	1
1.2. OBJETIVOS GERAIS .....	3
1.3. OBJETIVOS ESPECÍFICOS .....	4
1.4. METODOLOGIA .....	4
1.5. DISTRIBUIÇÃO DOS CAPÍTULOS.....	5
<b>CAPÍTULO 2 - A INFORMAÇÃO GEOGRÁFICA DISTRIBUÍDA .....</b>	<b>6</b>
2.1. A INTEROPERABILIDADE.....	6
2.1.1. <i>Heterogeneidade x Interoperabilidade</i> .....	6
2.2. A LINGUAGEM XML.....	9
2.2.1. <i>Definindo a XML</i> .....	10
2.2.2. <i>Informação Estruturada e com protocolo aberto</i> .....	10
2.2.3. <i>Estrutura de um documento XML</i> .....	12
2.2.4. <i>Vantagens do uso da XML</i> .....	12
2.2.5. <i>Componentes básicos de um documento XML</i> .....	14
2.2.6. <i>Componentes externos</i> .....	18
2.2.7. <i>XQuery</i> .....	26
2.3. A LINGUAGEM RDF.....	30
2.3.1. <i>Triplas RDF</i> .....	31
2.3.2. <i>Tipos RDF</i> .....	32
2.4. A LINGUAGEM GML.....	33
2.4.1. <i>OGC</i> .....	34
2.4.2. <i>Feições Geográficas Simples</i> .....	35
2.4.3. <i>Modelo de Objetos</i> .....	36
2.4.4. <i>Usando a GML</i> .....	38
<b>CAPÍTULO 3 - ARMAZENAMENTO EM SGBD ESPACIAL.....</b>	<b>44</b>
3.1. PARA ENTENDER O ARMAZENAMENTO .....	44
3.2. ORACLE SPATIAL.....	45
3.2.1. <i>Indexação dos dados espaciais</i> .....	47
3.2.2. <i>Relacionamento espacial e filtragem</i> .....	47
3.2.3. <i>Funções agregadas espaciais</i> .....	48
3.2.4. <i>Como manter os metadados</i> .....	50
3.3. MYSQL E SUA EXTENSÃO ESPACIAL.....	51
3.3.1. <i>Extensão espacial</i> .....	51
3.3.2. <i>Criação de colunas espaciais</i> .....	52
3.3.3. <i>Para popular colunas espaciais</i> .....	53
3.3.4. <i>Para recuperar dados espaciais</i> .....	53



3.4. POSTGRESQL E SUA EXTENSÃO ESPACIAL POSTGIS .....	54
3.4.1. <i>A extensão espacial PostGis</i> .....	54
3.5. COMPARANDO OS TRÊS BANCOS: .....	57
<b>CAPÍTULO 4 - O PROBLEMA DA INTEGRAÇÃO SEMÂNTICA.....</b>	<b>58</b>
4.1. CENÁRIO I: UMA APLICAÇÃO TÍPICA DE UM AMBIENTE GEOESPACIAL LOCAL.....	58
4.1.1. <i>Aquisição de dados</i> .....	59
4.1.2. <i>Funções de processamento</i> .....	59
4.1.3. <i>Banco Geoespacial</i> .....	60
4.1.4. <i>Consulta</i> .....	61
4.2. CENÁRIO II: O COMPONENTE DE DISTRIBUIÇÃO DA INFORMAÇÃO GEOGRÁFICA ....	62
4.2.1. <i>Centro de Dados Geoespaciais</i> .....	63
4.2.2. <i>Aquisição de Informações Externas</i> .....	64
4.2.3. <i>Conversão e Migração</i> .....	64
4.2.4. <i>Consulta Externa</i> .....	65
4.2.5. <i>Informação Distribuída</i> .....	66
4.3. OBSTÁCULOS PARA INTEGRAÇÃO SEMÂNTICA: .....	67
<b>CAPÍTULO 5 - O ESTADO DA ARTE DO ESTUDO DA INTEGRAÇÃO DA INFORMAÇÃO GEOGRÁFICA DISTRIBUÍDA.....</b>	<b>68</b>
5.1. O ESTADO DA ARTE EM CONSULTA SEMÂNTICA .....	68
5.2. O ESTADO DA ARTE EM ARMAZENAMENTO EM SGBD.....	74
5.3. O ESTADO DA ARTE EM INTEGRAÇÃO DA INFORMAÇÃO.....	79
<b>CAPÍTULO 6 - RECOMENDAÇÕES PARA INTEGRAÇÃO SEMÂNTICA .....</b>	<b>80</b>
6.1. AMBIENTE INTEGRADO SEMANTICAMENTE .....	80
6.2. ARMAZENAMENTO PADRÃO .....	81
6.3. FORMATO DE DISTRIBUIÇÃO PADRÃO.....	81
6.4. FORMATO DE METADADOS PADRÃO.....	82
6.5. LINGUAGEM DE CONSULTA PADRÃO .....	82
6.6. CENÁRIO UTILIZANDO A RECOMENDAÇÃO PROPOSTA .....	83
6.6.1. <i>Consulta Semântica Integrada</i> .....	84
<b>CAPÍTULO 7 - CONSIDERAÇÕES FINAIS .....</b>	<b>85</b>
7.1. CONCLUSÃO.....	85
7.2. SUGESTÕES E RECOMENDAÇÕES .....	86
<b>REFERÊNCIAS.....</b>	<b>87</b>

## ÍNDICE DE FIGURAS

Figura 2.1 – Estrutura de um documento XML.....	12
Figura 2.2 – Vantagens do uso da XML. ....	12
Figura 2.3 – Estrutura em árvore dos elementos. ....	15
Figura 2.4 – Modelo de feições usado na OGC.....	37
Figura 2.5 – Modelo de Geometria para Feições Simples da OGC.....	37
Figura 2.6 – Os três perfis da GML. ....	39
Figura 3.1 – Tipos de geométricos bidimensionais suportadas pelo Oracle Spatial[27].	46
Figura 3.2 – Relacionamento topológico do SDO_RELATE[27]. ....	48
Figura 4.1 – Cenário típico de um ambiente Geoespacial Local. ....	58
Figura 4.2 – Cenário anterior acrescido da componente distribuição. ....	62
Figura 5.1 – Convertendo um documento XML em DOM. ....	77
Figura 6.1 – Cenário utilizando a recomendação de integração semântica. ....	83

## ÍNDICE DE TABELAS

Tabela 2.1 – Comparação entre a edição estruturada e a informação estruturada.....	11
Tabela 2.2 – Exemplo da DTD. ....	11
Tabela 2.3 – Comparação entre parte de arquivo binário com um arquivo XML. ....	13
Tabela 2.4 – Exemplo de elemento em XML. ....	15
Tabela 2.5 – Exemplo de um arquivo XML para acesso a banco. ....	16
Tabela 2.6 – Exemplo do início de um documento XML. ....	16
Tabela 2.7 – Exemplo de entidades internas. ....	17
Tabela 2.8 – Exemplo de entidades externas. ....	18
Tabela 2.9 – Exemplo do uso do namespaces. ....	19
Tabela 2.10 – Exemplo do uso do XLink. ....	20
Tabela 2.11 – Exemplo de um XLink estendido. ....	21
Tabela 2.12 – Exemplo 1 do uso da XSL. ....	23
Tabela 2.13 – Exemplo 2 do uso da XSL. ....	23
Tabela 2.14 – Mudança de Formato com XSLT. ....	24
Tabela 2.15 – Exemplos de XML Schema. ....	26
Tabela 2.16 – Exemplo da Cláusula FOR. ....	27
Tabela 2.17 – Exemplo da Cláusula LET. ....	28
Tabela 2.18 – Exemplo das Cláusulas FOR e LET. ....	28
Tabela 2.19 – Exemplo da Cláusula WHERE. ....	28
Tabela 2.20 – Sintaxe básica da linguagem XQuery. ....	29
Tabela 2.21 – Comparação entre o modelo Relacional e o da XML. ....	29
Tabela 2.22 – Exemplos de URI. ....	31
Tabela 2.23 – Gramática de tuplas na GML. ....	41
Tabela 2.24 – Exemplo do elemento Coordinates. ....	42
Tabela 2.25 – Elementos geométricos da GML. ....	43
Tabela 3.1 – Modelo de dados usado no Oracle Spatial. ....	46
Tabela 3.2 – Operadores de relacionamento espacial do Oracle Spatial. ....	47
Tabela 3.3 – Codificação Oracle dos Tipos de Geometrias. ....	48
Tabela 3.4 – O tipo SDO_GEOMETRY. ....	49
Tabela 3.5 – Exemplo do uso do Oracle Spatial. ....	50
Tabela 3.6 – Infra-Estrutura para o sistemas de coordenadas do Oracle. ....	51
Tabela 3.7 – Funções espaciais no MySQL. ....	52
Tabela 3.8 – Inserindo informação Geoespacial no MySQL. ....	53
Tabela 3.9 – Exemplo de utilização no PostGIS. ....	55
Tabela 3.10 – Exemplo das formas padrão e canônicas OGC no PostGis. ....	56
Tabela 3.11 – Operadores do PostGis. ....	56
Tabela 3.12 – Comparação entre Oracle, MySQL e PostgreSQL. ....	57
Tabela 4.1 – Obstáculos à integração semântica. ....	67
Tabela 5.1 – Classificação de grupos de abordagens de armazenamento da XML. ....	75
Tabela 5.2 – As cinco estratégias de armazenamento definidas por Tian. ....	76
Tabela 5.3 – Exemplo de Conversão entre a XQuery e a SQL3. ....	78
Tabela 6.1 – Tabela de problemas e soluções propostas. ....	80

## LISTA DE SIGLAS

CORBA	- Common Object Request Broker Architecture
DAML	- DARPA Agent Markup Language
DER	- Departamento de Estradas de Rodagem
DOM	- Document Object Model
DTD	- Data Type Definition
FAT	- File Address Table
GIF	- Graphic Image Format
GML	- Geographic Markup Language
GPS	- Global Positioning Systems
HTML	- Hiper Text Markup Language
ICMS	- Imposto de Circulação de Mercadorias e Serviços
IGD	- Informação Geográfica Distribuída
IGL	- Informação Geográfica Local
ISO	- International Organization for Standardization
LBS	- Location Based Service
NFS	- Network File System
NTFS	- New Technology File System
OGC	- Open Geospatial Consortium
PDF	- Portable Document Format
RDF	- Resource Description Framework
RTF	- Rich Text Format
SDE	- Spatial Database Engine
SGBD	- Sistema Gerenciador de Banco de Dados
SGML	- Standard Generalized Markup Language
SIG	- Sistema de Informação Geográfica
SMI	- Sistema de Monitoração Imobiliária
SOAP	- Simple Object Access Protocol
SQL	- Structed Query Language
SRS	- Spatial Reference System
SVG	- Scalable Vector Graphic
URI	- Uniform Resource Identifier

URL	- Uniform Resource Locator
URN	- Uniform Resource Name
VML	- Vector Markup Language
VRML	- Virtual Reality Markup Language
W3C	- World Wide Web Consortium
XML	- eXtensible Markup Language
XPointer	- XML Pointer Language
XSL	- eXtensible Stylesheet Language
XSLT	- eXtensible Stylesheet Language Transformation

## CAPÍTULO 1 - INTRODUÇÃO

*"Não existe mudança, não importa quão ruim, que não beneficie alguma pessoa; nem mudança, não importa quão boa, que não prejudique alguém".*

Vemblen (economista)

### 1.1. Introdução

Antes do advento da Internet, buscar informações espaciais restringia o pesquisador a utilizar um número limitado de fontes “confiáveis”, pois a aquisição da informação geográfica tinha um custo bastante alto e mal aproveitado. Esse desperdício devia-se ao conjunto adquirido de informações ser bem maior do que era utilizado prontamente.

Com o desenvolvimento da Internet, passaram a existir novas fontes para aquisição da informação geográfica. Desta forma, surge a possibilidade de adquirir somente o necessário, no momento desejado e, também, que essa informação esteja atualizada. Criaram-se, assim, novos problemas para integrar a informação já existente com a informação recebida dessas novas fontes.

No início da Internet, referenciar e recuperar informações era uma tarefa para especialistas. Só estavam disponíveis protocolos pouco amigáveis, tal como telnet, para acesso remoto, e protocolos proprietários para acesso a informação. Após encontrar a informação relevante, era necessário copiá-la para utilizá-la em conjunto com as informações locais e, neste caso, aspectos como nível de relevância, atualização e precisão tinham de ser considerados.

Atualmente, existem diversos modelos, protocolos e tecnologias de armazenamento voltados à distribuição da informação geográfica. Entretanto, não adianta modelar os dados, estruturá-los e armazená-los eficientemente se não há como recuperar a informação com eficiência. É preciso obter a informação sem a preocupação de responder os seguintes questionamentos: onde a informação está? Como ter acesso a informação? O quanto essa informação é confiável? E, principalmente, como integrá-la às informações locais já existentes?

Puramente distribuir a informação geoespacial para responder a esses questionamentos é ineficiente, torna-se necessário então um arcabouço auxiliar para armazenar os metadados e regras de inferências. Assim, conhecer o real significado das informações geoespaciais que se quer distribuir, isto é, a semântica por trás das informações. A integração semântica é a característica mais desejável em qualquer sistema de informação[31].

Em pesquisa realizada pela Open Geospatial Consortium (OGC) de 2001, constatou-se que mais de 80% das informações gerais contêm componentes espaciais, Dessa forma, há uma tendência da informação espacial integrar-se, gradativamente, aos sistemas convencionais e assim aumentar o número de fontes de informação geoespacial.

Os Sistemas de Informações Geográficas, por sua vez, armazenam informações onde há características de interesse público muito forte e onde a necessidade de distribuir dados é muito maior que a dos sistemas convencionais. Por isso, há uma preocupação mundial com integração da informação geoespacial para garantir o acesso de um número maior de pessoas.

Entretanto, apesar de haver um consenso da necessidade da integração, os SIG(s) não incorporam naturalmente essa tecnologia. Isso talvez ocorra devido aos SIG(s) serem altamente interdisciplinares. Além disso, temos o fato de as equipes de desenvolvimento serem compostas por profissionais de Ciências da Terra (cartógrafos, geógrafos, geólogos etc.) que, por esse motivo, não têm a característica de desenvolvedores de sistemas, o que os leva a adotar uma ótica local, ou seja, sem a preocupação com integrações futuras.

A integração das informações geoespaciais é obtida, inicialmente, com a interoperabilidade dos SIG(s). Entretanto, a maioria deles é construída sem essa preocupação. Esse descaso inicia-se na etapa de seu desenvolvimento. O desenvolvedor, devido à diversidade de modelos de dados e plataformas disponíveis, adota a sua arquitetura preferida, causando a heterogeneidade, a qual irá dificultar ou mesmo impossibilitar a interoperabilidade entre os SIG(s).

Após a etapa de desenvolvimento, surge normalmente a necessidade de integração com informações advindas de outras bases geoespaciais (Centro de Dados), tais como: agências privadas, governamentais ou de outros SIG(s). Isso ocorre quando as informações geográficas locais, adquiridas anteriormente, dessas mesmas fontes, ficam obsoletas. Nessa hora se almeja por informações consistentes e atualizadas.

Se cada Centro de Dados tivesse maneiras diferentes para distribuir a informação geográfica, então, seriam necessárias diferentes formas para receber e administrar essas informações. Pois, organizar todos os dados em um mesmo formato e armazená-los em um mesmo lugar não os fazem integrados. Integrar é fazer com que a informação seja única entre os sistemas envolvidos rompendo as barreiras para sua distribuição.

O armazenamento persistente dos dados geoespaciais é, geralmente, muito caro. Frequentemente, dados de mapas digitalizados e imagens de satélites são armazenados em um mesmo banco de dados. Esse armazenamento é fonte de imprecisão e incertezas, porque, além do trabalho de manter os dados consistentes entre si, existe a necessidade de manter a consistência entre as mudanças do mundo real às respectivas mudanças no banco de dados.

Os Sistemas Gerenciadores de Banco de Dados Relacionais (SGBDRs) estão melhor preparados para tratar dados convencionais. Por isso, apresentam deficiências quando utilizados para armazenamento espacial que necessitam de um tratamento específico de seus dados. Pode-se citar como, por exemplo, o modelo utilizado, muito pobre para representar dados do tipo campo. Além disso, surgem outros problemas. Onde guardar os arquivos com as informações geográficas recebidas de outros Centros de Dados? Como os dados recebidos e os dados locais são integrados?

A proposta deste trabalho é estudar mais a fundo estes, problemas para que pesquisadores utilizem este trabalho como subsídio ao entendimento dos problemas e, por conseguinte, apontem soluções para tornar os SIG(s) futuros Centros de Dados [6].

Baseado no exposto, a hipótese deste trabalho é que o projeto de SIG para se tornar um Centro de Dados deve estar baseado em uma ótica distribuída desde o início do projeto.

## **1.2. Objetivos Gerais**

Estudar os problemas para a integração semântica entre a Informação Geográfica Local (IGL) e a Informação Geográfica Distribuída (IGD).

Propor recomendações no projeto de um SIG para a integração semântica da informação.



### 1.3. Objetivos Específicos

- Pesquisar o estado da arte em integração semântica da informação geoespacial.
- Reunir soluções dispersas numa proposta de recomendação. E com isso, propor um referencial e direcionamento para os projetos em SIG.
- Construir um cenário de um SIG para análise do problema da integração semântica.
- Motivar o uso da GML e das funcionalidades geoespaciais<sup>1</sup> dos bancos de dados para a distribuição da informação geoespacial.
- Estudar os padrões abertos (XML, RDF etc.) e os SGBD (Oracle, PostGreSQL, MySQL etc.) mais adequados para a distribuição.

### 1.4. Metodologia

Para alcançar os objetivos mencionados, inicialmente, fez-se uma pesquisa prévia com levantamento bibliográfico e entrevistas informais, a fim de definir os problemas encontrados para a integração semântica dos SIG(S).

Com esses dados, criou-se um cenário modelo, primeiramente, com enfoque local e, posteriormente, acrescentou-se a componente de distribuição, onde pôde-se observar explicitamente os problemas pesquisados e abordar exemplos para compreender-se esta questão.

Ao final, classificou-se o problema da integração semântica. Em seguida, buscou-se o estado da arte relacionado aos problemas encontrados e estudaram-se as tecnologias propostas para resolver ou minimizar tais problemas. Finalmente, fez-se uma avaliação sobre as possíveis soluções a fim de produzir uma recomendação para a integração semântica entre a Informação Geográfica Local a Informação Geográfica Distribuída.

---

<sup>1</sup> Entende-se por funcionalidades geoespaciais as características adicionais que têm um SGBD para atender ao armazenamento e consulta dos dados geoespaciais.

## **1.5. Distribuição dos Capítulos**

Nos capítulos 2 e 3 revisam-se os principais conceitos e tecnologias relativas, respectivamente, à informação geográfica distribuída e ao armazenamento da informação geoespacial em Sistemas Gerenciadores de Banco de Dados, que serão úteis para o entendimento pleno da análise dos problemas e recomendações propostas. No capítulo 4, apresenta-se uma análise dos problemas relacionados à integração semântica a partir da avaliação de dois cenários de ambientes geoespaciais com o objetivo de descobrir os obstáculos para esta integração. No capítulo 5, mostra-se o estado da arte dos estudos de soluções para integração da informação geográfica distribuída, com o objetivo de descobrir um direcionamento para recomendação proposta. No capítulo 6, é apresentada a recomendação proposta para integração semântica entre a informação geográfica local e a distribuída. Finalmente, no Capítulo 7, apresentam-se as considerações finais deste trabalho.

## CAPÍTULO 2 - A INFORMAÇÃO GEOGRÁFICA DISTRIBUÍDA

*“Nunca ande pelo caminho traçado, pois  
ele conduz somente até onde os outros  
foram.”*

Alexander Graham Bell

Neste capítulo serão apresentados os principais conceitos relacionados à informação geográfica distribuída e às tecnologias utilizadas para este fim. Inicialmente, apresentam-se conceitos iniciais sobre interoperabilidade, que é a base para integração da informação geográfica, e com isso, cria-se um alicerce no entendimento das tecnologias XML[7], RDF e GML que serão úteis para a análise dos problemas.

### 2.1. A Interoperabilidade

A Internet, a *Web* e as infra-estruturas de computação distribuídas têm possibilitado a distribuição da informação em um ambiente, que é autônomo e heterogêneo. Acessar informações relevantes e precisas tem se tornado altamente complexo. Esta complexidade é exacerbada devido à heterogeneidade estrutural e à semântica dos sistemas envolvidos. A interdisciplinaridade, a grande variedade de tecnologias e geotecnologias dificultam a real interoperabilidade neste ambiente[8].

Interoperabilidade é a capacidade que um sistema possui de compartilhar e trocar informações e aplicações<sup>2</sup>.

Segundo OUKSEL *et al* [9], existem várias formas de classificação da interoperabilidade baseadas nos correspondentes tipos de heterogeneidade. Neste trabalho apresenta-se uma destas classificações para o entendimento do problema interoperabilidade.

#### 2.1.1. Heterogeneidade x Interoperabilidade

Heterogeneidade é a diversidade que ocorre nos elementos dos sistemas de informação que dificultam a interoperabilidade. Dentre os tipos de heterogeneidade existentes destacam-se os seguintes:

---

<sup>2</sup> Definição proposta por Bishr, Y. (1997) *Semantic Aspect of Interoperable GIS*. Ph.D. Thesis, Wageningen Agricultural University, The Netherlands.

- **Heterogeneidade da Informação:** é a possibilidade de se representar a informação de formas distintas, a partir da perspectiva do observador. Por exemplo, uma feição (elemento geográfico) pode ser classificada de uma forma por um cartógrafo e de outra por um geólogo.

- **Heterogeneidade de Sistemas:** é como a informação está disposta em uma diversidade de sistemas e em várias camadas, como: na camada de armazenamento podem ser utilizados distintos sistemas, por exemplo SGBD(s) (Oracle, MySQL, PostgreSQL), na camada da aplicação utilizam-se diversos SIG(s), tais como: ArcGIS, Geomedia, Spring etc.

- **Heterogeneidade de Plataformas:** é o nível mais alto que se tem de heterogeneidade. A informação pode estar representada em diversas plataformas básicas de *hardware* e *software* - com o uso da Internet, esta heterogeneidade deixou de ser um problema em potencial. Vários protocolos abstraíram a heterogeneidade de plataforma, como por exemplo: Sistemas Operacionais (Windows 2000, Unix, Solaris, OS390) e Sistemas de Gerenciamento de Arquivos (NTFS, NFS, FAT).

Para que um usuário do sistema entenda melhor a informação em seu ambiente, torna-se essencial reduzir a necessidade de se conhecer o conteúdo semântico e estrutura das diversas fontes de informação geográfica. Por exemplo, através do uso de ontologias específicas de domínio, nas quais, este usuário, por familiaridade com o domínio, poderá entender o conteúdo mais facilmente.

Os principais níveis de interoperabilidade são:

- **Interoperabilidade Sintática:** é a capacidade de um sistema heterogêneo ter uma única estrutura formal, linguagem, lógica, dados, registros, softwares, arquivos etc. A XML, por permitir a estruturação formal da informação, provê a interoperabilidade sintática para todas as formas de acessos a dados na *Web*. Por exemplo, se todos os SIG(s) gerassem documentos em GML as informações contidas em qualquer documento seriam compartilhadas por todo o sistema.

- **Interoperabilidade Sistemática:** é a capacidade das aplicações de se relacionarem entre si. Neste patamar, existem os padrões CORBA e IIOP que são utilizados para interação entre objetos distribuídos e componentes.

- **Interoperabilidade Semântica:** é o maior desafio da interoperabilidade. Ter em um sistema heterogêneo o significado real dos objetos, suas proposições, validações, denotações etc.

OUKSEL[9] identificou que, para alcançar esta interoperabilidade, é preciso implementar quatro capacidades:

1 - *Transparência de nomes e linguagem:* um usuário/sistema deve ser capaz de escolher sua ontologia.

2 - *Processamento de informações sensíveis ao contexto:* um usuário/sistema obtém pesquisas mais precisas pela filtragem e transformação da informação de acordo com o contexto em questão.

3 - *Regras para mecanismos de interação:* um usuário/sistema cria regras e mecanismos para especificação de formato de mensagens e tipos de dados.

4 - *Correlação semântica:* Um usuário/sistema correlaciona informações em ambientes distribuídos e heterogêneos segundo uma semântica.

Para se obter a interoperabilidade semântica utiliza-se três componentes chave:

1 - **Metadados:** são dados que descrevem os dados. Para interoperabilidade semântica devem-se usar os que especificam domínio e baseados em contexto.

2 - **Contexto:** é caracterizado pela similaridade entre objetos baseados nas associações semânticas entre eles. Muitas vezes, não é possível definir-se completamente o significado de um objeto no mundo real, mas o entendimento do contexto ajuda o sistema a distinguir diferenças, por exemplo se a palavra manga aparecer próxima ou relacionada a palavra suco é mais provável que se trate de manga fruta do que de manga de camisa.

3 - **Ontologia:** é definida como sendo um vocabulário específico e suas relações utilizadas para descrever certos aspectos da realidade, e um conjunto aceitável explícito considerando o significado pretendido das palavras.

CÂMARA *et al* [3], investigou, na prática, a conversão semântica de sistemas de informações geoespaciais usando o modelo proposto pelo OGC. Em seu trabalho, os

sistemas MGE, ARCINFO e SPRING foram analisados. Nesta análise, percebeu-se a dificuldade de efetivamente obter a interoperabilidade devido às diferentes semânticas adotadas em cada modelo. Sua conclusão mostra que o modelo adotado pela OGC não observa totalmente as quatro capacidades descritas anteriormente. Isto causa problemas na adoção desta solução em sistemas reais.

## 2.2. A Linguagem XML

A linguagem XML (eXtensible Markup Language) [7] é uma metalinguagem de marcação<sup>3</sup> utilizada para distribuição da informação de forma aberta e estruturada. Foi inicialmente desenvolvida, em julho de 1996, por um grupo liderado por Jon Bosak (Sun Microsystems) e, sua equipe: Tim Bray (Netscape), Jean Paoli (Microsoft) e Michael Sperber-McQueen (Universidade de Chicago). Todos pertencentes a W3C [10](World Wide Web Consortium).

Essa poderosa linguagem é um subconjunto da SGML (Standard Generalized Markup Language), do qual se retirou parte da complexidade e adaptou-se para o uso na Internet. Em 1986, a SGML tornou-se um padrão ISO para criação de linguagens de marcação. Tem o objetivo de descrever o conteúdo de documentos e, então, publicar o mesmo conteúdo de diferentes maneiras. A SGML permite que os autores separem o conteúdo lógico do documento de sua representação, mas é de extrema complexidade.

Já a linguagem HTML (Hiper Text Markup Language), criada por Tim Berners-Lee, que teve seu início em 1990, é uma aplicação SGML, isto é, foi escrita na linguagem SGML.

A solução XML para metadados e contexto compartilhado é simples. Autores adicionam metadados através de marcações e o desenvolvedor do documento adiciona o contexto compartilhado através da Data Type Definition (DTD).

A especificação W3C atual da XML é a versão 1.0 que define regras de sintaxe para criar as marcações estendidas e seus atributos.

---

<sup>3</sup> São elementos que identificam o conteúdo da informação dentro do documento. As marcações são reconhecidas por seus delimitadores “<” e “>”. Exemplo: <title> isso é um título</title>.

### **2.2.1. Definindo a XML**

A principal característica da linguagem XML é a capacidade de ser extensível, isto é, o autor de um documento XML tem como criar suas próprias marcações e estas serem utilizadas por um processador XML, que será, por exemplo, um *plugin* de algum navegador (como Internet Explorer ou Firefox). Essa característica é a base de um documento estruturado, cujo nível de processamento associado é alto. O fato de poder ser processado é uma característica fundamental para a distribuição da informação [11,12]. Analogamente, é como comparar um arquivo com texto livre, isto é, sem nenhuma estruturação, com tabelas em banco de dados, altamente estruturado. Vale a pena destacar que o poder de resposta dos bancos de dados não está no poder de processamento de seus algoritmos, mas, principalmente, na estrutura em que os dados são armazenados.

### **2.2.2. Informação Estruturada e com protocolo aberto**

A informação estruturada traz vários benefícios, entre eles: a facilidade de pesquisa, a manutenção dos dados, a integração das informações etc. O uso de um protocolo aberto permite que qualquer um escreva um processador para esta linguagem e a distribua com um conjunto pequeno de códigos (ASCII).

Os documentos HTML só têm a edição do documento estruturada. As informações inseridas não têm nenhuma estrutura associada. Então, é comum encontrar muitas informações inúteis e poucas realmente úteis ao se fazer uma pesquisa[13]. Os *sites* de pesquisas utilizam-se de heurísticas para melhorar este resultado, o que ainda é pouco eficiente, porque sem a informação estruturada não é possível conhecer o significado real do que se procura.

Na Tabela 2.1, observa-se que em ambos os casos um leitor humano não teria dificuldades para interpretar corretamente a informação que se deseja transmitir. Mas por outro lado, um computador estaria mais apto a processar as informações no formato 2 do que no 1. Por exemplo, o conhecimento de que verde é uma cor, não aparece descrito no formato 1. Se este documento fosse extenso e alguém desejasse saber qual a descrição da legenda verde, seria muito mais fácil para um processo automático procurar no formato 2.

**Tabela 2.1 – Comparação entre a edição estruturada e a informação estruturada.**

1 - Edição Estruturada	2 - Informação Estruturada
<pre> &lt;HTML&gt; &lt;title&gt;Região do Pantanal Mato-Grossense &lt;head&gt;Mapa Temático de Uso de Solo &lt;body&gt;   &lt;h1&gt; Legenda   &lt;h2&gt; verde - lavoura   &lt;h3&gt; cinza - indústria   &lt;h4&gt; azul - improdutiva &lt;/body&gt; </pre>	<pre> &lt;Regiao&gt;   &lt;nome&gt;Região do Pantanal Mato-Grossense &lt;/nome&gt;   &lt;mapa&gt;     &lt;tipo&gt;Temático&lt;/tipo&gt;     &lt;nome_mapa&gt;Uso de Solo&lt;/nome_mapa&gt;     &lt;legenda&gt;       &lt;item&gt;&lt;cor&gt;verde&lt;/cor&gt;&lt;desc&gt;Lavoura&lt;/desc&gt;&lt;/item&gt;       &lt;item&gt;&lt;cor&gt;cinza&lt;/cor&gt;&lt;desc&gt;indústria&lt;/desc&gt;&lt;/item&gt;       &lt;item&gt;&lt;cor&gt;azul&lt;/cor&gt;&lt;desc&gt;improdutiva&lt;/desc&gt;&lt;/item&gt;     &lt;/legenda&gt;   &lt;/mapa&gt; &lt;/Regiao&gt; </pre>

Na Tabela 2.2 pode-se ver um exemplo de uma DTD (Data Type Definition) para a estrutura XML apresentada na Tabela 2.1. Os caracteres "?" e "\*" definem o número de vezes que o elemento pode aparecer. Para o caractere "?" 0 ou 1 vez e para o caractere "\*" 0 ou "n" vezes.

**Tabela 2.2 – Exemplo da DTD.**

A DTD
<pre> &lt;!ELEMENT regiao (nome, mapa)&gt; &lt;!ELEMENT nome (#PCDATA)&gt; &lt;!ELEMENT mapa (tipo, nome_mapa, legenda?)&gt; &lt;!ELEMENT tipo (#PCDATA)&gt; &lt;!ELEMENT nome_mapa (#PCDATA)&gt; &lt;!ELEMENT legenda (item*)&gt; &lt;!ELEMENT item (cor, desc)&gt; &lt;!ELEMENT cor (#PCDATA)&gt; &lt;!ELEMENT desc (#PCDATA)&gt; </pre>



### 2.2.3. Estrutura de um documento XML

A estrutura de um documento XML pode ser observada na Figura 2.1:

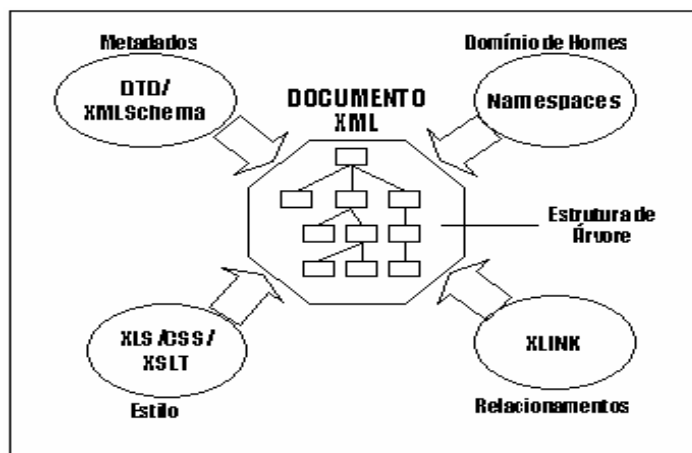


Figura 2.1 – Estrutura de um documento XML.

As informações ficam armazenadas no próprio documento XML em forma de árvore. Já os metadados ficam armazenados em outro documento (DTD ou XMLSchema). O domínio de nomes (identifica a ontologia das marcações) fica armazenado em URL. O estilo, isto é, como a informação é apresentada, fica armazenado em outro documento (XLS ou CSS ou XSLT). E os relacionamentos, que são as ligações entre os documentos, em outro documento (XLink).

### 2.2.4. Vantagens do uso da XML

A Figura 2.2 apresenta as principais vantagens do uso da XML para a estruturação da informação.

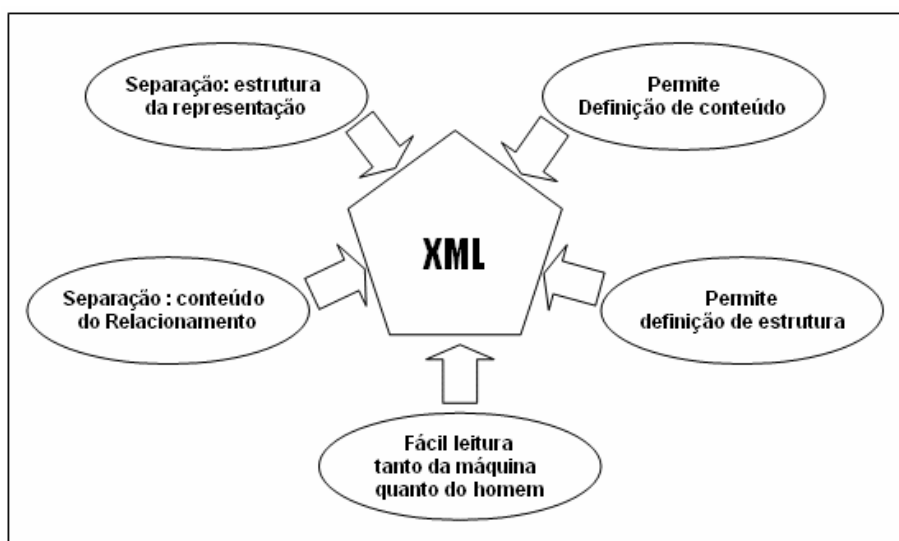


Figura 2.2 – Vantagens do uso da XML.



- **Separação entre conteúdo e relacionamento:** é possível manter a integridade dos relacionamentos, ou seja, a correta ligação entre os documentos XML, e ainda introduzir novos relacionamentos sem afetar os documentos referenciados ou ter de mudar seu conteúdo.

- **Separação entre estrutura e representação:** o estilo de exibição do documento fica em um lugar separado. Isto permite que o mesmo documento XML seja apresentado em diferentes formatos.

### **2.2.5. Componentes básicos de um documento XML**

Um documento XML é formado pelos seguintes componentes básicos:

#### **2.2.5.1. Elemento**

Um elemento é formado por seu nome e seu conteúdo. Deve ficar claro que se faz distinção quanto ao uso de letras maiúsculas e minúsculas.

O conteúdo do elemento consiste de caracteres ou outros elementos. Obrigatoriamente, um elemento deve começar com uma marcação de início (<elemento>) e terminar com uma marcação de fim (</elemento>), excetuando-se o elemento vazio que tem uma marcação especial (<elementovazio/>) que só aparece no início e não tem conteúdo de dados.

Um documento XML não precisa de uma DTD, mas usá-la é uma maneira conveniente das partes definirem claramente que estão usando o mesmo formato de dados. Existem quatro tipos de conteúdos possíveis em um elemento:

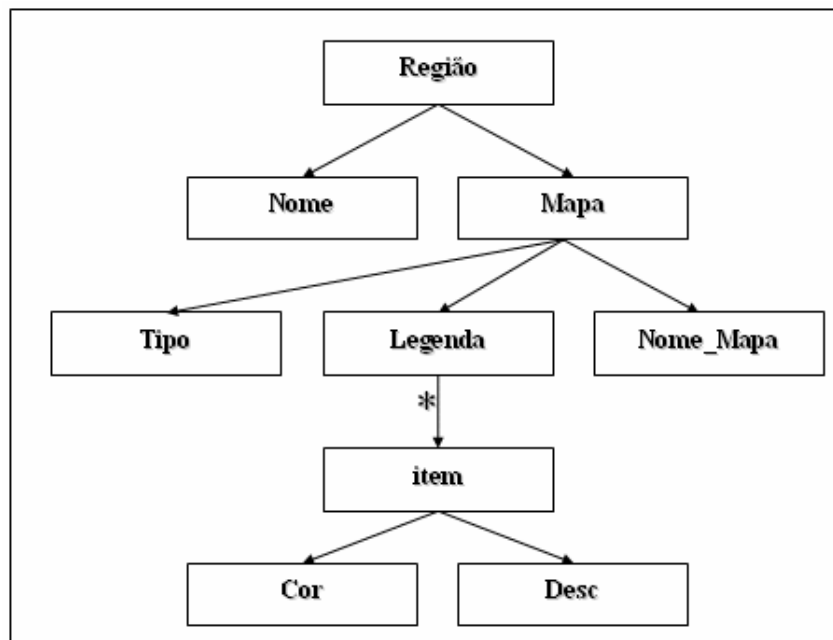
- O elemento pode conter somente dados;
- Pode conter somente outros elementos;
- Pode estar vazio;
- Pode conter uma mistura dos anteriores.

Um exemplo de cada tipo de Elemento em XML vê-se na Tabela 2.4.

**Tabela 2.4 – Exemplo de elemento em XML.**

<b>Elementos XML</b>	<b>Descrição</b>
<code>&lt;mapa&gt;</code> <code>&lt;nome_mapa&gt;</code> Mapa do Rio de Janeiro <code>&lt;/nome_mapa&gt;</code> <code>&lt;/mapa&gt;</code> <code>&lt;datum/&gt;</code>	<ul style="list-style-type: none"> <li>- Elemento raiz</li> <li>- Conteúdo Elemento</li> <li>- Conteúdo Dado</li> <li>- Fim do Elemento</li> <li>- Fim do Elemento</li> <li>- Elemento vazio</li> </ul>

A Figura 2.3 exibe a estrutura de árvore do documento mostrado na Tabela 2.4.



**Figura 2.3 – Estrutura em árvore dos elementos.**

#### **2.2.5.2. Atributos dos Elementos**

O nome do elemento, por si só, não traz toda a informação necessária para um programa processar um documento XML. Todavia, os atributos adicionam mais informação de metadados ao elemento. Zero ou mais atributos aparecem após o nome do elemento e são formados por nome do atributo e valor, no entanto, o valor do atributo sempre deverá aparecer entre aspas. Assim como no nome do elemento, o nome do atributo faz distinção entre maiúsculas e minúsculas. Alguns autores, por questão puramente de organização, costumam colocar os nomes de elementos em minúsculo e

os nomes de atributos em maiúsculo. Um exemplo de elementos com seus atributos é mostrado na Tabela 2.5. As informações adicionais dos atributos do elemento *database* permitiram que um programa efetuasse o acesso ao banco.

**Tabela 2.5 – Exemplo de um arquivo XML para acesso a banco.**

<b>Arquivo XML para acesso de banco de dados</b>
<pre> &lt;database DBTYPE="Oracle" HOSTSTRING="geodb" USERID="manager" PASSWORD="senha"&gt;   &lt;table NAME="Logradouros"&gt;     &lt;column DATATYPE="Varchar2"&gt;Nome&lt;/column&gt;     &lt;column DATATYPE="Number"&gt;Tipo&lt;/column&gt;     &lt;column DATATYPE="Varchar2"&gt;Bairro&lt;/column&gt; ...   &lt;/table&gt;   &lt;table NAME="Mapas"&gt;     &lt;column DATATYPE="Varchar2"&gt;Nome&lt;/column&gt;     &lt;column DATATYPE="Number"&gt;Area&lt;/column&gt;     &lt;column DATATYPE="Varchar2"&gt;Carta&lt;/column&gt; ...   &lt;/table&gt; &lt;/database&gt; </pre>

Um documento XML sempre começa com uma informação de cabeçalho que aparece antes do elemento raiz e contém instruções de processamento e o tipo da declaração do documento, isto é chamado de *prolog*. As instruções de processamento iniciam com "<?" e terminam com "?>". O alvo, na Tabela 2.6, é o próprio processador XML que recebe um ou mais pares de atributo e valor. No caso, a especificação da versão.

**Tabela 2.6 – Exemplo do início de um documento XML.**

<b>Cabeçalho de um documento XML</b>
<pre> &lt;?XML version="1.0"?&gt; &lt;!DOCTYPE FeatureCollection SYSTEM "gmlfeature.DTD" &gt; &lt;-- Este é um exemplo do uso de uma DTD da GML --&gt; </pre>

Um documento contém tantas instruções quantas forem necessárias ao processador. A declaração do tipo de documento indica que DTD o processador irá usar e onde encontrá-la inicia com "<!DOCTYPE " e termina com ">". A partir deste ponto é possível acrescentar os comentários, que em XML começam com "<--" e terminam com "-->". Não se terminar um comentário com três hífenens "--->".

### 2.2.5.3. DTD - Document Type Definition

O poder que a XML fornece, extrapola a criação isolada de elementos, manifesta-se quando vários documentos usam o mesmo formato de dados, ou seja, a mesma DTD. Dessa forma, o mesmo software pode processá-los e, na mesma tela, exibi-los. A DTD fornece um formato comum de dados públicos. É uma coleção de regras que especificam estruturas possíveis de uma classe de documentos, que servem tanto como modelo de referência para o desenvolvedor retirar os metadados para a construção das aplicações quanto para o computador validar documentos durante a execução. Mas estes são ditos válidos se obedecerem a todas as regras de sua DTD de origem, a qual é formada por regras para elementos e seus atributos e especifica a forma da árvore do documento. Um documento XML sempre inicia com o elemento raiz que aponta o tipo de documento indicado na DTD.

### 2.2.5.4. Entidades

Elementos e atributos são primeiramente mecanismos lógicos, os quais permitem ao autor especificar o significado lógico do conteúdo do documento. Já as Entidades são primeiramente mecanismos estruturais que permitem ao autor manipular estruturas físicas do documento.

Uma entidade associa o nome da entidade com um fragmento de conteúdo. O autor define a entidade com a declaração `<!ENTITY nomeEntidade tipo "valor">` no início do documento ou na DTD. Existem quatro tipos de entidades em XML, três aplicadas ao documento e uma aplicada à DTD.

**a - Entidades internas:** essas entidades permitem o autor definir um *alias* (apelido) para um fragmento de texto dentro de um documento. Na Tabela 2.7 apresenta-se um exemplo do uso de entidades internas é visualizada como uma frase muito repetida no texto, que poderia ser alterada com pouca manutenção.

**Tabela 2.7 – Exemplo de entidades internas.**

Na DTD	No documento
<code>&lt;!ENTITY UERJ "Universidade do Estado do Rio de Janeiro"&gt;</code> <code>&lt;!ENTITY GEO "Programa de Pós-Graduação em Computação, ênfase Geomática "&gt;</code>	A &UERJ tem um excelente curso de mestrado o &GEO;

Essa declaração faria com que a palavra &UERJ fosse trocada, pelo texto mostrado na Tabela 2.7, quando o documento fosse processado. Isso reduz o tamanho do documento além de proporcionar melhor manutenção.

**b - Entidades externas:** essas entidades habilitam ao autor definir um *alias* (apelido) para um documento externo dentro de outro documento. Isto é muito útil para gerenciar documentos longos. Na Tabela 2.8 tem-se um exemplo de entidades externas e percebe-se como vários documentos poderiam ser agrupados.

**Tabela 2.8 – Exemplo de entidades externas.**

Na DTD	No documento
<!ENTITY geopolitico SYSTEM "/mapageopolitico.xml"> <!ENTITY usodesolo SYSTEM "/usodesolo.xml"> <!ENTITY hidrografia SYSTEM "/hidrografia.xml">	&geopolitico; &usodesolo; &hidrografia;

**c - Entidades de texto e binária:** essas entidades capacitam ao autor inserir dados arbitrários em um documento. Esses dados podem não ser XML, tão pouco ser texto (ex: imagem). Portanto, o processador não tentará analisar cada entidade.

Exemplo: <!ENTITY figura.uerj SYSTEM "uerj.gif" NDATA GIF>

**d - Entidades parâmetro:** essas entidades são para uso no escopo da DTD. O projetista cria entidades que serão referenciadas dentro de declarações de elementos e não são utilizados nos documentos propriamente dito, isto é, aparece na DTD da GML, mas não aparece no documento GML.

Exemplo: <!ENTITY % lista\_feature "Point | PolyLine | Polig">  
feature (%lista\_feature) #REQUIRED>

## 2.2.6. Componentes externos

São aqueles que auxiliam a integração entre os documentos. Nessa seção apresentam-se alguns componentes externos de um documento XML:

### 2.2.6.1. XML Namespaces

Uma aplicação reconhece e obtém o conteúdo de um elemento através de seu nome. O autor, então, deve criar distinção entre os nomes dos elementos para não haver ambigüidades. Entretanto, duas aplicações podem processar o mesmo documento. Se o

mesmo termo tiver significados diferentes em aplicações diferentes, algum artifício torna-se necessário para evitar ambigüidades. O XML namespaces é uma recomendação da W3C [12] que permite aos desenvolvedores evitar colisão de nomes atribuindo o namespaces aos nomes de elementos e atributos.

Observe na Tabela 2.9 que o elemento *status* aparece em ambos, mas tem funções diferentes, isto é, *acct:status* é diferente de *ful:status*.

**Tabela 2.9 – Exemplo do uso do namespaces<sup>4</sup>.**

NAMESPACES
<pre> &lt;acct:customer xmlns:cart="http://www.foocompany.com/names/cart-ReV10"&gt;   &lt;acct:name&gt; Bar Corporation &lt;/ acct:name&gt;   &lt;acct:addressee&gt;Accounts Payable -- Bar Corporation&lt;/acct:addressee&gt;   &lt;acct:order reference="5566-1010"&gt;     &lt;acct:status&gt;invoice&lt;/acct:status&gt;   &lt;/acct:order&gt; &lt;/acct:customer&gt; &lt;ful:customer xmlns:cart="http://www.foocompany.com /names/cart-ReV10"&gt;   &lt;ful:name&gt; Bar Corporation &lt;/ful:name&gt;   &lt;ful:addressee&gt;Loading Dock&lt;/ful:addressee&gt;   &lt;ful:order reference="A9875656"&gt;     &lt;ful:status&gt;shipped&lt;/ful:status&gt;   &lt;/ful:order&gt; &lt;/ful:customer&gt; </pre>

O namespace pode ser aplicado aos elementos e atributos e a primeira vez que este é utilizado, associa-se o prefixo com o atributo reservado *xmlns*. O namespace é um URI (*Uniform Resource Identifier*). Essencialmente, indica que o proprietário de um recurso da Internet é também do namespace.

Quando um único namespace é aplicado a um elemento, todos os seus filhos podem ter seus prefixos suprimidos, permitindo alterar o namespace sem ter de mudar todos os elementos e atributos.

#### 2.2.6.2. Xlink - XML Linking Language

Para implementar o relacionamento entre documentos estruturados em XML, usa-se uma linguagem de ligação: o XLink.

A *Web* é o principal alvo para aplicações XML. Muitos autores *Web* estão familiarizados com HTML e seus links, portanto, um objetivo da especificação XLink é

<sup>4</sup> Exemplo retirado da referência [10] página 51.



fazer o uso dessa sintaxe mais direta para criar um link unidirecional de um ponto do documento fonte XML a um documento-alvo.

XLink vai além de um simples *link* HTML. Com a XML, torna-se mais fácil trocar informações estruturadas. Todavia, as pessoas freqüentemente querem relatar diferentes partes de informação de diversas maneiras. Portanto, a XLink permite que se especifique links com múltiplos documentos-alvo, isto é, um link estendido aponta para diversos links e ainda apresentá-los caracterizado pela relevância.

Um link simples é muito parecido com um link padrão HTML. A XLink é definida com a sintaxe XML, por isso para saber o que fazer com essa informação um processador XLink torna-se necessário.

Para separar as informações de link do resto do documento deve-se usar o namespace *xlink* mantido pela W3C, colocando a qualificação "xlink:" para ligar atributos de um elemento. Isso garante que o processador de XLink receba a informação como mostrado na Tabela 2.10.

**Tabela 2.10 – Exemplo do uso do XLink.**

XLink
<pre> &lt;Embrapa   xlink:form="simple"   xlink:inline="true"   xlink:href="http://www.embrapa.gov.br"   xlink:role="link Entrada-Dados"   xink:title="Embrapa Solos"   xink:show="new"   xink:actuate="user"   xlink:content-role="link Embrapa Entrada de Dados"   xlink:content-title="Link para Embrapa Solos"&gt; Embrapa Solos &lt;/Embrapa&gt; </pre>

Um link simples é um conjunto de atributos, pois compreende um conjunto de atributos sobre um elemento. O atributo "inline" indica se o conteúdo do elemento, no exemplo "Embrapa Solos", faz parte do link, enquanto o atributo "*form*" especifica que este link é do tipo simples, e o atributo "*href*" especifica a URI do documento-alvo.

O atributo "show" indica que será aberta pelo processador uma nova janela para exibir o documento-alvo; o atributo "*actuate*" instrui o processador a ativar o link quando o usuário clicar sobre este. O resto dos atributos fornece informações

semânticas para o processador sobre o link. Estes atributos podem ser especificados para um particular processador XLink.

Um outro tipo de link é o estendido, que permite a separação do relacionamento do conteúdo e múltiplos documentos-alvo.

A separação entre relacionamento e conteúdo é uma poderosa ferramenta para categorizar conteúdos na *Web*. Para um determinado conjunto lógico de documentos pode haver dezenas de caracterizações diferentes, todas implementadas com links e independentes entre si.

Um link estendido tem atributos "*form*" e "*inline*", assim como atributos que indicam informações semânticas sobre todo o link. Possui também subelementos para cada documento-alvo. Esses subelementos têm atributos que indicam: a localização de cada alvo, procedimento desejado quando se conecta ao alvo e a informação semântica específica para aquela parte do link.

Na Tabela 2.11 um link estendido é descrito.

**Tabela 2.11 – Exemplo de um XLink estendido.**

XLink Estendido
<pre>&lt;Universidades   xlink:form="extended"   xlink:inline="false"   xlink:role="Inscrição"   xlink:content-role="Formulário de inscrição"&gt;   &lt;Publica     xlink:form="locator"     xlink:href="Uerj.xml"     xlink:title="Universidade do Estado do Rio de Janeiro"     xlink:actuate="user"/&gt;   &lt;Privada     xlink:form="locator"     xlink:href="PUC.xml"     xlink:title="Pontífice Universidade Católica"     xlink:actuate="user"/&gt; &lt;/Universidades&gt;</pre>

Em XML, ainda existe o XML Pointer Language (XPointer) que permite um ponteiro para um elemento específico dentro do documento XML, já em HTML é necessário declarar um atributo nome explicitamente aonde se quer apontar.

### 2.2.6.3. XSL (eXtensible Stylesheet Language)

A XSL permite a apresentação de um documento na *Web* separando o conteúdo da informação, essa é uma das mais importantes características da XML.

O objetivo da XSL é especificar uma linguagem que permita aplicação de formatos de documentos XML diferentes, sem contaminar a lógica do conteúdo. Existem três principais requisitos para esta solução:

**a - Aplicar regras de formato aos elementos:** os autores podem especificar completamente regras de formato para cada tipo de elemento em um documento. Essas regras incluem: formato de fontes, endentação, espaçamento de linhas, formatação de tabelas e assim por diante. Aplicar regras de estilo aos elementos é muito eficiente para criar folhas de estilo em conjunto com DTD.

**b - Permitir diferentes tecnologias de exibição:** a mesma folha de estilo pode transformar o mesmo documento em uma variedade de estilos de páginas como: HTML, PDF, PostScript e RTF. Portanto, aplicando uma folha de estilo a um documento cria-se uma descrição de estilo intermediária chamada árvore de formatação de objetos.

**c - Permitir folhas de Estilo Customizadas:** o leitor escolhe e aplica a folha de estilo que desejar. Além disso, filtros são utilizados para questões de acessibilidade (cor, tamanho de fontes etc.). Essa característica também é usada para suprimir informações confidenciais, assim são obtidos perfis diferentes de acesso aos dados.

O conceito base da folha de estilo XSL é relativamente direto, mas a implementação acarreta complexidades.

A idéia fundamental é que pessoas diferentes querem ver o mesmo documento de maneiras diferentes e em formatos diferentes, mas o ideal é que o formato final possa ser escolhido pelo leitor sem que isto esteja associado ao seu perfil. As folhas de estilo são escritas usando a sintaxe XML.

Com XSL, define-se modelos (*templates*) de formatação que são aplicados a elementos e subelementos. Para selecionar um elemento em particular usa-se o namespace "xsl", para formatar a saída usa-se o namespace "fo".

No primeiro exemplo, mostrado na Tabela 2.12, a formatação será aplicada a todos os filhos do elemento rodovia.

**Tabela 2.12 – Exemplo 1 do uso da XSL.**

Uso da XSL
<pre> &lt;xsl:template match="rodovia"&gt;   &lt;fo:block     fo:font-size="large"     fo:font-weight="bold"     fo:font-family="Arial"     fo:line-height="2"     &lt;xsl:apply-templates/&gt;   &lt;/fo:block&gt; &lt;/xsl:template&gt; </pre>

No segundo exemplo, mostrado na Tabela 2.13, a formatação será aplicada a todos os filhos do elemento "rodovia" se o atributo TIPO for "federal".

**Tabela 2.13 – Exemplo 2 do uso da XSL.**

Uso da XSL
<pre> &lt;xsl:template match="rodovia[TIPO="federal"]"&gt;   &lt;fo:block     fo:font-size="small"     fo:font-weight="normal"     fo:font-family="Times"     fo:line-height="1"     &lt;xsl:apply-templates/&gt;   &lt;/fo:block&gt; &lt;/xsl:template&gt; </pre>

#### 2.2.6.4. XSLT (eXtensible Stylesheet Language Transformation)

A solução geral XML de elementos classificados em um determinado domínio às vezes suscita um novo problema: a proliferação de diferentes DTD para o mesmo tipo lógico de documento. Este cenário é muito comum em aplicações como no gerenciamento da cadeia de fornecedores onde duas companhias querem trocar conceitualmente a mesma informação, mas têm DTDs que usam internamente. Em alguns casos, vários grupos industriais estão trabalhando sobre o mesmo problema, o que gera um conflito entre padrões.

As características da XML resolvem o mencionado problema com o uso da XSLT (XSL Transformations), que é um subconjunto da XSL apropriado para transformar um documento XML de um formato em outro.

Considere o problema básico de enviar automaticamente pedidos através da Internet. A companhia “A” definiu uma DTD de pedidos e a usa internamente. Ocorre o mesmo na companhia “B” quanto a definição e o uso. Agora, as duas companhias querem trocar os pedidos automaticamente.

Quando a companhia “A” criou o documento de pedidos, era válido em relação à sua companhia. Entretanto, para a companhia “B” aceitar o pedido este tem de ser válido em relação a DTD da companhia “B”. Para resolver este problema, as duas companhias criaram um documento de transformação para traduzir as DTD de “A” para a DTD de “B”. Para entender como o XSLT trabalha considere o seguinte exemplo apresentado na Tabela 2.14:

**Tabela 2.14 – Mudança de Formato com XSLT.**

Formato de A	Formato de B	XSLT
<pre>&lt;pedido moeda="Real"&gt; ... &lt;/pedido&gt;</pre>	<pre>&lt;pedido&gt;   &lt;moeda&gt;     Real   &lt;/moeda&gt; &lt;/pedido&gt;</pre>	<pre>&lt;xsl:template match="/"&gt;   &lt;xsl:for-each select="pedido"&gt;     &lt;pedido&gt;       &lt;moeda&gt;         &lt;xsl:value-of select="@moeda"&gt;       &lt;/moeda&gt;     &lt;/pedido&gt;   &lt;/xsl:for-each&gt; &lt;/xsl:template&gt;</pre>

Na Tabela 2.14 é mostrado o código XSLT necessário para converter o formato da DTD da companhia “A” no formato da DTD da companhia “B”. Observe que o namespace xsl é usado, isso mostra que a transformação é feita com construtores XSL específicos. O documento de transformação seleciona o elemento "pedido" no documento fonte. Então, inicia um novo elemento "pedido" com "moeda" como elemento filho e insere o valor do atributo "moeda" no conteúdo do elemento "moeda". Isto é feito para todos os elementos "pedido" até que todo documento esteja convertido.

Como pode-se ver, o XSLT é a própria XML. Logo, todo documento XSLT é também um documento XML. E todas as ferramentas para criação e manutenção de documentos XML são aplicáveis a documentos XSLT, porém estas ferramentas somente validarão sintaxes específicas do XML e não aquelas inerentes ao XSLT, fornecendo algum nível de entendimento.

A XSLT tem evoluído e adquirido um papel importante na especificação da família XML, a qual fornece uma gramática geral para definição de dados em documentos. Mas como alguns dados trafegam dentro e fora da organização, diferentes leitores provavelmente irão querer os documentos em diferentes formatos. A XSLT fornece mecanismos para suportar o fluxo de dados customizados. Isto, realmente, é uma contribuição para as trocas de informações em formatos comuns para todas as aplicações.

#### **2.2.6.5. XML Schema**

Pode-se perceber que, de todas as especificações da família XML, a DTD é a única a não usar sintaxe de um documento XML. Isto é uma desvantagem, já que seria necessária outra ferramenta somente para tratar um documento DTD. Aplicações e ferramentas devem ser capazes de processar tanto documento XML quanto a sintaxe DTD. Fornecedores e desenvolvedores querem usar a XML como um formato de dados universal para trocas de dados. Mas, a incapacidade de identificar automaticamente, se um valor é do tipo inteiro, ponto flutuante ou string impõe um fardo maior sobre o programador da aplicação. Pensando nisso criou-se a XML Schema.

A Tabela 2.15 demonstra expressividades impossíveis com a DTD. O primeiro quadro mostra como definir o elemento "Pedido" sendo um conjunto de 1 a 25 elementos "itempedido" onde declara precisamente mais restrições de cardinalidade do que as DTD permitem. Tais restrições serão fundamentais se, por exemplo, o sistema de preenchimento de pedidos estiver restrito manipular somente o máximo de 25 itens.

O segundo quadro mostra como criar um elemento "Estado" e como a sigla dos Estados é escrita com 2 caracteres para definir um tipo de dados SiglaUF e referenciá-la na declaração do elemento "Estado". Tanto se define que o tipo de dados é string quanto se define que deve ser formado por dois caracteres. XML Schema inclui padrões de especificação similares à sintaxe das PICTURES do COBOL e expressões condicionais para restrições do formato do conteúdo.

**Tabela 2.15 – Exemplos de XML Schema.**

Mais Restrição de cardinalidade	Mais restrições de formato
<pre>&lt;elementType name=" Pedido"&gt;   &lt;all minOccurs="1" maxOccurs="25"&gt;     &lt;elementType name=" itemdepedido "&gt;       &lt;any/&gt;    &lt;/elementType&gt;     &lt;/all&gt;   &lt;/elementType&gt;</pre>	<pre>&lt;dataType name="SiglaUF"&gt;   &lt;basetype="string"/&gt;   &lt;lexicalRepresentation&gt;     &lt;lexical&gt;AA&lt;/lexical&gt;   &lt;/lexicalRepresentation&gt; &lt;/dataType&gt; &lt;elementType name="Estado"&gt;   &lt;dataTypeRef name="SiglaUF"&gt; &lt;/elementType&gt;</pre>

A XML Schema tem várias outras características úteis, tais como a capacidade de criar blocos recursivos de elementos ou atributos. Com a XML Schema, desenvolver aplicações para troca de informações será muito mais fácil.

### 2.2.7. XQuery

A XQuery[14] é uma linguagem funcional que permite que vários tipos de expressões sejam aninhados. Também é uma linguagem fortemente tipada na qual os operandos de várias expressões, operadores e funções devem conformar-se aos tipos designados. Assim como a XML, a XQuery é *case-sensitive*. Todas as palavras-chave na XQuery usam caracteres em minúsculo. Esta linguagem é ideal para consultas, pois é incapaz de suportar atualizações ou inserções.

A versão 1.0 da XQuery é uma extensão da versão 2.0 da linguagem Path. Qualquer expressão, que seja sintaticamente válida e seja executada com sucesso em ambas as versões, retornará o mesmo resultado. A sua base é a expressão e sua linguagem provê de vários tipos de expressões que podem ser construídas com palavras-chave, símbolos e operandos. Em geral, os operandos de uma expressão são outras expressões.

A XQuery é uma linguagem padrão da XML para acessar e trabalhar com sistemas de armazenamento de informações distribuídas, inclusive repositórios de conteúdo, documentos, bancos de dados relacionais, e repositórios de objeto com sintaxe simples e concisa. A XQuery é derivada de uma linguagem de consulta XML chamada Quilt, que incorporou características de várias outras linguagens. Ela utiliza a estrutura de XML de forma inteligente e expressa consultas por todos estes tipos de dados, se fisicamente armazenados em XML ou vistos como XML por middleware. Devido à grande quantidade de informação armazenada, trocada e apresentada com a

utilização da XML, esta linguagem de consulta ganhou notoriedade. Uma das grandes forças da XML é a flexibilidade de representar diferentes tipos de informações de diversas fontes [15,16].

A base da linguagem XQuery consiste da expressão FLWR que significa as cláusulas FOR-LET-WHERE-RETURN. Esta expressão forma o esqueleto de uma sentença de consulta e é análogo à sentença SELECT-FROM-WHERE na linguagem SQL. Abaixo, cada cláusula é descrita separadamente:

#### 2.2.7.1. Cláusula FOR

A cláusula FOR usa expressões XPath para ligar os valores de uma ou mais variáveis. Cada expressão retorna um conjunto de nós, e a cláusula FOR gera uma lista ordenada de *tuplas*. Cada uma contém um valor para cada variável de ligação. A ordem é determinada pela ordem dos elementos de ligação no documento de entrada e a primeira variável de ligação precede, seguida pela segunda. Afinal, preservar a mesma ordem como no documento original é importante em muitas aplicações. Todavia, se qualquer das expressões usadas em uma cláusula FOR estiver desordenada, então, as *tuplas* geradas pela sequência FOR-LET também estarão desordenadas. Quando um nó está ligado à uma variável, seus nós filhos o acompanham. Por exemplo, a consulta seguinte retorna todos os conteúdos dos elementos *Rodovia* do documento *rodovias.gml*, como mostrado na Tabela 2.16.

**Tabela 2.16 – Exemplo da Cláusula FOR.**

Cláusula FOR
FOR \$rod IN document (' rodovias.gml') //Rodovia RETURN \$rod

#### 2.2.7.2. Cláusula LET

A cláusula inicial FOR, em uma expressão FLWR, pode ser seguida por uma ou mais cláusulas LET e cláusulas FOR adicionais que forneçam ligação para variáveis adicionais. A cláusula LET liga uma variável ao valor de uma expressão, pois a principal diferença entre uma cláusula FOR e LET é que as variáveis ligadas pela cláusula FOR consistem de um único nó com os seus filhos. Por outro lado, as variáveis ligadas por uma cláusula LET representam coleções de nós (por exemplo; uma floresta).



Freqüentemente, uma cláusula LET é usada quando precisamos armazenar um grupo de valores, e estes podem ser usados com alguma função agregada (por exemplo avg()). A consulta seguinte gera a distância média de rodovias, ilustrando a utilização da cláusula LET (Tabela 2.17).

**Tabela 2.17 – Exemplo da Cláusula LET.**

<b>Cláusula LET</b>
LET \$rod := document (“rodovias.gml”) //Rodovia RETURN <distanciamedia> avg (\$rod/distancia) </distanciamedia>

Quando se deseja aplicar uma função agregada a um conjunto de grupos, as cláusulas FOR e LET podem ser combinadas para gerar estes grupos. Por exemplo: a consulta seguinte, apresentada na Tabela 2.18, gera a distância média de rodovias em cada município.

**Tabela 2.18 – Exemplo das Cláusulas FOR e LET.**

<b>Cláusulas FOR e LET</b>
FOR \$mun IN document (“rodovias.gml”) //Município LET \$med := avg(document (“rodovias.gml”) [Município = \$mun]/Rodovia/distancia RETURN     <Município> <gml:name> \$mun/text() </gml:name> <distanciamedia> \$med </distanciamedia> </Município>

### 2.2.7.3. Cláusula WHERE

A cláusula WHERE é usada para selecionar as tuplas geradas pelas cláusulas FOR e LET. As tuplas que satisfazem todas as condições da cláusula WHERE estão disponíveis na cláusula RETURN. Por exemplo, a consulta seguinte apresentada na Tabela 2.19 retorna aqueles municípios onde o número de rodovias é maior que sete.

**Tabela 2.19 – Exemplo da Cláusula WHERE.**

<b>Cláusula WHERE</b>
FOR \$mun IN document (“rodovias.gml”) //Município LET \$rod := document (“rodovias.gml”) [Município = \$mun]/Rodovia WHERE count(\$rod) > 7 RETURN \$s <Município> <gml:name> \$sd/text() </gml:name> </Município>

#### 2.2.7.4. Cláusula RETURN

O conjunto resultado é construído na cláusula RETURN. Como já foi mostrado nos exemplos anteriores, o resultado pode ser um valor primitivo, um nó, ou uma árvore de nós.

A sintaxe básica da linguagem XQuery é mostrada na Tabela 2.20:

**Tabela 2.20 – Sintaxe básica da linguagem XQuery.**

Sintaxe Básica
FOR var IN expressão ( igual a expressão XPath 2.0) LET var:= expressão [[FOR var IN expressão] [LET var:= expressão] ...] WHERE condição [[AND   OR   NOT] condição] RETURN elementos construtores (usando as variáveis anteriores)

A Expressão FOR na cláusula FOR é do tipo <variável> 'IN' 'DISTINCT'? Em que a expressão é uma constante, uma variável, um elemento construtor, uma função, uma expressão XPath ou outra expressão FLWR. Deste modo, constroem-se subconsultas (sub-queries) usando expressões FLWR aninhadas.

#### 2.2.7.5. Modelo Relacional vs. XML

O uso da XML para armazenamento de dados tem se tornado freqüente e um estudo criterioso deve ser feito na sua utilização. Na Tabela 2.21 correlacionaram-se algumas características dos dois modelos para avaliação.

**Tabela 2.21 – Comparação entre o modelo Relacional e o da XML.**

Relacional	XML
<ul style="list-style-type: none"><li>• Banco relacional contém tabelas.</li><li>• Tabela relacional contém registros com mesmo esquema.</li><li>• Registro relacional é lista de valores.</li><li>• SQL query retorna conjunto não-ordenado de registros.</li></ul>	<ul style="list-style-type: none"><li>• Bancos XML contêm coleções.</li><li>• Coleção contém documentos XML com mesmo DTD.</li><li>• Documento XML é uma árvore de nós.</li><li>• XML Query retorna uma seqüência não-ordenada de nós.</li></ul>

### 2.3. A Linguagem RDF

A Resource Description Framework (RDF) [17] é um padrão W3C que foi desenhado para representar informações sobre recursos na *Web*, isto é, representar metadados sobre recursos e, com isso, habilitar aplicações e agentes inteligentes a utilizar esses recursos e, então, suportar a interoperabilidade entre eles.

É praticamente impossível para um desenvolvedor manter uma aplicação reescrevendo constantemente adequações ao sistema. O problema está na incapacidade das aplicações automaticamente “entenderem” o significado do conteúdo, isto é, ter maior semântica.

Os metadados sobre os recursos tanto podem ser informações descritivas, relativamente simples, tais como o título ou autor do recurso como relacionamentos mais complexos entre os recursos [17a][18].

Recursos são identificados por seus URI's [19], que são objetos disponíveis na *Web*, tais como: documentos eletrônicos, imagens, fonte de informações com um propósito consistente, serviços e coleções de recursos. Um recurso não é necessariamente acessível via Internet, como por exemplo: seres humanos, corporações e livros em biblioteca. Além destes, incluem-se os conceitos abstratos, tais como: operadores e operandos em equações matemáticas, tipos de relacionamento ou ainda valores numéricos.

Uma URI é uma sequência compacta de caracteres que identificam um recurso físico ou abstrato na *Web*. É iniciada pelo esquema, o qual, por sua vez, identifica de que tipo é o recurso.

A uniformidade permite que diferentes identificadores de recursos sejam utilizados no mesmo contexto e, ainda que novos identificadores sejam criados sem interferir nos já existentes. Propicia a utilização dos recursos de diferentes maneiras e em diferentes contextos. Precisa-se da URI para distinguir um recurso de todos os outros recursos, esta classifica-se como uma indicação de localização ou nome. Uma URL é uma URI que além de identificar o recurso, informa onde e como utilizá-lo. O URN é uma URI que identifica o recurso pelo nome em um namespace particular. Um exemplo do uso da URI é mostrado na Tabela 2.22.

**Tabela 2.22 – Exemplos de URI.**

URIs
ftp://ftp.is.co.za/rfc/rfc1808.txt http://www.ietf.org/rfc/rfc2396.txt ldap://[2001:db8::7]c=GB?objectClass=one mailto:ricardoleal@hotmail.com news:comp.infosystems.www.servers.unix tel:+1-816-555-1212 telnet://192.0.2.16:80/ urn:oasis:names:specification:docbook:dtd:xml:4.1.2 urn:ISBN:0-3954-36341-1

A RDF é usada para criar modelos de metadados que podem ser entendidos por um agente, enquanto a linguagem XML é usada para armazenar e transportar essas informações em arquivo, mas não é exclusiva, pois outros mecanismos podem ser usados.

Imaginando a *Web* como uma rede de recursos interconectados através de seus relacionamentos, pessoas seriam “*nós*” nessa rede e estariam ligadas umas as outras através de relacionamentos, tais como: parentesco, irmandade, casamento etc. Um outro tipo de relacionamento é a propriedade física de um nó como a idade ou o nome de uma pessoa. A qualquer momento, é viável a introdução de um novo relacionamento a um nó existente. Em RDF, um nó consiste em dois tipos mais gerais: recurso ou literal. Um literal é um valor de uma propriedade como o nome “Ricardo” ou o número “3”, pois retrata uma string ou um número. Contudo, os recursos são identificados por sua URI, que elimina as ambigüidades.

### **2.3.1. Triplas RDF**

Tripla é a combinação de dois recursos através de um relacionamento ou de um recurso e um literal. Se uma tripla não é especificada no grafo, não significa que ela não exista, mas que o valor é desconhecido ou será fornecido mais tarde.

O sistema infere uma tripla através de outras, se regras forem armazenadas. A tripla é formada por sujeito -> predicado -> objeto ou alvo. Para garantir a unicidade do predicado, a URI deve ser usada. Pode-se ter múltiplas triplas com o mesmo sujeito e predicado para objetos diferentes. A ordem das triplas é irrelevante.

### 2.3.2. Tipos RDF

Algumas vezes, é necessário identificar-se que tipo de objeto é o recurso, assim como, em sistemas orientados a objetos, a RDF usa as classes para esse propósito. Todo recurso é de um tipo preciso. Assim, por exemplo, o recurso “Ricardo” é do tipo pessoa e, também, pode-se criar tipos próprios novos.

A RDF tem um número de tipos predefinidos para representar lista de objetos. As vezes, é útil colocar valores sobre uma determinada ordem. Faz-se isso numerando o próprio predicado, como por exemplo: nome1, nome2 e assim por diante.

Em RDF, um predicado que tem um número precedido com um sublinhado é usado como um item de uma lista. Por exemplo: rdf:\_1 é usado para indicar o primeiro item da lista, rdf:\_2 o segundo e assim por diante. Porém não existem restrições que impeçam a repetição de identificadores da lista, nem que coloque-os na ordem correta.

Vários tipos de listas são disponíveis:

- rdf:Seq : uma lista ordenada
- rdf:Bag : uma lista desordenada;
- rdf:Alt : uma lista de valores alternados onde somente um valor é esperado usar.

A RDF permite que listas de triplas de uma fonte sejam combinadas com triplas de outras fontes, desde que a ordem não seja problema.

As aplicações beneficiam-se desse modelo simplista incluindo *sites* de pesquisa, que agregam conteúdo de diferentes fontes.

## 2.4. A Linguagem GML

A GML (Geographic Markup Language)[4] é uma linguagem para estruturar a informação geoespacial com o objetivo de conseguir a interoperabilidade de uma infraestrutura de dados geoespaciais [20,21]. Foi especificada pelo Open Geospatial Consortium (OGC) e fornece um meio para distribuição da informação geográfica.

No aspecto da modelagem, a GML é uma solução para a codificação de Feições Simples OGC[22] utilizando a XML. O modelo destas representa uma simplificação do modelo mais geral descrito na especificação OGC é resultado do desenvolvimento de várias especificações de implementação. São duas as principais simplificações:

- As feições só têm propriedades simples (booleanos, inteiros, pontos flutuantes, strings) ou propriedades geométricas.
- As geometrias só são definidas em um SRS (*Spatial Reference System*) bidimensional utilizando interpolação linear entre coordenadas.

Várias consequências surgem destas simplificações. Uma, particularmente importante, é que feições simples só suportam "dados vetoriais". Pretende-se reparar algumas destas limitações em versões futuras da GML.

Para desenhar um mapa com GML, é necessário transformar o documento GML em um formato gráfico. Isto pode ser feito através de manipulação direta. Por exemplo, um applet java usando DOM (Document Object Model) ou ainda, transformando-as em formatos gráficos, tais como: SVG , VML ou VRML , os quais também são codificados em XML.

Uma Feição Simples separa as propriedades geoespaciais das geométricas. As propriedades geométricas são utilizadas para visualizar os elementos do mapa, pois independem de qualquer especificação gráfica, sem informação de como aparecem as feições codificadas. Assim, a representação visual de uma estrutura GML depende do uso de recursos gráficos adicionais.

Vale considerar que em algumas aplicações não haverá exibição de resultados gráficos, pois dados geográficos podem, simplesmente, ser dirigidos a um modelo matemático para processamento, como por exemplo, um modelo de previsão de inundação. Todavia, a saída deste modelo matemático pode ser expressa em GML.

#### 2.4.1. OGC

O Open Geospatial Consortium (OGC) é um consórcio de empresas que atualmente é responsável pela criação de novas padronizações técnicas e comerciais para garantir interoperabilidade em Geomática.

O OGC foi fundado em 1994 devido ao reconhecimento do problema da falta de interoperabilidade em SIG. Parte dos membros OGC tem uma visão positiva de uma infra-estrutura de informação global, na qual dados geográficos e recursos de geomática devem ser distribuídos gratuitamente e estão completamente integrados com as mais recentes tecnologias de computação distribuída. Tais dados, devem ser acessíveis a todos, a fim de habilitar uma grande variedade de atividades que estão, atualmente, fora do domínio da Geomática, e abrindo novos mercados, oferecer oportunidade a novos tipos de negócios e benefícios para o público.

Fornecedores de software de geoprocessamento, software de banco de dados, software de visualização, computadores, integradores de sistemas, companhias de telecomunicações, universidades, provedores de informação, e agências federais se uniram num consórcio para criar uma especificação de software e novas estratégias empresariais a fim de resolver estes problemas e realizar estes potenciais.

A especificação de software da OGC é um documento OGC com a descrição de um arcabouço para software que permita acessar dados geográficos e apresente recursos de geoprocessamento distribuídos. A mesma visa prover aos desenvolvedores de software, do mundo todo, um modelo de interface comum e detalhado, para que desenvolvam softwares capazes de interoperar com outros softwares no mesmo formato. Esta encontra-se em vias de se tornar padrão internacional de tecnologia de informação, onde fornecedores e outros participantes da OGC a implementarão, amplamente, em produtos de software comercial, projetos de integração comercial, centros de dados governamentais e pesquisas acadêmicas. Este processo deve fazer da Especificação OGC, um padrão de interoperabilidade da Geomática.

Por estar firmemente unido ao software comercial e permitir transações de geoprocessamento e dados compartilhados em um formato básico e amplo, o OGC, provavelmente, será a base para a interoperabilidade de geoprocessamento por um longo tempo no futuro.

A proposta da Especificação abstrata é criar e documentar um modelo conceitual suficiente o bastante para permitir a criação de Especificações de Implementação. A Especificação Abstrata consiste de dois modelos [23]: um é o Modelo Essencial, que representa fatos do mundo real. O segundo, é o Modelo Abstrato, que representa a descrição de como o SIG irá funcionar.

#### **2.4.2. Feições Geográficas Simples**

Primeiramente, deve-se entender como a GML modela o mundo. A chave deste entendimento está baseado no Resumo da Especificação OGC [22] que define uma feição geográfica como: *“Uma feição é uma abstração de um fenômeno do mundo real; esta é geográfica se está associada a uma posição relativa à Terra”*.

Dessa forma, uma representação digital do mundo real compreende-se como um conjunto de feições. O estado de uma feição é definido por um conjunto de propriedades, onde cada propriedade representada por uma *tupla* {nome, tipo, valor}. O conjunto de propriedades que uma feição apresenta, juntando seus nomes e tipos, é determinado por seu tipo de feição. Feições geográficas são aquelas cujo valor da propriedade possa ser uma geometria. Uma coleção de feições tanto é um conjunto de feições como é também considerada recursivamente uma feição. Por conseguinte, uma coleção de feição tem um tipo e pode ter propriedades próprias, além das feições que contém.

A GML define o termo "feições simples" como feições cujas propriedades da geometria estão restritas a propriedades geométricas simples (por exemplo, coordenadas são definidas em duas dimensões e o caminho da curva entre duas ou mais coordenadas assume-se que é interpolado linearmente). Este termo foi cunhado para descrever a funcionalidade definida no conjunto de especificações de implementação da OGC.

A GML obedece ao modelo de geometria definido nas especificações OGC. Por exemplo, o tradicional 0, 1 e geometrias 2D definidas em um Sistema de Referência Espacial bidimensional (*Spatial Reference System - SRS*) são representados por pontos, linhas e polígonos. Além do modelo geométrico para feições simples, também permite geometrias que são coleções de outras geometrias (homogêneas: multipontos, multilinhas e multipolígonos, ou heterogênea: coleção de geometria). Em todos os casos, a geometria "pai" é responsável para indicar em qual SRS as medidas foram feitas.



Considere o exemplo em que se deseja construir uma representação digital da Rodovia Presidente Dutra. A rodovia poderia ser representada como uma coleção de feições onde as feições individuais representam objetos tais como: estradas, pontes, limites e imóveis.

A classificação de um fenômeno do mundo real determina os tipos de feições que precisam ser definidos. A escolha da classificação está relacionada à tarefa para a qual a representação digital será colocada ao final.

A feição "Casa" tem uma propriedade chamada "nome" cujo valor deve ser do tipo string. É comum referi-la como uma propriedade tipada. Assim, no exemplo anterior, a feição "Casa" é dita como uma propriedade do tipo string chamada "nome". Da mesma forma que a feição "Estrada" pode ter uma propriedade string chamada "classificação" e uma propriedade do tipo inteiro chamada "Numero".

As feições exigidas para modelar a rodovia podem ter tanto propriedades geométricas quanto simples. Assim como outras propriedades, as geométricas, devem ser nomeadas. Dessa forma, a feição "Posto" pode ter uma propriedade geométrica chamada "PontoCentral" e a feição "Estrada", chamada "GeometriaLinear". É possível ser mais preciso sobre o tipo de geometria usada com um valor. Portanto, no exemplo "Estrada", a propriedade geométrica pode ser especializada para ser uma propriedade cadeia de linhas (linestring). Da mesma maneira que é comum ter múltiplas propriedades simples definidas em uma única feição, as quais, por sua vez, poder ter múltiplas propriedades geométricas. Por exemplo, a feição "Ponte" abrangeria as propriedades "Altura" e "Extensão" do tipo inteiro.

Finalmente, todo o modelo da Rodovia Presidente Dutra expressa-se como uma única coleção de feições. Esta pode ter uma feição chamada "ModeloRodovia" e denominada uma propriedade string chamada "DataDoModelo", com a informação da data quando foi construído, uma propriedade geométrica, "LimitadaPor", ofereceria a extensão que abrange o modelo.

### **2.4.3. Modelo de Objetos**

É prática comum na comunidade de Informação Geoespacial referir-se às propriedades de uma feição como atributos. Porém, com a finalidade de evitar confusão com os atributos da XML, este documento as refere como propriedades. O modelo de feições simples está demonstrado na Figura 2.4.

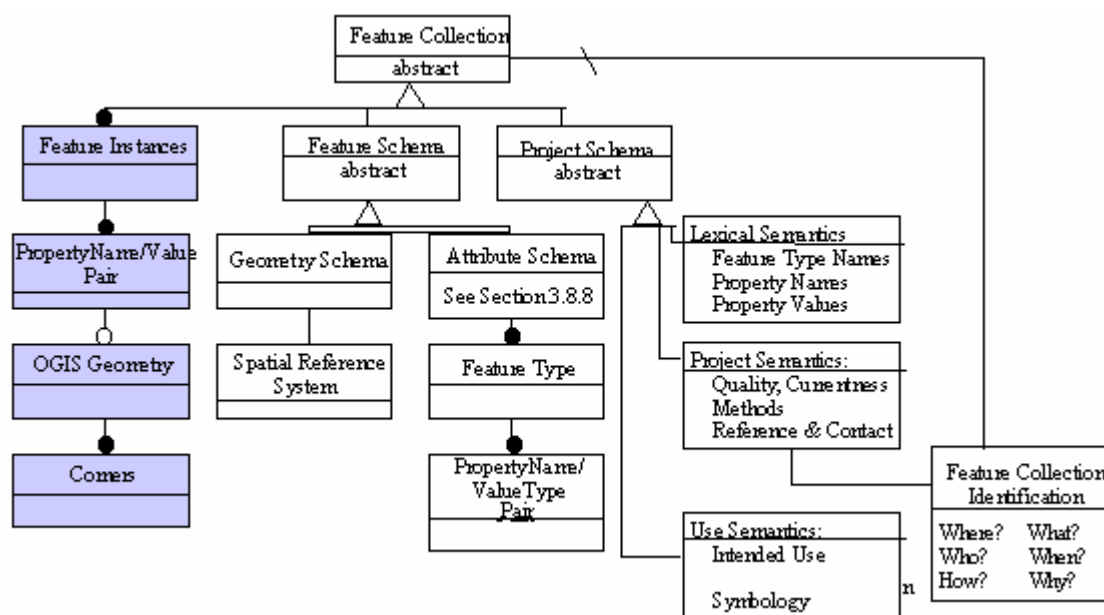


Figura 2.4 – Modelo<sup>5</sup> de feições usado na OGC.

O modelo de geometria para Feições Simples conforme a Figura 2.5. Este modelo é comum a várias especificações e documentos.

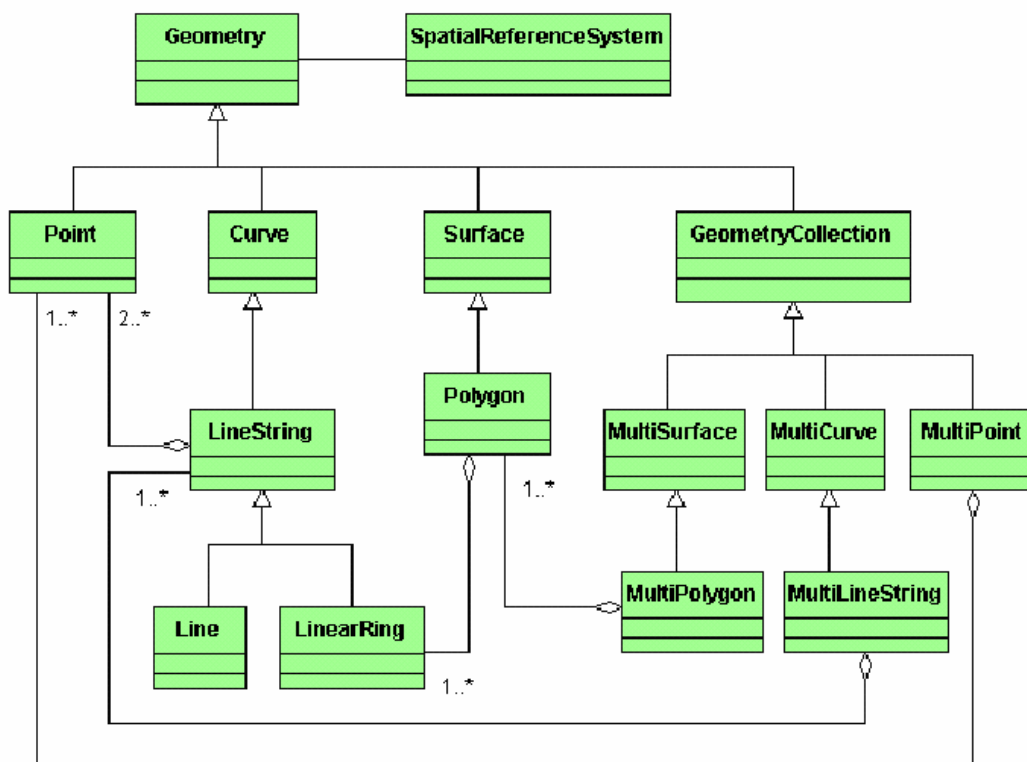


Figura 2.5 – Modelo de Geometria<sup>6</sup> para Feições Simples da OGC.

<sup>5</sup> Figura retirada da Especificação da Implementação GML versão 2.1.1, p.5

#### 2.4.4. Usando a GML

A GML é uma solução para a codificação das Feições Simples OGC em XML. Esta linguagem, presumivelmente, atrairá várias classes de usuários que desejarão empregar uma variedade de tecnologias XML, voltada para três perfis:

- **Perfil 1:** para usuários que desejam uma solução baseada em DTD pura e não estão preparados para desenvolver uma DTD específica para sua aplicação ou ainda desejam que dados geoespaciais sejam trocados com um conjunto fixo de DTD. Este perfil requer o uso das DTD de Feições e Geometria GML.
- **Perfil 2:** para usuários que desejam uma solução baseada em DTD pura, mas estão preparados para desenvolver uma DTD específica para sua aplicação ou ainda estão preparados para aceitar dados codificados em uma DTD referenciada. Este perfil exige que o usuário crie uma DTD com feições específicas para sua aplicação que use a DTD de Geometria da GML.
- **Perfil 3:** para usuários que estejam preparados para fazer uso de RDF e Esquemas RDF. Estes usuários exigirão um controle mais forte da estrutura de tipos geoespaciais (por exemplo, eles devem relacionar um nome de tipo a uma definição de esquema atual). Este perfil exige que o usuário crie uma definição de Esquema RDF específico à aplicação que use a definição GML. De outra forma, usuários do Perfil 3 utilizariam DTD que são derivadas, de alguma maneira, de um Esquema RDF ou que localize seus elementos em tipos definidos em um Esquema RDF associado.

---

<sup>6</sup> Figura retirada da Especificação da Implementação GML versão 2.1.1, p.6

Resumo dos três perfis Figura 2.6.

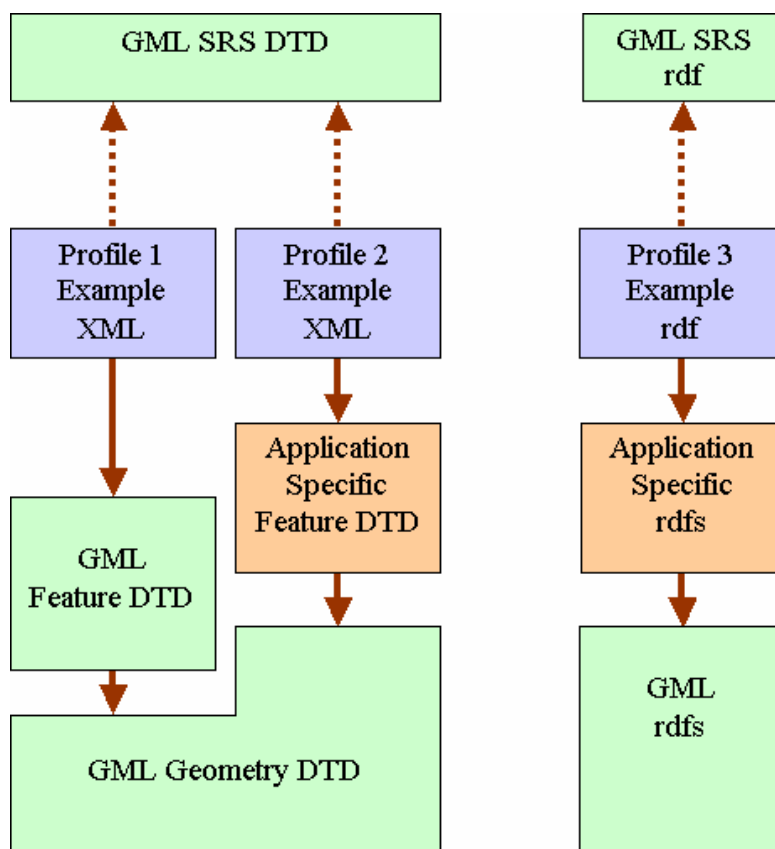


Figura 2.6 – Os três perfis da GML <sup>7</sup>.

As setas contínuas indicam uma referência, enquanto as pontilhadas uma referência opcional. As caixas verdes são definições de GML predefinidas. As caixas bege representam definições específicas da aplicação construídas de acordo com as regras de perfil GML apropriadas. Finalmente as caixas roxas retratam informação geoespacial codificada com o uso perfil GML apropriado.

A GML versão 1.0 está de acordo com a XML versão 1.0 e, por isso, usa DTD em lugar de XML Schemas, diferente das versões 2.1.1 e 3.1.0.

A GML também foi desenvolvida para ser consistente com o modelo e a sintaxe do formato RDF da W3C. A codificação geométrica da GML é utilizada para descrever as propriedades geométricas de qualquer recurso de RDF, assim como sua extensão, cobertura ou localização. Isto permite o uso da GML em grande variedade de aplicações as quais não sejam inerentemente espaciais.

<sup>7</sup> Figura retirada da Especificação da Implementação GML versão 1.0, p.8

A GML ainda é compatível com o Namespaces. Nos Perfis 2 e 3 da GML, usa-se o Namespaces XML para distinguir as definições de feições geográficas e propriedades definidas no domínio da aplicação daquelas predefinidas pela GML OGC.

#### **2.4.4.1. Geografia e Gráficos**

Feições simples propõem-se a descrever a geografia das entidades do mundo real, embora sua codificação não se preocupe com a visualização de feições geográficas, como em desenhos de mapas. A GML é distinta e independente de qualquer outra especificação gráfica. Nesta não consta qualquer informação sobre como as feições codificadas aparecem. Para desenhar um mapa com o uso da GML é necessário transformá-la em um formato gráfico através de manipulação direta ou, preferivelmente, através da transformação em elementos de gráficos codificados em XML como SVG[24], VML[25], ou VRML[26]. Esta transformação realiza-se em qualquer momento na cadeia de processos entre o armazenamento e o dispositivo de visualização. Transformá-la nos formatos citados é fortemente recomendado para visualização de dados.

Um mesmo documento GML tem a capacidade de gerar muitas representações simbólicas diferentes. Estas incluiriam formatos gráficos e simbolizações diferentes, ou seja, um mesmo documento GML acarretará múltiplos tipos de mapas. Logo, pretende-se que a GML suporte tanto o armazenamento quanto o transporte de dados. Implementadores decidem armazenar informação geoespacial na GML, ou converter de algum outro formato de armazenamento e sobre demanda usar GML somente para o transporte de dados.

Em algumas aplicações não haverá exibição de qualquer dado gráfico, conforme mencionado anteriormente, dados geográficos podem simplesmente ser dirigidos a um modelo numérico para processamento, com por exemplo, um modelo de previsão de inundação. A saída deste modelo numérico pode também ser expressa na GML.

As coordenadas de pontos em um documento GML são especificadas a partir do Sistema de Referência Espacial cuja descrição encontra-se codificada na GML. É possível um servidor de dados fornecer dados codificados na GML, mas não fornecer a descrição do Sistema de Referência Espacial, contanto que uma referência nomeada para cada descrição seja incluída. As descrições do Sistema de Referência Espacial estão sempre conectadas aos dados geográficos por meio de uma referência nomeada.

#### 2.4.4.2. Geometria

Esta seção descreve como a GML codifica geometria em XML, além de apresentar a DTD de geometria. Isto é explicitamente usado por GML Perfis 1 e 2. Porém, a XML codificada também é consistente com a definição de esquemas RDF de Geometria usada pelo perfil 3 da GML.

De acordo com o modelo OGC de Feições Simples, a GML fornece elementos de geometria que correspondem às seguintes Classes de Geometria: Point, LineString, LinearRing, Polygon, MultiPoint, MultiLineString, MultiPolygon, GeometryCollection. Além de fornecer um elemento "coordinates" para codificar coordenadas, e um elemento "Box" para definir áreas.

As seções seguintes descrevem a codificação de cada geometria em detalhes.

#### 2.4.4.3. O Elemento Coordinates (coordenadas)

Uma lista de coordenadas é uma lista simples de tuplas de coordenadas. Os separadores usados para analisar gramaticalmente a lista de coordenadas são codificados como atributos da marcação <coordinates>. No exemplo da Tabela 2.23, as coordenadas em um tupla estão separadas através de vírgulas e as tuplas sucessivas no elemento <coordinates> estão separadas através de espaço em branco. Uma lista de coordenada não é uma geometria, e sim um repositório de coordenadas. Todas as tuplas no string devem ter a mesma dimensão. Uma lista de coordenada é determinada pela gramática apresentada na Tabela 2.23.

**Tabela 2.23 – Gramática de tuplas na GML.**

<b>Tuplas na GML</b>
<code>&lt;decimal&gt;: := ' . '</code> <code>&lt;D&gt;:= [0-9]</code> <code>&lt;cs&gt;: := " , "</code> <code>&lt;ts&gt;: := whitespace (veja XML 1.0 [XML])</code> <code>&lt;coordinates&gt;: := ' - ' &lt;D&gt;+( &lt;decimal&gt;&lt;D&gt;+ )?</code> <code>&lt;ctuple&gt;::= &lt;ctuple&gt; &lt;coordinates&gt;&lt;cs&gt;&lt;ctuple&gt;</code> <code>&lt;coordinatelist&gt;::= &lt;coordinatelist&gt; &lt;ctuple&gt;&lt;ts&gt;&lt;coordinatelist&gt;</code>

Observe que o valor de decimal, cs, e de ts são determinados pela codificação GML <coordinates> .

A gramática está ilustrada para valores predefinidos de decimal, cs e ts.

Para achar as coordenadas de qualquer instância de classe geométrica, introduziu-se a propriedade coordenada. Pensou-se nisto como uma função de classe geométrica na instância que devolve as coordenadas como uma lista de coordenada.

A propriedade coordenada tem o fragmento de DTD associado:

```
<!ELEMENT coordinates (#PCDATA)>
<!ATTLIST coordinates
    decimal      CDATA #IMPLIED
    cs           CDATA #IMPLIED
    ts           CDATA #IMPLIED>
```

Nota-se que o valor de coordenada é determinado pelo elemento `<coordinates>= ' - ' <D>+(<decimal><D>+)?`, conseqüentemente possibilita a codificação das coordenadas como 1.45 ou 1,45. dependendo dos valores que contidos nos atributos de `<coordinates>`. Observa-se na Tabela 2.24 que a falta para decimal é ' ', para cs é ' ' e para ts é whitespace.

**Tabela 2.24 – Exemplo do elemento Coordinates.**

<b>Elemento coordinates</b>
<pre>&lt;coordinates decimal = "." cs = " " ts = "whitespace"&gt;   1.03,2.167 4.167,2.34 4.87,3.0 1.06,2.3 &lt;/coordinates&gt;</pre>

#### **2.4.4.4. Demais elementos geométricos**

Uma característica semelhante a todos os elementos é a apresentação de atributos identificadores (ID) que são opcionais e serve para identificar cada instância de objeto criado. Uma outra característica comum a quase todos os objetos, exceto *linestring*, é o atributo *SRSname* que determina o Sistema de Referência no qual suas coordenadas estão medidas. Uma geometria tem de especificar um SRS, pois este é referenciado através de nome. Porém, isto é definido para ser opcional, porque ocorre para permitir que o elemento *Point* esteja contido em outros elementos que teriam um SRS.

Considerações como estas se aplicam aos outros elementos geométricos. Na Tabela 2.25 visualizam-se os elementos geométricos da GML e sua respectiva descrição do uso e características.

**Tabela 2.25 – Elementos geométricos da GML.**

<b>Elemento</b>	<b>Descrição</b>
<i>Point</i>	Codifica instâncias da classe geométrica ponto. Cada Elemento inclui um único <i>coordinates</i> , contendo uma e somente uma tupla.
<i>Box</i>	Codifica instâncias de áreas. Cada Elemento inclui um único <i>coordinates</i> , que contém duas tuplas. A primeira destas é constituída do valor mínimo medidos juntos a todos os eixos, e a segunda é constituída do valor máximo medido ao longo de todos os eixos. Este elemento tem um <i>srsName</i> obrigatório, pois não está contido em outras classes geométricas.
<i>LineString</i>	É um caminho linear de trechos conhecidos. O caminho é definido por uma lista de <i>coordinates</i> e assume-se que eles estão conectados através de segmentos de retas. É fechado se a primeira e última coordenada coincidirem. Exigem-se, pelo menos, duas coordenadas.
<i>LinearRing</i>	É um caminho simples linear fechado e definido por uma lista de <i>coordinates</i> , assume-se que estejam conectados através de segmentos de retas. A primeira e a última coordenadas devem ser coincidentes. No mínimo, exigem-se quatro coordenadas (três para definir um anel e a quarta a <i>coordinates</i> duplicata). Considerando que um <i>LinearRing</i> é usado na construção de Polígonos que definem o seu próprio SRS é desnecessário definir um SRS próprio.
<i>Polygon</i>	É uma superfície conectada. Qualquer par de pontos no polígono pode estar conectado um ao outro por um caminho. As bordas do Polígono são formadas por um conjunto de <i>LinearRing</i> . Há a distinção entre a borda exterior e a borda interior. Os <i>LinearRings</i> da borda interior nem devem ter cruzamentos e nem podem estar contido um no outro. Deve haver no máximo um elemento de borda exterior e zero ou mais de borda interior. O sentido dos <i>LinearRing</i> , horário ou anti-horário, é irrelevante.
<i>MultiPoint</i>	É uma coleção de pontos. Observe que o atributo <i>srsName</i> somente deve ocorrer no elemento <i>MultiPoint</i> , isto é, não deve aparecer nos atributos de quaisquer dos <i>Point</i> inclusos.
<i>MultiLineString</i>	É uma coleção de <i>LineString</i> . Note que o atributo <i>srsName</i> somente deve ocorrer no elemento <i>LineString</i> , isto é, não deve aparecer nos atributos de qualquer do <i>LineStrings</i> incluso.
<i>MultiPolygon</i>	É uma geometria do OGC. Perceba que o atributo <i>srsName</i> somente deve ocorrer no elemento <i>MultiPolygon</i> , isto é, não deve aparecer nos atributos de quaisquer dos <i>Polygon</i> inclusos.
<i>GeometryCollection</i>	É usado como um repositório para elementos de geometrias arbitrárias e contém quaisquer um dos outros elementos geométricos tais como: <i>Points</i> , <i>LineStrings</i> , <i>Polygon</i> , <i>MultiPoints</i> , <i>MultiLineStrings</i> , <i>MultiPolygons</i> e até mesmo outro <i>GeometryCollections</i> . O Elemento de <i>GeometryCollection</i> tem a propriedade <i>geometryMember</i> que devolve o próximo elemento Geométrico da coleção, esta propriedade pode conter quaisquer dos elementos geométricos da GML. Observe, no entanto, que o atributo <i>srsName</i> somente deve ocorrer no elemento <i>MultiPoint</i> , isto é, não deve aparecer nos atributos de quaisquer dos elementos geométricos inclusos.



## CAPÍTULO 3 - ARMAZENAMENTO EM SGBD ESPACIAL

*"A melhor maneira de se ter uma boa idéia  
é ter muitas idéias".*

Linus Pauling (químico)

Neste capítulo serão apresentados os principais conceitos sobre armazenamento da informação geográfica distribuída em banco de dados. O enfoque será dado para utilização das funcionalidades geoespaciais dos SGBD's. Para tanto, fez-se avaliação de três ferramentas: Oracle Spatial, MySQL Espacial e PostgreSQL com PostGis. Encerrará com um quadro de características relacionados a armazenamento da informação geográfica distribuída.

### 3.1. Para entender o Armazenamento

Para garantir a persistência da informação geoespacial é necessário definir a tecnologia de armazenamento usada. Segundo CÂMARA *et al*[3], as estratégias para implementação do Banco Geoespacial podem ser: dual, campos longos ou do tipo extensível (usando as funcionalidades geoespaciais dos SGBDs).

Neste capítulo fez-se avaliação apenas da terceira opção, por esta ser considerada como a melhor estratégia para escolha do armazenamento de grandes volumes de dados geoespaciais, apesar de limitar os SIG(s) a serem definidos com o modelo de Feições Simples da OGC.

O SGBD extensível é aquele que dispõe de mecanismos que permitam implementar o tratamento das componentes espaciais através de extensões ao seu ambiente, tanto no nível da linguagem de definição quanto na manipulação de dados, ou seja, na sua interface com o usuário. Um SGBD extensível tipicamente possibilita a definição de novos tipos de objetos. Essa especificação inclui a definição dos atributos dos objetos e dos métodos que atuam sobre eles. Portanto, esta facilidade vai muito além da estratégia de campos longos, pois permite capturar a semântica dos objetos.

### 3.2. Oracle Spatial

O Oracle Spatial[27] é uma opção (*option*) do servidor de banco de dados Oracle que explora as características objeto-relacional do servidor para implementar classes de objetos espaciais chamadas SDO (*Spatial Data Objects*) que permitem armazenar, consultar, manter e analisar dados espaciais.

O Oracle Spatial é composto dos seguintes itens:

- **Esquema:** conjunto de objetos lógicos que se interagem, tais como: tabelas, views, funções, procedimentos etc.
- **Mecanismo de Indexação:** estruturas auxiliares para um acesso mais rápido as estruturas de armazenamento.
- **Operadores e Funções:** extensões da linguagem SQL para executar consultas sobre áreas, junções espaciais e outras operações de análise espacial.
- **Utilitário para Administração:** ferramenta para administração e visualização da informação espacial.

O modelo objeto relacional do Oracle permite que as geometrias sejam modeladas com uma única linha e única coluna (colunas multivaloradas) evitando o excesso de junções, como ocorre no modelo puramente relacional. O modelo objeto-relacional utilizado é compatível com a implementação de tabelas de feições conforme a Especificação OGC ODBC/SQL[22] para feições geoespaciais.

O Oracle Spatial suporta tipos geométricos primitivos e tipos geométricos compostos de coleções desses tipos. Na Figura 3.1 são mostrados os tipos suportados pelo Oracle Spatial. Os tipos são bidimensionais, entretanto, o Oracle armazena tipos de três ou quatro dimensões, mas implicará na perda de funcionalidade. Assim, apenas a primeira dimensão será considerada para o índice espacial e todos os operadores espaciais serão desabilitados exceto o SDO\_FILTER.

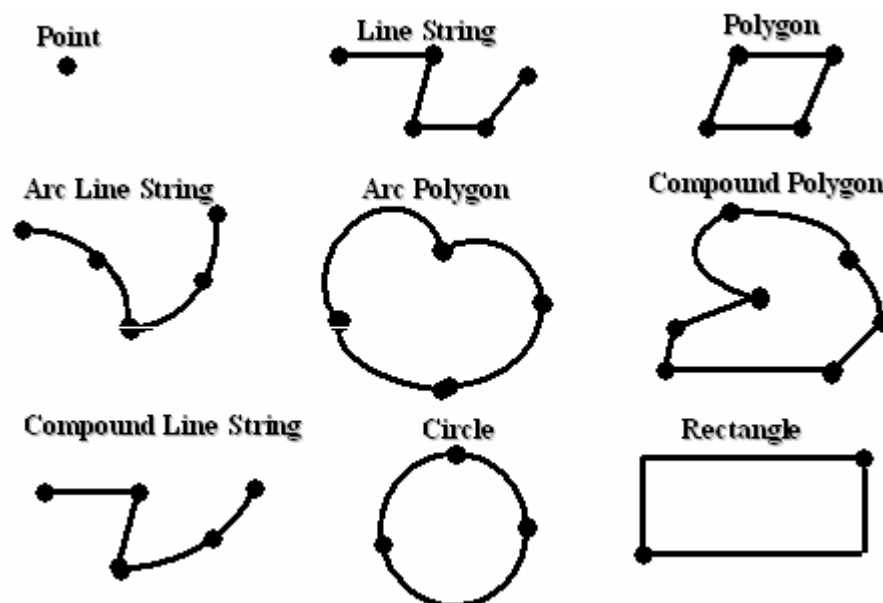


Figura 3.1 – Tipos de geométricos bidimensionais suportadas pelo Oracle Spatial [27].

O modelo de dados apresentado na Tabela 3.1 consiste de uma estrutura hierárquica de elementos, geometrias e camadas (*layer*) que correspondem à representação de dados espaciais.

Tabela 3.1 – Modelo de dados usado no Oracle Spatial.

Modelo	Descrição
Elemento	Representa uma geometria básica, como pontos, linhas e polígonos. Cada coordenada é armazenada como um par X,Y. O anel interno e externo de um polígono é considerado como dois elementos distintos trabalhando em conjunto. As coordenadas definidas ao redor do polígono são ordenadas no sentido horário para o anel exterior e no sentido anti-horário para o anel interior.
Geometria	É uma representação espacial de uma feição, modelada como um conjunto de elementos primitivos. Uma geometria consiste de um único elemento ou um conjunto homogêneo ou heterogêneo de elementos.
Camada	É uma coleção de geometrias que tenham o mesmo conjunto de atributos. Por exemplo, a topografia de um terreno pode ser uma camada. Cada camada da geometria e o índice espacial associado estão armazenados em tabelas padrão.
Sistema de coordenadas	Pode ser georeferenciado (isto é relativo à posição na Terra) ou não-georeferenciado (cartesiana). Sendo georeferenciado as coordenadas podem ser geodésicas ou projetadas. A unidade de medida predefinida retornada nas consultas é o metro.
Tolerância	É o nível de precisão predefinido com que os dados espaciais serão tratados. Para dados geodésicos a tolerância é dada em metros. Por exemplo, uma tolerância de 100 indica igualdade na comparação entre dois pontos distante até 100 metros. Muitas funções espaciais aceitam parâmetros de tolerância opcionais.

O modelo de consulta utilizado é em duas camadas, isto é, duas operações distintas serão realizadas para obter o resultado. Essas operações são chamadas operações primárias e secundárias de filtragem. A operação primária de filtragem permite uma seleção rápida de registros, candidatos a passar pela filtragem secundária.

A primeira filtragem utiliza aproximações geométricas para diminuir a complexidade computacional e é considerado um filtro de baixo-custo. A segunda filtragem é uma operação de alto-custo e aplica computação exata às geometrias que resultaram da primeira filtragem produzindo uma resposta mais acurada da consulta espacial.

### 3.2.1. Indexação dos dados espaciais

Os índices espaciais são necessários para encontrar objetos dentro de um espaço de dados indexado que interage com um ponto ou uma área de interesse, ou ainda, para encontrar pares de objetos dentro de dois espaços de dados indexados que interajam espacialmente um com o outro (junção espacial).

O Oracle Spatial permite usar índices do tipo R-Tree (por padrão), QuadTree ou ainda ambos.

### 3.2.2. Relacionamento espacial e filtragem

O Oracle Spatial usa os filtros secundários para determinar o relacionamento espacial entre entidades no banco de dados. O relacionamento espacial é baseado na localização geométrica. Três operadores determinam o relacionamento espacial e são mostrados na Tabela 3.2.

**Tabela 3.2 – Operadores de relacionamento espacial do Oracle Spatial.**

<b>Operador</b>	<b>Descrição</b>
SDO_RELATE	Avalia critérios topológicos. Implementa o modelo de nove interseções entre pontos linhas e polígonos. Mostrado na Figura 3.2.
SDO_WITHIN_DISTANCE	Determina se dois objetos espaciais estão a uma distância específica um do outro. ( <i>Buffer</i> )
SDO_NN	Identifica um número específico de objetos espaciais vizinhos mais próximos.

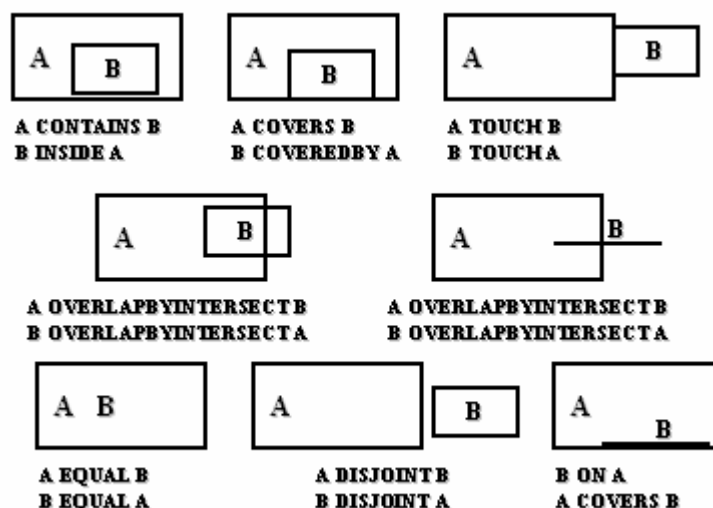


Figura 3.2 – Relacionamento topológico do SDO\_RELATE[27].

### 3.2.3. Funções agregadas espaciais

O Oracle spatial permite que se utilizem funções agregadas sobre os objetos geométricos. Estas funções podem retornar valores a partir da leitura de várias linhas obtendo apenas as linhas de interesse. Por exemplo, SDO\_AGGR\_MBR (coluna espacial) retorna o retângulo envolvente mínimo da coluna espacial.

*Geocoding* é outra característica que permite um processo de transformação de coordenadas de dados espaciais armazenadas em endereços padronizados (lista de conversão) possibilitando a integração com softwares de terceiros.

A geometria dos objetos geográficos são definidas no Oracle Spatial utilizando o tipo complexo SDO\_GEOMETRY que é explicitado na Tabela 3.3.

Tabela 3.3 – Codificação Oracle dos Tipos de Geometrias.

Valor	Tipo da geometria	Descrição
dl00	Geometria desconhecida	O Oracle Spatial ignora esta geometria.
dl01	Ponto	Contém um ponto.
dl02	Linha ou Curva	Contém uma linha, ou um arco ou ambos (linha e curva são sinônimos nesse contexto).
dl03	Polígono	Contém um polígono com ou sem buraco.
dl04	Coleção	É uma coleção heterogênea de elementos e inclui todos os outros tipos.
dl05	Multiponto	Contém um ou mais pontos.
dl06	Multilinha ou Multicurva	Contém uma ou mais linhas ou curvas.
dl07	Multipolígono	Contém múltiplos polígonos disjuntos com uma ou mais linhas.

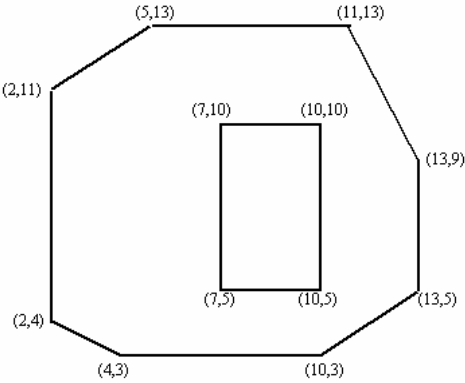
A Tabela 3.4 mostra os atributos do tipo espacial objeto-relacional do Oracle.

**Tabela 3.4 – O tipo SDO\_GEOMETRY.**

<b>SDO_GEOMETRY</b>	
<b>Atributo</b>	<b>Descrição</b>
<b>SDO_GTYPE</b>	Indica o tipo da geometria e é do tipo numérico. Suporta as Feições Simples da OGC com exceção de superfícies. O valor é codificado em 4 dígitos no formato <i>dl</i> <i>tt</i> onde <i>d</i> indica a dimensão (2,3 ou 4). <i>l</i> indica a LRS. <i>tt</i> indica a geometria. Como apresentado na Tabela 3.4.
<b>SDO_SRID</b>	Identifica o sistema de coordenada da geometria e é do tipo numérico. Se o valor for nulo nenhum sistema de coordenadas está associado a geometria e usa-se o sistema global obtido na tabela SDO_SRID.
<b>SDO_POINT</b>	É de um tipo complexo (SDO_POINT_TYPE) formado pelos atributos X, Y e Z numéricos e trabalha junto dos atributos SDO_ELEMENT_INFO e SDO_ORDINATES, este atributo só será considerado se os outros dois atributos forem nulos e ele não for. Caso contrário, será ignorado. Esse atributo deve ser usado preferencialmente por uma questão de melhor armazenamento.
<b>SDO_ELEM_INFO</b>	É do tipo complexo SDO_ELEM_INFO_ARRAY formado por um vetor de 1.048.576 ocorrências de valores numéricos. Cada elemento armazenado em SDO_ORDINATES gera uma tripla de valores, e é composta de SDO_STARTING_OFFSET, que indica a posição inicial no vetor SDO_ORDINATES de cada elemento. A primeira é sempre 1, o SDO_ETYPE indica o tipo do elemento e SDO_INTERPRETATION utilizado quando o elemento definido é composto, e se for composto do tipo 4, 1005 ou 2005 então especifica a quantidade de ocorrências e assim a quantidade de triplas subsequentes pertencentes a coleção.
<b>SDO_ORDINATES</b>	É do tipo complexo SDO_ORDINATE_ARRAY formado por um vetor de 1.048.576 ocorrências de valores numéricos. Esse tipo armazena os valores das coordenadas de um objeto espacial e deve sempre ser usado em conjunto com SDO_ELEM_INFO. Se o objeto espacial é bidimensional os pontos são armazenados dessa forma: (x1, y1, x2, y2, x3, y3...) e se são tridimensionais dessa forma: (x1, y1, z1, x2, y2, z2, x3, y3, z3...). Sem que haja nulos no vetor.

Na Tabela 3.5 um exemplo de criação, inserção e consulta utilizando o Oracle Spatial é apresentado.

**Tabela 3.5 – Exemplo do uso do Oracle Spatial.**

Representação Gráfica	Exemplo de criação, inserção e consulta
	<pre> create table empresas ( codigo number primary key, nome varchar2(50), geometria MDSYS.SDO_Geometry);  insert into empresas values (10,'área livre do terraço', MDSYS.SDO_GEOMETRY (2003, -- polígono bidimensional NULL, NULL, MDSYS.SDO_ELEM_INFO_ARRAY (1,1003,1,19,2003,1), -- polígono com buraco MDSYS.SDO_ORDINATE_ARRAY (2,4,4,3,10,3,13,5,13,9, 11,13,5,13,2,11,2,4,7,5,7,10,10,10,10,5,7,5)));  insert into USER_SDO_GEOM_METADATA values ('empresas', 'geometria', MDSYS.SDO_DIM_ARRAY ( -- 20X20 grid MDSYS.SDO_DIM_ELEMENT ('X', 0, 20, 0.005), MDSYS.SDO_DIM_ELEMENT ('Y', 0, 20, 0.005) ), NULL);  create index geotind on empresas (geometria) indextype is MDSYS.SPATIAL_INDEX;</pre>

### 3.2.4. Como manter os metadados

Os metadados descrevem as dimensões, e os limites inferior e superior da janela de visualização, além da tolerância. O Oracle Spatial permite que cada usuário espacial controle seus metadados a partir de duas visões: a USER\_SDO\_GEOM\_METADATA e a ALL\_SDO\_GEOM\_METADATA. Na primeira, o usuário tem privilégios de atualização e, na segunda, a visualização das tabelas em que há permissão de leitura.

Adicionalmente à estrutura para o armazenamento das geometrias, o Oracle Spatial fornece uma série de estruturas auxiliares que armazenam informações essenciais às aplicações de Geomática.

Uma infra-estrutura para o sistema de coordenadas é apresentada na Tabela 3.6.

**Tabela 3.6 – Infra-Estrutura para o sistemas de coordenadas do Oracle.**

<b>Campo</b>	<b>Descrição</b>
CS_SRS	Define o sistema de coordenadas válidos que podem ser atribuídos ao parâmetro SDO_SRID.
SDO_ANGLE_UNITS	Define as unidades de ângulos válidos através de uma tabela de unidades de ângulos e o fator de conversão. Dessa forma, pode-se fazer a conversão automática dos resultados.
SDO_DIST_UNITS	Define as unidades de distâncias válidas através de uma tabela de unidades de distância e o fator de conversão. Dessa forma, pode-se fazer a conversão automática dos resultados.
SDO_DATUMS	Define o <i>datum</i> válido e a sua variação de deslocamento e rotação do centro do WGS84, além da escala para ajuste.
SDO_ELLIPSOIDS	Define os elipsóides válidos (semi-eixo, esferóide etc.).
SDO_PROJECTIONS	Define as projeções cartográficas válidas (Lamberts, UTM etc.).

O Oracle Spatial apresenta ferramentas para carga em massa dos dados geográficos e vários métodos implementados. Permitem análise das informações geoespaciais, além de funções para *tuning* e performance das consultas e atualizações.

### **3.3. MySQL e sua extensão espacial**

O MySQL [28] é um SGBD Relacional desenvolvido, distribuído e suportado pela MySQL AB[28]. Possui código-fonte aberto e dois tipos de licença, uma do tipo GPL (*GNU General Public License*) e outra comercial. Na sua versão 5.0 de outubro de 2005 passou a suportar procedimentos armazenados, visões e gatilhos (*triggers*).

Trabalha com a linguagem padrão SQL99 e é um banco muito rápido quando usado com as tabelas do tipo MyIsam. Suporta várias plataformas diferentes, cerca de 60, e é um dos bancos mais populares atualmente. Apresenta uma extensão que permite a geração, armazenamento e análise de informações geoespaciais.

#### **3.3.1. Extensão espacial**

O MySQL segue a especificação da Open Geospatial Consortium (OGC). Utiliza o modelo de feições simples para SQL da OGC. Por este modelo, cada objeto geométrico está associado a um sistema de referência espacial que define o espaço de coordenadas na qual o objeto está definido e também, a que classe geométrica ele pertence.



Internamente, o MySQL utiliza um formato proprietário (em código aberto) para o armazenamento de feições espaciais, mas tanto a consulta quanto à manipulação das informações geoespaciais podem ser feitas utilizando dois formatos de dados padrão OGC, que são conhecidos como WKT (*Well-Known Text*) e WKB (*Well-Known Binary*). O primeiro serve para trocar informações geoespaciais no formato ASCII, como por exemplo, um ponto é representado pela expressão: POINT(15 20) e uma Linha com quatro pontos pela expressão: LINESTRING (0 0, 10 10, 20 25, 50 60). A gramática para definir esses objetos encontra-se no site da OGC; O segundo serve para trocar informações geoespaciais no formato binário. Por exemplo, o valor WKB correspondente ao WKT POINT(1 1) consiste de uma sequência de 21 bytes representados por 0101000000000000000000F03F000000000000F03F. Pode-se utilizar o SRID, identificador do sistema de referenciamento espacial, para determinar os sistemas de referência usados.

Para se criar valores espaciais no MySQL, utilizamos funções binárias e de texto do tipo WKB e WKT. A descrição dessas funções é mostrada na Tabela 3.7.

**Tabela 3.7 – Funções espaciais no MySQL.**

<b>Função</b>	<b>Descrição</b>
GeometryFromText (wkt [,srid])	Constrói um valor geométrico de qualquer tipo usando WKT e opcionalmente o SRID
PointFromText(wkt [,srid])	Constrói um valor ponto usando WKT e opcionalmente o SRID.
GeometryFromWKB (wkb [,srid])	Constrói um valor geométrico de qualquer tipo usando WKB e opcionalmente o SRID
PointFromWKB (wkb [,srid])	Constrói um valor ponto usando WKB e opcionalmente o SRID.

### 3.3.2. Criação de colunas espaciais

Para se utilizar a extensão geoespacial no MySQL, basta definir o tipo *geometry* na coluna da tabela em que se deseja associar a informação geoespacial,

Como por exemplo:

```
create table escola (
    id number,
    nome varchar(50),
    geometria geometry);
```

### 3.3.3. Para popular colunas espaciais

Para inserir ou alterar uma informação geoespacial em uma coluna espacial no MySQL, utiliza-se as funções anteriormente definidas,

Como por exemplo:

```
insert into escola
values ( 123,"Escola Fulano de Tal",
        GeomFromText ( 'POINT(1 1)' ) );
```

Um exemplo de um comando *insert* com geometrias mais complexas é mostrado na Tabela 3.8.

**Tabela 3.8 – Inserindo informação Geoespacial no MySQL.**

Um exemplo com WKT	Um exemplo com WKB
SET @g='POLYGON ((0 0, 10 0, 10 10, 0 10, 0 0),(5 5 , 7 5, 7 7 , 5 76, 5 5 ))'; insert into geom values (GeomFromText (@g));	insert into geom values(GeomFromWKB(0x0100000000000000 000000F03F000000000000F03F));

### 3.3.4. Para recuperar dados espaciais

A função AsText (g) converte a geometria do formato interno para o formato wkt, como por exemplo, o comando: “select AsText(g) from escola;”, retorna:

```
+-----+
| AsText(p1) |
+-----+
| POINT(1 1) |
| LINESTRING(0 0,1 1, 2 2) |
+-----+
```

Da mesma forma, a função AsBinary(g) retorna a geometria no formato binário para uso com as colunas do tipo BLOB. Por exemplo, a mesma consulta acima poderia ser feita da seguinte forma: select AsBinary(g) from geom;

O MySQL disponibiliza uma série de funções do padrão OGC que executam várias operações para a análise da informação geoespacial. Essas operações são agrupadas da seguinte forma:

- Funções de conversão de geometrias.

Ex: `GeomFromText(wkt[,srid])`, `GeomFromWKB(wkb[,srid])`, `AsText(g)`, `AsBinary(g)`;

- Funções de acesso qualitativo e quantitativo as propriedades das geometrias.

Ex: `GeometryType(g)`, `Dimension(g)`, `SRID(g)`, `Envelope(g)`;

- Funções que descrevem o relacionamento entre duas geometrias.

- Funções que criam novas geometrias a partir de geometrias existentes.

Ex: `Envelope(g)`, `StartPoint(ls)`, `EndPoint(ls)`, `PointN(ls,n)`.

O MySQL utiliza para índice espacial o R-Tree. O índice é construído usando o MBR (Minimal Bound Retangle) da geometria.

### **3.4. PostGreSQL e sua extensão espacial PostGis**

O PostGreSQL[29] é um Sistema Gerenciador de Banco de Dados Objeto Relacional (SGBDR), assim como o Oracle. Foi desenvolvido no Departamento de Computação da Universidade da Califórnia em Berkeley. Possui código-fonte aberto e licença GNU BSD. Implementa vários conceitos básicos e avançados de banco de dados tais como: procedimentos armazenados, visões, integridade referencial, transações, herança etc.

Escolheu-se analisar esse banco por ser objeto-relacional e apresentar características avançadas de banco de dados e principalmente, por apresentar uma extensão que permita a geração, armazenamento e análise de informações geoespaciais.

#### **3.4.1. A extensão espacial PostGis**

O PostGis[30] é mais do que uma extensão do SGBD. Esse módulo, desenvolvido pela Refrations Research Inc[30], provê ferramentas para visualização, análise e consulta de dados geoespaciais. Sua implementação suporta completamente a especificação OGC para especificação de feições simples.

O PostGis possui utilitários que permitem a conversão e carga de arquivos do formato shape da ESRI para o formato do PostgreSQL e vice-versa (`shp2pgsql` e `pgsql2shp`).

No PostGis, assim como no MySQL, para inserir uma informação geoespacial é necessário criar ou adicionar a uma tabela uma coluna do tipo *geometry*. A inserção das geometrias nessa coluna utiliza o processo já discutido no MySQL de WKT (*Well-Known Text*).

Os nomes das funções obedecem ao padrão definido pela OGC. A Tabela 3.9 mostra um exemplo de criação da tabela, inserção de dados e consulta às informações geoespaciais.

**Tabela 3.9 – Exemplo de utilização no PostGIS.**

Operação	Exemplo
Criação	<pre>CREATE TABLE gtest ( ID int4, NAME varchar(20) );</pre> <pre>SELECT AddGeometryColumn('dbname','gtest','geom',-1,'LINESTRING',2);</pre>
Inserção	<pre>INSERT INTO gtest (ID, NAME, GEOM) VALUES</pre> <pre>(1,'First Geometry',</pre> <pre>GeometryFromText('LINESTRING(2 3,4 5,6 5,7 8)',1));</pre>
Consulta	<pre>SELECT id, name, AsText(geom) AS geom FROM gtest;</pre> <pre>-----+-----+-----</pre> <pre>id   name             geom</pre> <pre>-----+-----+-----</pre> <pre>1    First Geometry   LINESTRING(2 3,4 5,6 5,7 8)</pre> <pre>(1 row)</pre>

A versão de produção atual do PostGis utiliza o GIST (*Generalized Search Tree*) como tipo de indexação para as colunas espaciais. Nas versões posteriores o índice R-Tree poderia ter sido usado, mas por questões de limitação do PostGreSQL o índice GIST foi adotado. Uma das limitações tecnológicas era que feições com tamanho maior que 8Kbytes não podiam ser manipuladas. E ainda, os índices R-Tree não eram à prova de nulos e colunas do tipo *geometry* que contivessem nulos poderiam gerar erros.

O PostGis provê um suporte a sistemas de referências espaciais que permite que as informações geoespaciais sejam convertidas segundo um sistema de projeções previamente carregado. A função *Transform(coluna\_espacial, SRID)* permite efetuar essa conversão.

Os objetos suportados pelo PostGis são as feições simples definidas no OGC. Essas feições podem ter duas ou três dimensões. Pode-se usar as funções *force\_2D* e *force\_3D* para converter as dimensões de uma para outra. Os objetos suportados são: POINT, LINESTRING, POLYGON, MULTIPOINT, MULTILINESTRING, MULTIPOLYGON e GEOMETRYCOLLECTION.

Os objetos são expressos utilizando a forma padrão WKT e WKB ou ainda, por sua forma canônica que permite que o identificador do sistema de referenciamento espacial seja incluído. Um exemplo de uma consulta usando a forma padrão e canônica é mostrado na Tabela 3.10. Observe que o SRID não aparece na forma padrão e a forma canônica poderia então ser utilizada em algum processo de conversão.

**Tabela 3.10 – Exemplo das formas padrão e canônicas OGC no PostGis.**

Forma padrão	Forma canônica
<pre>SELECT AsText(geom) AS OGCGeom FROM thetable; OGCGeom ----- LINESTRING(-123.741378393049 48.9124018962261,-123.741587115639 48.9123981907507) (1 row)</pre>	<pre>SELECT geom AS PostGISGeom FROM thetable; PostGISGeom ----- SRID=123;LINESTRING(-123.741378393049 48.9124018962261, -123.741587115639 48.9123981907507) (1 row)</pre>

O PostGIS provê duas tabelas para armazenamento de metadados: SPATIAL\_REF\_SYS e GEOMETRY\_COLUMNS. A primeira armazena as informações sobre os sistemas de referencia espaciais que podem ser armazenados no banco. A segunda armazena informações sobre as colunas do tipo *geometry* criadas, como por exemplo, as informações do SRID, o tipo de geometria, nome e localização (esquema, tabela). Informações adicionais sobre os metadados são encontradas na documentação do produto.

Para recuperação da informação geoespacial o PostGIS disponibiliza alguns operadores, vistos na Tabela 3.11. Estes são utilizados juntamente com a sintaxe normal do SQL não-espacial.

**Tabela 3.11 – Operadores do PostGis.**

Operador	Descrição
<b>&amp;&amp;</b>	Indica se há sobreposição dos retângulos envolventes mínimos das duas geometrias relacionadas.
<b>~=</b>	Testa se duas geometrias são geometricamente idênticas.
<b>=</b>	Testa se os retângulos envolventes mínimos das duas geometrias relacionadas coincidem.

Exemplo:

```
SELECT ID, NAME
FROM ROADS_GEOM
WHERE GEOM ~=
GeometryFromText('LINESTRING(191232
243118,191108 243242)',-1);
```

Além dos operadores já discutidos, o PostGIS suporta todas as funções definidas no OGC, tais como: AddGeometryColumn, Dimension, Disjoint, Overlaps e ExteriorRing, entre outras.

### 3.5. Comparando os três bancos:

Escolheu-se o Oracle porque parece ser a mais completa ferramenta para armazenamento geoespacial comercial e o MySQL e o PostGis por serem alternativas em Software Livre. Ao final criou-se um quadro comparativo das três ferramentas. Para melhor avaliação pelo leitor da utilização destes SGBD's, organizou-se na Tabela 3.12 algumas de suas características.

**Tabela 3.12 – Comparação entre Oracle, MySQL e PostgreSQL.**

	<b>Oracle</b>	<b>MySQL</b>	<b>PostGreSQL</b>
<b>Tipo do banco</b>	objeto-relacional	relacional	objeto-relacional
<b>Licença</b>	Comercial	GLP e comercial	GNU
<b>Preço Licença em 2006</b>	US\$40.000	0 ou US\$400,00	--
<b>Atende a OGC</b>	totalmente	parcialmente	totalmente
<b>Indexação</b>	Quad-Tree e R-Tree	R-Tree	GIST
<b>Suporte XML</b>	sim	não	não
<b>Ferramentas de visualização</b>	sim	não	sim
<b>Sistema de Referências</b>	sim	não	sim
<b>Armazena Metadados</b>	sim	não	precariamente
<b>Conversão para GML</b>	não	não	não
<b>Documentação</b>	bastante	pouca	pouca
<b>Suporte Oficial</b>	sim	sim	não

## CAPÍTULO 4 - O PROBLEMA DA INTEGRAÇÃO SEMÂNTICA

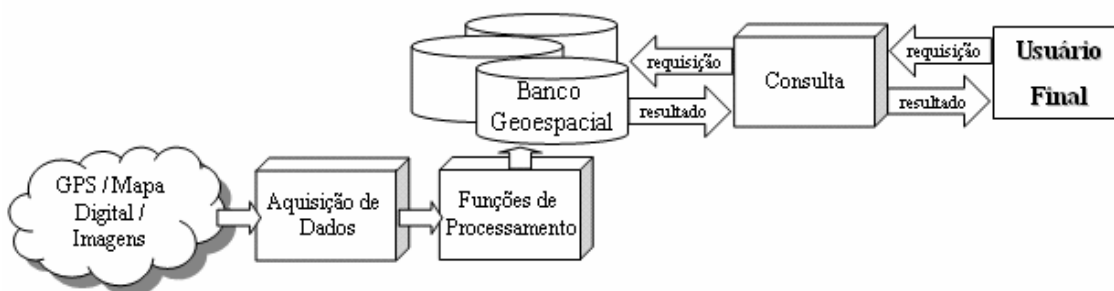
*"Quando o mapa e o terreno não  
coincidem, acreditem sempre no terreno".*

Ronald C. Gause

Neste capítulo serão apresentados dois cenários de ambientes geoespaciais. O cenário I descreve uma aplicação típica de um ambiente geoespacial local. O cenário II descreve as alterações que ocorrem quando a componente distribuição é acrescida. De cada módulo dos cenários estudados serão apresentadas descrições, caracterização e exemplificação de problemas. Com isso, levantam-se os obstáculos para a integração semântica.

### 4.1. Cenário I: uma aplicação típica de um ambiente geoespacial local

Neste cenário, mostrado na Figura 4.1, limitou-se a análise de um SIG de baixa complexidade, isto é, que possa ser implementado com o modelo de “feições simples” (*simple feature*) da OGC[22]. Entende-se que esse tipo de SIG atende a maioria dos requisitos de um SIG e, por isso, levantar os problemas da integração para este tipo de SIG seria uma grande contribuição.



**Figura 4.1 – Cenário típico de um ambiente Geoespacial Local.**

Dentre os diversos processos e serviços que um sistema geoespacial pode ter, os seguintes aspectos foram abordados: a Aquisição de Dados, as Funções de Processamento, a Consulta das informações armazenadas e o Banco Geoespacial, apresentados como módulos do cenário. A caracterização do problema foi baseada na premissa de que os sistemas foram construídos sem a preocupação da integração semântica.

#### 4.1.1. Aquisição de dados

- **Descrição:** esse processo define a entrada de dados espaciais para o sistema de informação geográfica, ou seja, que dados o sistema irá gerenciar e manter para responder aos seus requisitos.

- **Caracterização:** normalmente, os SIG(s) precisam de dados de outras fontes, além daqueles dados que é proprietário, isto é, que irão coletar e manter atualizados. Por conseguinte, para atender as suas necessidades, bases inteiras são compradas. Apesar de necessárias ao sistema, essas bases não são atualizadas, tornando-se obsoletas. Além disso, a impossibilidade de comprar apenas a informação que se deseja torna essas bases subutilizadas.

- **Exemplo:** supondo a construção de um SIG de um Sistema de Monitoração Imobiliária (SMI) a qual monitora as construções de imóveis junto à Rodovia Presidente Dutra e adjacências e que o objetivo do sistema é fiscalizar se as construções estão fora do limite de distância e área proibidos por lei.

Para atender a este requisito, o usuário do SIG entraria ou com as informações georeferenciadas dos imóveis suspeitos ou com pedidos de licença para construção e os manteriam atualizados sob uma base cartográfica geopolítica dos Estados do Rio de Janeiro e São Paulo.

Convém destacar que era necessário conhecer as informações georeferenciadas da Rodovia Presidente Dutra e rodovias adjacentes. Para tanto, uma base do Departamento Nacional de Infra-Estrutura de Transportes (DNIT) com todas as estradas do país foi adquirida, pois era a única disponível. Além da subutilização, quando uma nova estrada fosse aberta ou modificada, a base adquirida não seria atualizada, o que traria inconsistência.

#### 4.1.2. Funções de processamento

- **Descrição:** essas funções definem que procedimentos serão realizados sobre as bases adquiridas e os dados coletados para agregar valor à informação e qualidade ao sistema. Por exemplo: tratamento de imagens, orto-retificação, cadastramento de dados textuais, transformações de datum etc. Após esse procedimento, para garantir a



persistência dos dados, torna-se necessário o armazenamento da informação em um banco de dados geoespacial.

- **Caracterização:** os dados coletados eram tratados e armazenados diretamente no SIG, mas ao receber a base externa era necessário decidir como integrá-los com as informações do sistema. Era preciso adequar estas aos requisitos do sistema. A conversão ou migração para a base do SIG era a solução mais comum. Entretanto, a conversão de formatos e migrações pode levar a erros devido aos diferentes modelos dos formatos e ao volume de dados migrados causas de perdas tanto semânticas quanto sintáticas.

- **Exemplo:** no Sistema de Monitoração Imobiliária, o usuário entrava com os dados espaciais dos imóveis coletados por meio de um GPS. A base de rodovias adquirida do Departamento Nacional de Infra-Estrutura de Transportes (DNIT) com 20 milhões de registros foi entregue no formato “shape”<sup>8</sup> e na escala de 1:50000 com data de atualização de 2000. Esta base foi importada para a base de dados local no formato MGE (Microstation), concebido na escala 1:20000, com informações de 2006. Algumas aproximações foram feitas para adequação das informações, o que gerou perdas semânticas e erros. Por exemplo, em 2006 um novo conceito para rodovias foi criado tornando agora 400 novas rodovias que antes eram classificadas como vias secundárias. Esta informação só foi atualizada na base do DNIT, e implicou em inconsistência entre as bases.

#### 4.1.3. Banco Geoespacial

- **Descrição:** é o repositório da informação geográfica local.

A arquitetura do banco de dados geográfico determina o modelo utilizado e a estrutura de armazenamento da informação geoespacial, que pode ser dos tipos: dual, baseada em campos longos ou utilizando as funcionalidades geoespaciais dos Sistemas Gerenciadores de Banco de Dados.

---

<sup>8</sup> Formato Shape formato ESRI para armazenamento e distribuição da informação geográfica.

- **Caracterização:** no ambiente geoespacial local, a escolha da arquitetura é intrínseca ao SIG utilizado. E muitas vezes, as abordagens dual e a que utiliza campos longos, implica em maior custo na administração dos dados, porque existe dissociação entre as informações descritivas e a geometria das feições modeladas. Essa dissociação prejudica a manutenção da base e pode gerar inconsistências e perda de performance. Estas são agravadas por terem um volume de dados muito grande. Por outro lado, a solução de banco de dados extensível permite maior associação entre os dados textuais e geométricos.

- **Exemplo:** supondo-se que no Sistema de Monitoração Imobiliária seja utilizado a arquitetura dual. Dessa forma, as bases descritivas dos imóveis estarão armazenadas no SGBD SQLServer, enquanto os mapas (informação geométrica) estarão armazenados no SIG ArcGIS. Sempre que as informações geográficas sofrerem alterações, as bases deverão ser atualizadas para manterem a consistência. Porém, por estar dissociado o SGBD do SIG, esta atualização não é automática, o que pode causar inconsistências, como por exemplo, exclusão de polígonos sem exclusão da referida informação na base SQLServer.

Além disso, cada ferramenta tem forma particular para salvar dados para recuperação (*backup e restore*), oferece segurança de acesso aos dados, indexar os dados, fazer o gerenciamento de processos e memória, enquanto fica a cargo do projetista manter a integração entre as duas ferramentas. E como não há estruturas auxiliares, índices, comuns as duas implementações, a consulta que envolve dados dos dois lados exige mais processamento, e isto prejudica a performance.

#### 4.1.4. Consulta

- **Descrição:** esse processo justifica o uso do SIG, isto é, como o usuário final recuperará as informações geoespaciais. As consultas e as respostas seriam tanto gráficas como textuais.

- **Caracterização:** normalmente, os SIG(s) não disponibilizam para o usuário final uma linguagem de consulta. A consulta é feita a partir de uma interface gráfica e o resultado da consulta é um gráfico ou uma tabela. Quando deseja-se disponibilizar informações, geralmente, publica-se na Web uma interface para consulta que retorna um

mapa ao usuário. O resultado da consulta não é apropriado para a distribuição da informação por não agregar os metadados.

- **Exemplo:** No Sistema de Monitoração Imobiliária a consulta era feita com acesso o modo de consultas do ARCGIS que disponibilizava uma interface gráfica onde o resultado era visualizado na tela e tratado de diversas formas. Porém, quando decidiu-se colocar a consulta na Web, o retorno era uma tabela textual com a lista de proprietários e uma imagem no formato GIF com as geometrias dos imóveis em um mapa. Nesse caso, a informação espacial vetorial do imóvel não foi recuperada pela consulta e não poderia ser reaproveitada por outro sistema que reconhecesse e tratasse essa informação. Os metadados e as restrições (regras de integridade) também não foram repassados e diminuiu a integração semântica.

#### 4.2. Cenário II: o componente de distribuição da informação geográfica

O cenário seguinte, ilustrado na Figura 4.2, representa uma aplicação típica de um ambiente geoespacial local, agora acrescido das informações geoespaciais advindas de Centros de Dados Geoespaciais. Esse cenário seria uma evolução do primeiro cenário, onde surgiu a necessidade de integrar as informações espaciais.

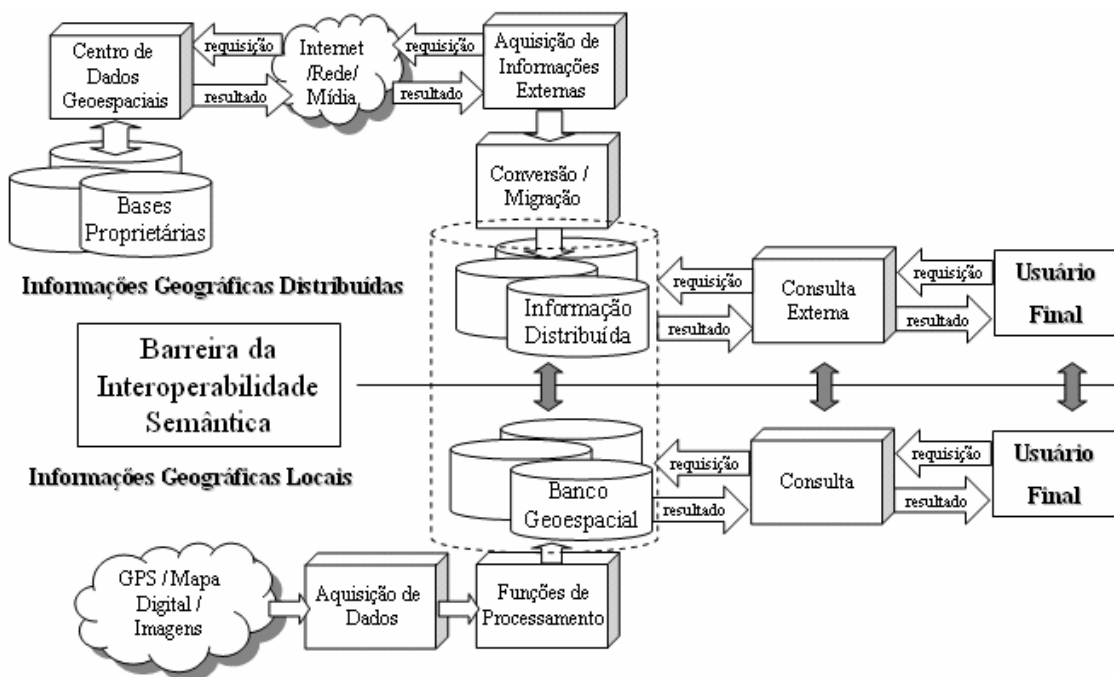


Figura 4.2 – Cenário anterior acrescido da componente distribuição.

#### 4.2.1. Centro de Dados Geoespaciais

- **Descrição:** é o fornecedor da Informação Geográfica Distribuída. Normalmente, são agências governamentais ou particulares que armazenam, processam e distribuem dados geográficos para os mais variados fins.

A forma de consulta (requisição) e o formato do resultado dependem do Centro de Dados e, geralmente, está relacionado ao formato de armazenamento proprietário destes Centros.

- **Caracterização:** o surgimento dos Centros de Dados possibilitou que um usuário adquira das informações externas somente o que seja de seu interesse e que esteja atualizado. Porém, apesar dos esforços da OGC, ainda não existe padronização das formas de consulta e da apresentação dos resultados. Apesar de ser consenso que o documento GML seja o melhor formato para este fim, inexiste uma forma padronizada de consultar e receber o resultado em GML. Dessa forma, os primeiros Centros de Dados podem disponibilizar a informação em outros formatos por estarem mais familiarizados, e isto poderá exigir conversão ou migração pelo receptor. Outra questão é a granularidade da distribuição, isto é, quanto de informação será recuperada por cada vez. Devido ao fato de não haver padrão na linguagem de consulta e formato de distribuição estes variam de centro para centro.

- **Exemplo:** no Sistema de Monitoração Imobiliária, após perceber que o sistema ficou defasado, visto que novas estradas foram criadas ou as antigas atualizadas, descobriu-se que o Departamento Nacional de Infra-Estrutura de Transportes (DNIT) havia disponibilizado um novo Centro de Dados onde essa informação seria obtida em tempo real. Procedeu-se, então, um convênio e toda vez que havia necessidade de consultar, o usuário acessaria uma página *Web*, selecionaria trechos de 10Km e receberia os no formato shape.

#### 4.2.2. Aquisição de Informações Externas

- **Descrição:** esse procedimento é responsável por obter a Informação Geográfica Distribuída e, para isso, detém as formas de interagir com cada Centro de Dados, ou sejam, formas de acesso (localização e conta de acesso), estruturas dos dados, linguagem de consulta e formato da resposta.

- **Caracterização:** para cada Centro de Dados que se quer obter a informação distribuída, torna-se necessário que o sistema conheça: a localização do Centro, a estrutura de armazenamento, o significado dos dados, a forma de recuperação, o formato de retorno. Com isso, terá condição de desenvolver uma rotina para aquisição automática dessa informação. Se algum dos aspectos mencionados acima mudar, sem que haja comunicação e preparação prévia, haverá falha neste processo.

Outra questão é que, apesar dos metadados relativos à distribuição (quando a informação foi adquirida, de qual fonte, sobre qual consulta realizada etc.) serem essenciais para análise futuras e para a verificação da confiabilidade, normalmente, estes metadados são descartados logo após a consulta.

- **Exemplo:** supondo que no Sistema de Monitoramento Imobiliário, após o estabelecimento de um convênio com o DNIT, foi entregue uma conta e fornecido uma senha para acesso e consulta. O usuário acessou a informação através da *Web* por algum tempo, porém algum tempo depois, o sistema foi atualizado e o formato de resposta alterado para documentos GML. Apesar de continuar com permissão de consulta, o resultado não será utilizado automaticamente, como antes. Apesar de mudar o formato de saída a forma de consulta permanecerá a mesma e a consulta de trecho de 100 Km da rodovia será entregue em 10 arquivos. A data da consulta, o fornecedor da informação e o critério da consulta estarão perdidos após a consulta, restará o armazenamento apenas dos documentos GML.

#### 4.2.3. Conversão e Migração

- **Descrição:** este procedimento é responsável por tratar a Informação Geográfica Distribuída e armazená-la após o recebimento. É análogo ao processo de funções de processamento da Informação Geográfica Local.

- **Caracterização:** a falta de um padrão para distribuição dos metadados gera perdas de informação após as conversões e migrações. Criar métodos de migração e conversão sem essa preocupação não resolve o problema da falta de integração.

Provavelmente o SIG local usará métodos e padrões ligeiramente diferentes dos métodos utilizados pelos Centros de Dados. Assim, se os dois juntarem informações do mesmo local, estas não se fundirão imediatamente. Talvez alguns dados, de um mesmo elemento geográfico, coletados em um, não tenham sido coletados em outro, apesar de todos os outros parâmetros serem os mesmos. Os destinatários desses dados precisarão ajustar suas análises para corrigir esta deficiência. Talvez o sistema de projeção utilizado seja diferente. Um Centro de Dados utiliza coordenadas longitude/latitude e o outro CD em coordenadas UTM. Talvez um CD comece a colecionar dados de umidade do solo e a documentar seu método de análise, mas há vários métodos e aparelhos para medir e representar a umidade do solo. É provável ainda que o método de aquisição para os dados coletados previamente pelo outro não tenha sido documentado, ou seja, há distintas formas de aquisição, tratamento, representação da informação geográfica nos diversos Centros de Dados.

- **Exemplo:** Supondo que no sistema Sistema de Monitoração Imobiliária (SMI), os dados recebidos do Departamento Nacional de Infra-Estrutura de Transportes (DNIT) que inicialmente estavam no formato vetorial, sejam convertidos para o SGBD Oracle Spatial por um processo próprio desenvolvido que filtrava do formato vetorial apenas as informações de interesse. Daqui a algum tempo, o DNIT disponibilizará o resultado em GML, este será armazenado no sistema de arquivos, criando assim uma dissociação entre os dados recebidos antes da alteração armazenados no SGBD e os dados armazenados nos documentos GML em sistema de arquivos. Além disso, o DNIT não armazenava a largura das pistas de rolamento e essa informação era importante para os cálculos no SMI embora não fosse para o sistema da DNIT.

#### 4.2.4. Consulta Externa

- **Descrição:** esse procedimento é responsável por consultar a Informação Geográfica Distribuída e permitir a utilização da informação distribuída pelo SIG, isto é, fazer algum nível de integração entre a Informação Geográfica Local e a Informação Geográfica Distribuída.

- **Caracterização:** na consulta externa o seguinte problema surge, o sistema já está preparado para consultar a informação local. Porém, se obteve a informação distribuída e a mesma foi armazenada segundo algum critério discutido anteriormente. Se essas informações estão dissociadas, isto é, em repositórios diferentes, formas de consulta diferentes para cada repositório terão de ser utilizadas. Então, caberia ao projetista implementar todos os aspectos auxiliares para integrar as duas consultas e garantir sua eficiência e performance. Além disso, os problemas relacionados à utilização da estratégia dual para banco de dados geoespacial também seriam aplicados neste novo enfoque, questões como backup, segurança de acesso etc.

- **Exemplo:** no Sistema de Monitoração Imobiliária não havia procedimentos prontos para integrar a GML no mesmo repositório, por isso, essas informações eram consultadas em separado e analisadas através de uma ferramenta que visualizava o documento GML, utilizando a conversão para SVG e exibido no navegador *Web*. Esse documento era analisado visualmente pelo técnico e comparado com a informação local. Com o tempo, muitos arquivos ficavam disponíveis no sistema de arquivos e tornado difícil a recuperação da informação.

#### 4.2.5. Informação Distribuída

- **Descrição:** é o repositório da informação geográfica distribuída.

O formato do arquivo recebido dos Centros de Dados e a estratégia utilizada de migração ou conversão determina o tipo de armazenamento, que pode ser: em sistema de arquivo ou em SGBD.

- **Caracterização:** a questão é o que fazer com os arquivos recebidos de diversas fontes de dados? Várias estratégias são possíveis, porém, pouco se discute a respeito desse problema. Normalmente, o documento recebido fica armazenado no sistema de arquivos. Entretanto, para o caso do documento GML, ainda existe a possibilidade do armazenamento em banco de dados nativo XML, por exemplo, o Tamino da Software AG. Apesar de ser uma estratégia melhor do que manter em sistema de arquivo, cria a mesma dissociação entre as informações locais e distribuídas. Além disso, esses bancos priorizam o armazenamento da estrutura XML, mas não tem recursos adicionais para tratar a informação geoespacial. Outra solução seria misturar as informações locais e

distribuídas na mesma base. Todavia, essa solução exige um trabalho muito maior e com risco de perda de informação devido às diferenças entre modelos. Por outro lado, a GML tem uma deficiência em relação à definição das linhas. Usa-se o elemento <coordinates> para representar as coordenadas dos pontos de uma linha ou polígono. Essa coordenada é uma string de pares de coordenadas X e Y. Essa deficiência, na verdade, é uma limitação da GML para diminuir o tamanho do arquivo e foi corrigida na versão 3 da GML na qual incorporou os elementos x e y ao elemento ponto. Dessa forma, o uso da GML aparentemente é muito bom para distribuição, mas deficiente para o armazenamento da informação geoespacial.

- **Exemplo:** no Sistema de Monitoração Imobiliária, toda vez que o usuário faz uma nova requisição, coloca os documentos GML numa pasta chamada DNIT e a disponibiliza no sistema de arquivos. Quando se tem de recuperar essa informação novamente é preciso encontrar essa informação e descobrir qual o critério de pesquisa para estes documentos, cujo o tamanho exige muito processamento para recuperar a informação desejada.

#### 4.3. Obstáculos para integração semântica:

Tendo em vista as análises realizadas, encontraram-se os seguintes obstáculos para se conseguir a integração semântica entre a informação geográfica local e a informação geográfica distribuída, conforme apresentados na Tabela 4.1.

**Tabela 4.1 – Obstáculos à integração semântica.**

<b>Obstáculos à Integração</b>
Grande Volume de Dados
Diversos Tipos de Armazenamento de Dados
Diversidade de Fontes de Informação
Formatos Heterogêneos de Dados
Dependência da Estrutura e Localização dos Dados
Falta de Estruturação dos Metadados
Descasamento dos Dados
Diversidade de Linguagens de Consulta



## **CAPÍTULO 5 - O ESTADO DA ARTE DO ESTUDO DA INTEGRAÇÃO DA INFORMAÇÃO GEOGRÁFICA DISTRIBUÍDA**

*"Não se pode ensinar tudo a alguém, pode-se apenas ajudá-lo a encontrar por si mesmo."*

Galileu Galilei

Este capítulo tem o intuito de explanar e discutir a visão de vários autores sobre os temas abordados nessa pesquisa, principalmente, a consulta semântica, o armazenamento de informações geoespaciais utilizando SGBD e a integração das informações.

### **5.1. O Estado da Arte em Consulta Semântica**

A consulta semântica é a característica, mais desejável, em qualquer sistema de informação[31]. Não adianta modelar os dados, estruturá-los e armazená-los eficientemente se não há como recuperar a informação com eficiência. E isso exige que cada vez mais se elevem os níveis de abstração da informação e se distancie da tecnologia. No restante desse tópico será mostrado como alguns autores abordam essa questão.

BERNERS-LEE *et al*, em seu artigo “The Semantic Web” [32], sonha com uma rede de informações onde as consultas submetidas por usuários sejam efetuadas através de programas (agentes) que recebam os detalhes de um problema, como parâmetros, e a partir de inferências consigam resolvê-lo. Berners-Lee sabe que para este sonho se tornar realidade é preciso que as informações disponíveis na *Web* tenham cada vez mais semântica, isto é, que estejam estruturadas e relacionadas entre si (rede semântica). O pai da HTML define a *Web* Semântica não apenas como um ambiente separado, mas uma extensão da *Web* atual.

Para que a *Web* Semântica funcione, os computadores devem ter acesso a coleções estruturadas de informações e a conjuntos de regras de inferências e que estes sejam usados para conduzir a reações automáticas. Essas regras necessitam de uma linguagem tão expressiva quanto o necessário para permitir à *Web* reagir tanto quando desejado, sem a obrigação de tentar resolver todas as questões. Essa característica

filosófica é baseada no teorema de Godel: “Qualquer sistema que seja complexo o suficiente para ser útil também conduz a questões inexplicáveis” e nos conduz a pensar em resultados mais práticos ao invés de tentar resolver todas as questões. Tim afirma que a *Web Semântica* permitirá que máquinas compreendam tanto documentos semânticos quanto falas humanas.

BERNERS-LEE considera importante, para desenvolver a *Web Semântica*, o uso de duas tecnologias: a XML, que permite a todos criarem suas próprias marcações e, com isso, fazer com que programas usem essas marcações de maneira mais sofisticada; e a RDF, para expressar o significado usando um conjunto de triplas, sendo cada tripla preferivelmente: sujeito, verbo e objetos. Essas triplas podem ser escritas usando marcações XML. Os sujeitos, verbos e predicados são identificados por um URI, sendo a URL a URI mais comum. As triplas RDF formam redes de informações sobre objetos relacionados.

Berners-Lee acrescenta que para aumentar-se a semântica é necessário ter os mesmos identificadores e, normalmente, bancos diferentes trazem identificadores diferentes. Um programa que compara ou combina (integra) informações através de dois bancos de dados, tem de saber se os termos usados significam a mesma coisa. Idealmente os programas devem ter uma maneira de descobrir o significado comum da informação em qualquer banco de dados onde a informação for encontrada. Uma solução para esse problema é fornecer uma terceira componente para a *Web Semântica*, que é uma coleção de informações chamada Ontologia. Na filosofia, ontologia é a teoria sobre a natureza da existência, do ser concebido como tendo uma natureza comum que é inerente a todos e a cada um dos seres. Os desenvolvedores *Web* e de inteligência artificial tem corrompido o termo ontologia segundo seu próprio jargão. Para eles, ontologia é um documento ou um arquivo que formalmente define as relações entre os termos. O tipo mais comum de ontologia na *Web* tem uma taxonomia e um conjunto de regras de inferência.

A taxonomia define classes de objetos e o relacionamento entre estes. Contudo, as regras de inferências são mais poderosas e são como as restrições de integridade de um banco de dados. Ora, se uma tripla RDF indica que uma ponte deve estar sobre a água mesmo que sintaticamente esteja correto escrever em GML uma feição ponte sobre uma feição prédio, semanticamente estaria errado, assim como ocorre com as restrições de integridade de um SGBD.

Em outro artigo, BERNERS-LEE[33] relata que o verdadeiro poder da *Web Semântica* será alcançado quando forem criados muitos programas que colem conteúdo da *Web* de diversas fontes, processem as informações e troquem o resultado com outros programas. Para tanto, o grupo que trabalha com ontologia na *Web* dedica-se a construir sobre um núcleo RDF uma linguagem para definir ontologias *Web* estruturadas, conforme observa-se em [34][35].

MACKDAY [36] relata que com o aumento da interconectividade global surge a necessidade de negociar com mais informações heterogêneas que consistem não somente de dados digitais, mas também de operações e modelos de simulação criadores de novos dados e informações. Essas aplicações criam um desafio de integração significativa, desde diferenças de escala dos dados no tempo e espaço aos modelos que são dificuldades à reconciliação. O autor afirma que a ontologia, semântica e contexto são idéias fundamentais para integração.

Em outro artigo, ENGENHOFER *et al.* [37] mostra como é difícil encontrar informações geográficas com conteúdo na *Web*, devido à enorme variedade de codificações da semântica geoespacial, e que é necessária uma forma de organizar a informação. Ele propõe que múltiplas ontologias espaciais sejam criadas para aumentar a semântica, isto é, o significado das informações encontradas na *Web*.

A maneira proposta por Engenhofer é a utilização da linguagem XML e da RDF, onde a linguagem XML fornece uma solução sintática para codificar a informação geoespacial. Em sua proposta, Engenhofer considera quatro representações para a semântica geoespacial: linguagem natural, metadados simples baseados em XML, modelo de dados com RDF e um modelo semântico lógico baseado em DAML+OIL.

Estas representações são necessárias para que tanto pessoas quanto máquinas interajam cooperativamente, a fim de explorar os diferentes significados disponíveis nas informações geoespaciais.

Criar objetos primitivos e operações geoespaciais que possam ser efetuadas com estes objetos não fazem com que haja maior semântica na consulta. Por exemplo, achar todas as livrarias que estão contidas em viadutos, apesar de ser operacionalmente possível em consultas, na vida real não tem significado algum.

Para Engenhofer, existem duas questões iniciais a serem pesquisadas:

- Uma forma canônica plausível para criar consultas a dados espaciais. (Ou seja, a XQuery atende a este requisito pois permite consultar tanto documento XML, quanto ter acesso a banco de dados e ainda obter a resposta em GML).

- Métodos para julgar a semântica das fontes de dados disponíveis e assim verificar se a estrutura semântica pode ser explorada por uma consulta em particular.

A gerência das informações geoespaciais é uma necessidade cada vez maior para fornecer uma estrutura semântica à *Web*. CASEY *et al* [38] apresentam a necessidade de uma metodologia semântica na *Web* para oferecer suporte às decisões espaciais. Casey mostra a necessidade de interoperar com um conjunto mínimo e aberto, com interfaces de baixo nível como SQL para acessar dados espaciais, diminuindo a heterogeneidade.

Outra tendência é o surgimento de Servidores de Mapas na Internet. *Web* sites que fornecem uma interface interativa estão se tornando cada vez mais comuns, como: *maps.google.com*, *www.apontador.com.br* ou *www.maprequest.com*. Isto permitirá que um sistema de suporte a decisão consulte diversos IMS (*Internet Map Server*) para seus objetivos.

Uma terceira tendência é o crescimento dos serviços baseados em localização (LBS) na indústria de telecomunicações sem fio.

Essas três tendências estão resumidos em bancos de dados transacionais cliente/servidor, aplicações baseadas em mapas na *Web* e serviços *Web* sem fio. Entretanto, falta interface comum para a integração dessas tecnologias.

Pesquisadores trabalham no objetivo de adicionar maior semântica à *Web* e, nesta luta, vê-se trabalhos como os de HENDLER[41], que procura integrar aplicações na *Web* com maior semântica entre elas. Sua motivação é a existência da necessidade de prover uma plataforma universalmente acessível cujo compartilhamento dos dados e processos, seja possível tanto por ferramentas automatizadas quanto por pessoas que anseiam por informação. Para tanto, propõe: primeiro, a ligação entre banco de dados com o uso da XML; segundo, o uso da RDF para compartilhamento de conteúdo e restrição entre dados; e por fim, a padronização das interfaces entre os serviços e aplicações na *Web* utilizando SOAP.

HENDLER[39] menciona o desejo do internauta pela integração das informações contidas em diferentes meios na Internet para seu melhor aproveitamento. Exemplo: um e-mail contendo um compromisso da agenda, uma planilha e um mapa da cidade.

Um exemplo da revolução que a *Web* proporcionou na forma de consultar, aumentando a semântica, é que o usuário apesar de nunca ter estado ou mesmo ter ouvido falar do endereço de uma página, infere um endereço sem a ajuda de *sites* de pesquisa (como *www.google.com*). Por exemplo, se alguém estivesse a procura de

informações de hotéis, um *site* que naturalmente seria tentado é [www.hoteis.com.br](http://www.hoteis.com.br) mesmo que nunca tivesse tido conhecimento neste site. A *Web Semântica* proveria a facilidade de recuperar a informação.

VATSAVAI[40] comenta como a XML vem se tornando um padrão de fato para a distribuição da informação e espera que a GML também se torne um padrão para a distribuição da informação geográfica. Entretanto, explica a necessidade de se estender a XQuery para atender aos requisitos geográficos e propõe a linguagem GML-QL, uma extensão da XQuery, em concordância com a posição defendida neste trabalho.

Vatsavai defende que entender a XQuery é importante, pois retrata da base no futuro para os sistemas de processamento de consultas a documento XML. Afirmar também que, em grandes quantidades de dados contidos em documento GML, a escolha do armazenamento é a principal decisão para um processamento de consultas eficiente.

Em seu artigo, Vatsavai considera que os SIG(S) *Web* comerciais não são baseados em SGBD Relacional, mas essa afirmação tornou-se inverídica atualmente, pois o ARCIMS<sup>9</sup>, por exemplo, publica mapas na *Web* e tem diversos conectores para trabalhar com banco de dados.

O mesmo autor escolheu a XQuery por ser uma linguagem poderosa, capaz de oferecer flexibilidade para especificar consultas complexas (que envolve vários tipos de join), de ser extensível através de funções definidas pelo usuário e, principalmente, por ser um padrão atual. A resposta à consulta com XQuery é um documento XML. Isto lhe permite ser um documento GML.

Consultas espaciais classificam-se em 3 categorias: consultas puramente espaciais (que envolvem apenas atributos e relacionamentos espaciais); consultas não-espaciais e consultas mistas.

Convém resaltar que, para o autor, a escolha para armazenamento da GML em banco de dados Objeto Relacional é uma boa decisão devido à complexidade de implementação de predicados espaciais e funções de análise espacial sobre dados semi-estruturados.

Os usuários tentam combinar as informações de diferentes bancos de dados. Essa é uma tarefa difícil por se tratar de formas proprietárias de ligação. Mas, com o uso da XML, um programador aprende através dos metadados e combina as informações de

---

<sup>9</sup> Servidor de Mapas para Internet da ESRI

diferentes documentos XML e ainda tenha restrições entre eles, nunca como no SGBD Relacional.

Um componente fundamental para a *Web Semântica* é o RDF. Quando informações de duas fontes precisam ser integradas, pode-se basicamente concatenar esses arquivos em um maior, juntando todos os termos que foram definidos que correspondam à URI. Quando o objetivo é estender uma consulta sobre um arquivo RDF e incluir restrições do outro é só adicioná-las como parte da integração. Deste modo, onde a XML produz elementos e atributos para descrever os dados do arquivo, a linguagem RDF usa declarações para expressar o valor da propriedade de algum objeto, o que equivale exatamente a uma célula de tabela em um SGBD.

O uso do URI e da XML são componentes fundamentais para a base da *Web Semântica*, além do RDF para o relacionamento de significados. O grupo que trabalha com Ontologias na *Web* está endossando estruturas com RDF para este fim.

A XML começou como um projeto dirigido às limitações da HTML sobre documentos estruturados e agora tem emergido muito bem como a infra-estrutura para dados estruturados intercambiados.

Vatsavai entende que no intercambiamento de documentos ou banco de dados é tradição que o produtor e o consumidor concordem com as estruturas das informações. Porém, isso não é suficiente para o sucesso em longo prazo da *Web Semântica*, a qual deve distribuir a informação independentemente do aumento do entendimento, com a incorporação de novas informações sem insistir que as velhas sejam modificadas.

Outro exemplo dos esforços para produzir uma linguagem de consulta depara-se no trabalho de BOUCELMA *et al*[41], em que o autor propõe uma linguagem de consulta chamada GQuery baseada na XQuery.

Boucelma entende que um documento GML com milhares de feições chega a ser proibitivo devido ao aumento da complexidade e tamanho do documento GML. Assim torna-se necessário uma linguagem de consulta a GML com operadores espaciais que capture a semântica espacial no código GML. Linguagens usadas para consultar dados XML são dedicadas a manipulação de árvores e conseguem, somente, capturar feições alfanuméricas, nada ideais para o cálculo espacial.

A solução proposta pelo autor consiste na adição de operadores espaciais à XQuery para capturar a semântica de um documento GML, ou seja, que a GQuery seja usada como mediador num sistema de integração geográfica, em total acordo com a recomendação proposta nesse trabalho.

## 5.2. O Estado da Arte em Armazenamento em SGBD

BRINKHOFF *et al*[42,43] comenta que o gerenciamento de dados espaciais em SIG tem maior importância do que em sistemas convencionais devido à alta complexidade dos objetos e consultas, além do excessivo volume de dados. Bancos Geoespaciais impõem restrições severas às arquiteturas de armazenamento e acesso (índices), com o objetivo de se obter o processamento de consultas eficientes.

O SGBD Relacional está melhor preparado para tratar dados convencionais por apresentar deficiências quando utilizado para armazenamento espacial, a começar pelo modelo utilizado, muito pobre para representar dados do tipo campo. Para entender o problema, BRINKHOFF define que o objeto espacial caracteriza-se por duas componentes: a espacial e a temática. A componente espacial refere-se à representação topológica do objeto no plano através de pontos, linhas e polígonos. Enquanto a componente temática caracteriza o objeto segundo as suas propriedades temáticas. Essas propriedades são qualitativas, como o uso da terra ou quantitativas, como a precipitação pluviométrica em uma região.

Devido à arbitrária complexidade dos objetos espaciais, é impossível construir um índice considerando informações completas sobre a extensão dos objetos. A solução adotada baseia-se no princípio de que o método de acesso não produz exatamente o resultado da consulta, mas apenas filtra, refina os dados gerando conjuntos menores a serem consultados e, a partir de processos mais custosos, obter o resultado. Diferente dos bancos convencionais, esse resultado, por sua vez, necessita ser guardado para análises posteriores.

ENGENHOFER[44] disserta sobre a necessidade de uma linguagem para consultas espaciais em banco de dados que preserve os conceitos da SQL, trate objetos espaciais em alto nível e incorpore operações e relacionamentos espaciais.

Duas regras são necessárias para a definição da linguagem de consulta e de apresentação (resultado): a linguagem deve abstrair as características do armazenamento e da implementação para conseguir maior semântica da informação; e o resultado de uma consulta espacial não pode ser apenas uma relação (tabela) sem expressão espacial.

Engenhofer define onze requisitos necessários para uma linguagem de consulta espacial que descobertos pelos sistemas convencionais. Dentre os quais, destacamos dois: a necessidade de mostrar o resultado de uma forma gráfica (a GML como resultado permitirá que as informações sejam renderizadas e apresentadas de uma forma

visual) e a inclusão de informações adicionais associadas ao contexto, essencial para a interpretação do resultado (ex. uma consulta sobre a distribuição de escolas em um bairro deve mostrar, além da posição das escolas, a divisão do bairro).

As ferramentas de interação com os SGBD possuem todo um aparato de funcionalidades para melhorar a visualização de tabelas, mas faltam funcionalidades para visualização de tipos mais complexos, como: mapas e imagens. O texto fala de várias funcionalidades que a linguagem de consulta e apresentação deve ter. A Graphical Presentation Language GPL[45] fornece ferramentas para a apresentação gráfica do resultado consultado.

O método de acesso mais comum entre os banco de dados espaciais é o RTree, que é aplicado às estruturas bidimensionais simples, formadas pelo retângulo envolvente dos objetos e organizadas em árvores, onde se utiliza este retângulo e obtém-se uma hierarquia de níveis sucessivos de objetos neles contidos.

Podem ser feitas otimizações com o intuito de melhorar a performance das consultas. A abordagem propõe reduzir a diferença entre o objeto real e sua representação no banco de dados e com isso diminuir o conjunto de objetos retornados na filtragem. Algumas abordagens utilizadas são: convex hull, 5-corner e a ellipse.

Outra solução é a decomposição, que consiste em fracionar os objetos geoespaciais em objetos menores habilitando ao processamento paralelo e, dependendo da consulta, descartar partes não relevantes do objeto. Essas são algumas decomposições: polígonos convexos, triângulos e trapezóides.

TIAN *et al* [46] avalia cinco estratégias de armazenamento da XML divididas em três grupos (Tabela 5.1). Essas avaliações levaram em conta a performance de consultas utilizando a linguagem XQuery. Na Tabela 5.2 listam-se as cinco estratégias.

**Tabela 5.1 – Classificação de grupos de abordagens de armazenamento da XML.**

<b>Tipo</b>	<b>Vantagem</b>	<b>Desvantagem</b>
<b>Sistema de Arquivos</b>	- Mais simples de ser implementado. - Não requer o uso de Banco de Dados.	- Muitos parses (análise) desnecessários. - Gasto excessivo de memória.
<b>SGBD Relacional</b>	- Facilidades inerentes ao uso do Banco de Dados.	- Necessário mapeamento entre os objetos do XML e do Banco de Dados. - O uso do SQL pode ocasionar sobrecargas indesejáveis.
<b>SGBD Orientado a Objetos</b>	- Mapeamentos mais naturais do que SGBD Relacionais.	- Pouca cultura, requer aprendizado.



**Tabela 5.2 – As cinco estratégias de armazenamento definidas por Tian.**

	Sistema de Arquivos	SGBD Relacionais			SGBD Orientado a Objetos
		DTD	EDGE	Atributo	
<b>Estrutura</b>	<ul style="list-style-type: none"> <li>- O próprio documento armazenado em diretórios.</li> <li>- Criou-se índices para elementos, atributos e marcações tornando a solução competitiva.</li> </ul>	<ul style="list-style-type: none"> <li>- Solução normalizada onde cada elemento gera uma tabela que captura o relacionamento dos nós pais com os nós filhos.</li> <li>- Nova tabela deve ser criada se houver grafo com ciclos.</li> </ul>	<ul style="list-style-type: none"> <li>- Solução desnormalizada onde o documento gera uma única tabela que implementa o grafo.</li> <li>- Índices podem ser construídos para aumentar a performance.</li> </ul>	<ul style="list-style-type: none"> <li>- Variação da solução Edge onde ocorre uma partição horizontal do grafo pelo atributo economizando espaço de armazenamento.</li> <li>- A consulta requer join entre elementos e atributos.</li> </ul>	<ul style="list-style-type: none"> <li>- Todos os elementos do documento XML são armazenados como um único objeto que implementa a estrutura do grafo do documento XML.</li> </ul>
<b>Conclusões</b>	<ul style="list-style-type: none"> <li>- Performace similar à solução de objetos, mas apresenta alto custo de gerenciamento dos índices.</li> <li>- Pode ser usada quando houver baixa frequência de atualização.</li> </ul>	<ul style="list-style-type: none"> <li>- Agrupamento dos elementos do documento XML .</li> <li>- Menor espaço utilizado.</li> <li>- Melhor performance em diferentes conjuntos de dados e consultas.</li> <li>- Melhor solução.</li> </ul>	<ul style="list-style-type: none"> <li>- Agrupamento de elementos semelhantes.</li> <li>- Pior performance em consulta relacionada a mais de um elemento.</li> <li>- Custo menor para reconstrução do documento que na solução de atributo.</li> </ul>	<ul style="list-style-type: none"> <li>- Agrupamento de elementos semelhantes.</li> <li>- Pior performance em consulta relacionada a mais de um elemento.</li> <li>- Ocupa menos espaço que a solução EDGE.</li> </ul>	<ul style="list-style-type: none"> <li>- Agrupamento de elementos usando a mesma ordem de documentos XML originais.</li> <li>- Reconstrução mais rápida do documento original.</li> <li>- Muitos joins para consultas.</li> </ul>

RUBERG *et al* [47] avalia diversas estratégias de armazenamento e propõe uma nova estratégia para armazenamento utilizando o banco de dados Objeto-Relacional. A mesma idéia foi proposta por KLETTKE[48] e o pelo formato de representação DOM[49] (*Document Object Model*) para o mapeamento entre as tecnologias XML e SGBD Objeto-Relacional.

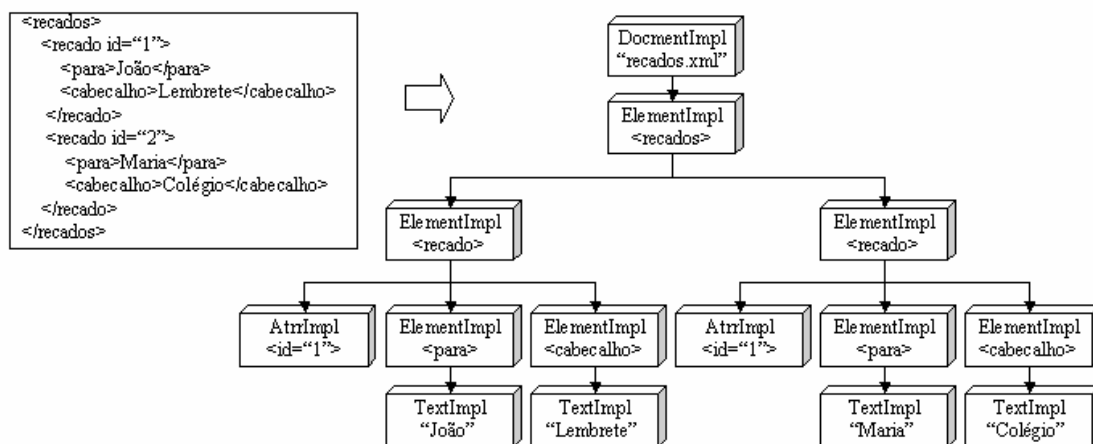
Ruberg, assim como outros autores [50][51], sugere a consulta através de uma conversão da XQuery para SQL3. As regras de conversão da XQuery em SQL3 foram escritas com a tecnologia XSLT.

A autora demonstra que o modelo de dados Orientado a Objetos oferece maior semântica para a representação de relacionamentos, através do uso de algoritmos específicos para consultas, que contêm expressões de caminho (path). Portanto, propõe uma arquitetura para o armazenamento de documento XML em um SGBD Objeto-Relacional e para execução transparente de consultas escritas na linguagem XQuery.

Ruberg explica que o mapeamento da XML em SGBD Relacional implica na criação de estruturas auxiliares que geram comandos SQL verborrágicos e um número excessivo de junções (*joins*), tornando a consulta extremamente custosa. Se for utilizado um banco Objeto-Relacional, haverá maior transparência entre os dois modelos e isso

reduz significativamente o impacto da transição, que, por sua vez, diminui o número de estruturas auxiliares e facilitando a tradução da linguagem de consulta.

Essa proposta baseia-se no armazenamento da XML no formato DOM.



**Figura 5.1 – Convertendo um documento XML em DOM.**

De acordo com a Figura 5.1, o documento XML é mapeado na Classe *DocumentImpl* e compresnde diversos documentos armazenados. Para atender as necessidades da IGD, mais informações poderiam ser armazenadas em conjunto, tais como: a data da recuperação desse documento, a fonte origem, o tipo de consulta que o originou etc. A Classe *ElementImpl* armazena os elementos do documento enquanto a classe *AttrImpl* guarda o atributo e seu valor. Finalmente, para armazenar o valor dos atributos usa-se a classe *TextImpl*.

Com o documento armazenado no formato DOM é preciso a transformação da XQuery em SQL3. Para tanto, Ruberg efetuou a tradução de todas as operações da XQuery com exceção das condicionais, isto por limitação do SQL3.

Para essa conversão foi preciso estender as formas de consultas através das expressões de caminho e criar regras de transformação para o SQL3. Uma consulta XQuery possui como estrutura básica uma expressão FLWR da forma FOR (declaração de cursores), LET (atribuição de cursores), WHERE (seleção) e RETURN (projeção dos resultados). Enquanto que, uma consulta SQL está no formato SELECT-FROM-WHERE. As regras de transformação são baseadas nas expressões de caminho e cursores são criados para executar este mapeamento. O número de cursores é definido

pelo comprimento de cada expressão de caminho. Para ilustrar a conversão da XQuery para SQL3, um exemplo é mostrado na Tabela 5.3:

**Tabela 5.3 – Exemplo de Conversão entre a XQuery e a SQL3.**

<pre> &lt;recados&gt;   &lt;recado id="1"&gt;     &lt;para&gt;João&lt;/para&gt;&lt;de&gt;Jane&lt;/de&gt;     &lt;cabecalho&gt;Lembrete&lt;/cabecalho&gt;     &lt;corpo&gt;Não se esqueça de mim no fim de semana.&lt;/corpo&gt;     &lt;data&gt;&lt;dia&gt;01&lt;/dia&gt;&lt;mês&gt;02&lt;/mês&gt;&lt;ano&gt;2001&lt;/ano&gt;&lt;/data&gt;   &lt;/recado&gt;   &lt;recado id="2"&gt;     &lt;para&gt;Maria&lt;/para&gt;&lt;de&gt;Jane&lt;/de&gt;     &lt;cabecalho&gt;Colégio&lt;/cabecalho&gt;     &lt;corpo&gt;Não se esqueça do dever de casa.&lt;/corpo&gt;     &lt;data&gt;&lt;dia&gt;01&lt;/dia&gt;&lt;mês&gt;02&lt;/mês&gt;&lt;ano&gt;2001&lt;/ano&gt;&lt;/data&gt;   &lt;/recado&gt;   &lt;recado id="3"&gt;     &lt;para&gt;Maria&lt;/para&gt;&lt;de&gt;João&lt;/de&gt;     &lt;cabecalho&gt;Almoço&lt;/cabecalho&gt;     &lt;corpo&gt;Hoje eu vou almoçar em casa.&lt;/corpo&gt;     &lt;data&gt;&lt;dia&gt;01&lt;/dia&gt;&lt;mês&gt;02&lt;/mês&gt;&lt;ano&gt;2001&lt;/ano&gt;&lt;/data&gt;   &lt;/recado&gt; &lt;/recados&gt; </pre>	<p><b>Consulta:</b> Retornar os dias de todos os recados do documento recados.xml.</p> <p><b>XQuery:</b></p> <pre> FOR \$n IN document ("recado.xml")/recados/recado/data/dia RETURN \$n </pre> <p><b>SQL3:</b></p> <pre> select resultado(resul:c5.value) from d1 in DocumentImpls, c1 in d1.childrenNode, c2 in c1.childrenNode, c2 in c1.childrenNode, c2 in c1.childrenNode, c2 in c1.childrenNode where (d1.name="recados") and (c2.name="recado") and (c3.name="data") and c4.name="dia") </pre> <p><b>Resultado:</b></p> <pre> { ["01"] ["01"] ["02"]} </pre>
---	--

Ruberg conclui, em seu trabalho, que existem diversas vantagens na utilização dessa proposta, dentre elas a integração com outras informações já armazenadas, o que é um dos objetivos deste trabalho. O uso do Objeto-Relacional torna a tradução da consulta para SQL mais fácil do que se fosse utilizado um banco puramente relacional. O uso do Formato DOM também permite recuperação total ou parcial do documento original, diferente das abordagens DTD e edge do artigo de TIAN [46].

Se os documentos XML utilizados fossem escritos em GML e a linguagem de consulta fosse a XQuery com a extensão GML-QL proposta por VATSAVAI[40] e implementados no banco Objeto-Relacional Oracle com extensão espacial, a proposta de Ruberg poderia ser estendida para atender as necessidades dos Sistemas de Informações Geográficas.

### 5.3. O Estado da Arte em Integração da Informação

O uso de mediadores pode ser uma solução para a consulta a banco de dados heterogêneos. LIMA[52] mostra a necessidade de integração de informações disponíveis em diversas fontes de dados, sejam elas banco de dados, servidores *Web* ou outros repositórios. As aplicações acessam uma representação uniforme das fontes de dados através de linguagens declarativas. Os mediadores encapsulam a representação das múltiplas fontes de dados para esta linguagem de consulta, fornecendo acesso uniforme.

O processamento da consulta é feito nas seguintes fases: Decomposição, onde uma consulta global é decomposta em outras consultas locais; Tradução, onde cada consulta local é traduzida na linguagem de consulta da fonte local; e Otimização, onde cada consulta local é ajustada individualmente, essa última é muito difícil de ser obtida devido a alguns problemas como: inconsistência dos dados, disponibilidade de informações do sistema local para o módulo otimizador global, como estatísticas de cardinalidade e seletividade.

O uso de mediadores permite que diferentes fontes de dados sejam adicionadas ao sistema e diferentes tradutores (wrappers) para que uma mesma fonte de dados sejam criados de acordo com a conveniência. Além de permitir a modularização ao invés da centralização, a qual é naturalmente suportada em ambientes distribuídos e presentes nos ambientes computacionais atuais.

FERREIRA[53] avança a utilização de SGBD objeto-relacional para armazenar documentos XML, por entender a necessidade de um sistema que gerencie grande quantidade de dados.

Entre as dificuldades de se manter os documentos XML em disco, o autor aborda as seguintes:

- Descobrir esquemas similares entre vários documentos XML,
- Encontrar um esquema comum que seja ótimo,
- Manipular evoluções de esquemas.

## CAPÍTULO 6 - RECOMENDAÇÕES PARA INTEGRAÇÃO SEMÂNTICA

*“Chegará um dia em que as máquinas  
pensem, porém elas nunca terão fantasias.”*

Theodor Heuss, presidente alemão

Neste capítulo será apresentado um conjunto de recomendações que poderá ser utilizado para atender o problema da integração semântica da informação geográfica distribuída. Para tanto, sugere-se a padronização dos seguintes aspectos: armazenamento da informação distribuída, forma de distribuição, armazenamento dos metadados e consulta. A seguir um novo cenário, com a adoção desta recomendação, é apresentado e analisado.

### 6.1. Ambiente integrado semanticamente

Com os conhecimentos apresentados no capítulo 2 e 3, e baseado no estudo descrito nos capítulos 4 e 5, propõe-se as seguintes recomendações para integração, apresentados na Tabela 6.1.

**Tabela 6.1 – Tabela de problemas e soluções propostas.**

<b>Problemas</b>	<b>Recomendação Proposta</b>
Grande Volume de Dados	Armazenamento Padrão da IGD em SGBD usando as extensões espaciais
Diversos Tipos de Armazenamento de Dados	
Diversidade de Fontes de Informação	Formato de Distribuição Padrão com XML/GML
Formatos Heterogêneos de Dados	
Dependência da Estrutura e Localização dos Dados	
Falta de Estruturação dos Metadados	Formato de Metadados Padrão com RDF
Descasamento dos Dados	
Diversidade de Linguagens de Consulta	Linguagem de Consulta Padrão com XQuery

## **6.2. Armazenamento Padrão**

Para se obter a integração semântica é preciso alcançar a interoperabilidade semântica e, dessa forma, deve-se ter transparência de nomes e linguagens, processamento de informações sensíveis ao contexto, regras para mecanismos de interação e correlação semântica. Esses requisitos podem ser obtidos utilizando a GML em conjunto com a RDF. Então, a GML deixa de ser apenas um meio de distribuição e passa a ser um elemento essencial para a integração da informação.

Portanto, a escolha da forma de armazenamento torna-se importante para essa integração. Segundo este raciocínio, ter uma única forma de armazenamento garante maior interoperabilidade. Entretanto, observam-se dois universos: um local em que as informações estariam em banco de dados e outro distribuído, em que as informações podem estar em diversos formatos e formas de armazenamento.

Apesar de, atualmente, atender somente aos SIG(S) descritos com Feições Simples da OGC, o uso do SGBD, por trazer diversas vantagens, é a melhor estratégia para armazenamento padrão. E tem maior eficiência, devido aos índices espaciais, quando utilizado com as extensões espaciais. Conclui-se então que o armazenamento padrão permite maior integração entre a IGD e IGL.

## **6.3. Formato de Distribuição Padrão**

A XML, que nasceu como forma de distribuição na *Web*, hoje é muito mais do que isso. Para se ter noção do benefício da XML, note que o resultado de uma consulta SQL é uma tabela e, dessa forma, os metadados, a estrutura e os relacionamentos não acompanham este resultado. Se a XML fosse usada neste caso, estes a acompanhariam. Por isso, usar a XML aumenta a integração não só entre a IGL e a IGD pelo sistema local, mas qualquer outro programa seria acrescido ao sistema sem muita alteração. Isto é, qualquer um teria como escrever um processador novo para os dados existentes e tratá-los sem alterar o legado. Essa é uma característica encontrada nas informações estruturadas (independência lógica de dados). Além disso, a XML permite a separação da informação do estilo e dos relacionamentos (links) entre os dados.

Criar elementos e atributos não é o melhor uso da XML. Esta é a ferramenta para escrever ontologias.

Mais importante do que criar marcações novas é criar uma nova linguagem, e a GML é a nova linguagem engendrada por um grupo formado por fornecedores de soluções de geoprocessamento. Isto é importante porque valida a linguagem, cuja escrita ocorre a partir de dois processos: DTD, como foi concebida a GML 1.0 ou XMLSchema, como foi concebida a GML 2.0 e 3.0. Usando XSL, a GML transforma-se facilmente para outros formatos objetivando a visualização ou mesmo a simples conversão. Os relacionamentos podem ser escritos com XLink.

#### **6.4. Formato de Metadados Padrão**

A XML puramente sem o RDF é capaz de armazenar os metadados relativos à estrutura do documento e alguns níveis de relacionamento. Mas com o uso do RDF os metadados de várias fontes serão integrados somente com combinação de linhas.

Informações de quando, como e onde a consulta foi realizada podem ser armazenadas e recuperadas para aumentar a semântica da IGD.

O maior benefício do uso da RDF será as regras de inferência que permitirão que a IGD tenha restrições de integridade tão poderosas quanto os SGBD. Por isso, o uso do RDF para armazenamento dos metadados permitirá a maior semântica das consultas às informações locais e distribuídas e, em consequência haverá aumento da semântica da IGD.

Os usuários podem enviar consultas em função do esquema global RDF e um sistema mediador deve conter um módulo que usa a descrição de recursos para traduzir consultas do usuário em consultas que se refiram diretamente aos esquemas GML.

É possível estabelecer correspondência entre cada recurso e o esquema global usando instâncias RDF, com isso será possível integrar recursos GML com outros recursos como HTML, PDF etc.

#### **6.5. Linguagem de consulta padrão**

Quando se armazena informações em Banco de Dados convencionais existem diversos artifícios adicionais para fazer essa consulta de forma eficiente. Porém, consultar uma informação fora do banco de dados exige um esforço maior, se as linguagens utilizadas não são as mesmas. Por isso, para integrar a IGL com a IGD

torna-se necessário uma padronização da linguagem de consulta. Não adianta armazenar qualquer informação se não há meios eficientes de recuperá-la.

Existem muitos esforços das comunidades da *Web* e de Banco de Dados na padronização de uma linguagem de consulta a documento XML. Por outro lado, o OGC não recomenda uma linguagem específica para consulta a documentos GML. A linguagem de consulta mais usada nos banco de dados é a SQL, enquanto a linguagem de consulta mais usada em informações armazenadas com XML é a XQuery. Porém, a XQuery por ser uma linguagem bem mais recente, posterior a Internet, é melhor preparada para atender aos requisitos atuais de integração, podendo ainda ser estendida para atender às necessidades da GML. Além disso, não existe ainda uma padronização das formas de consultas quando utilizamos as funcionalidades geoespaciais dos bancos. Isto é, o SQL ainda não está preparado para a consulta às informações geoespaciais. Por sua vez, a XQuery atenderia tanto as informações armazenadas em SGBD quanto as armazenadas em documento XML. Esta deve ser a linguagem indicada para a padronização, pois já existem trabalhos que estendem a XQuery para consultar a GML.

## 6.6. Cenário utilizando a recomendação proposta

Usando a recomendação proposta, descrevemos um cenário de um ambiente geoespacial, apresentado na Figura 6.1, no qual a integração foi pensada desde o desenvolvimento.

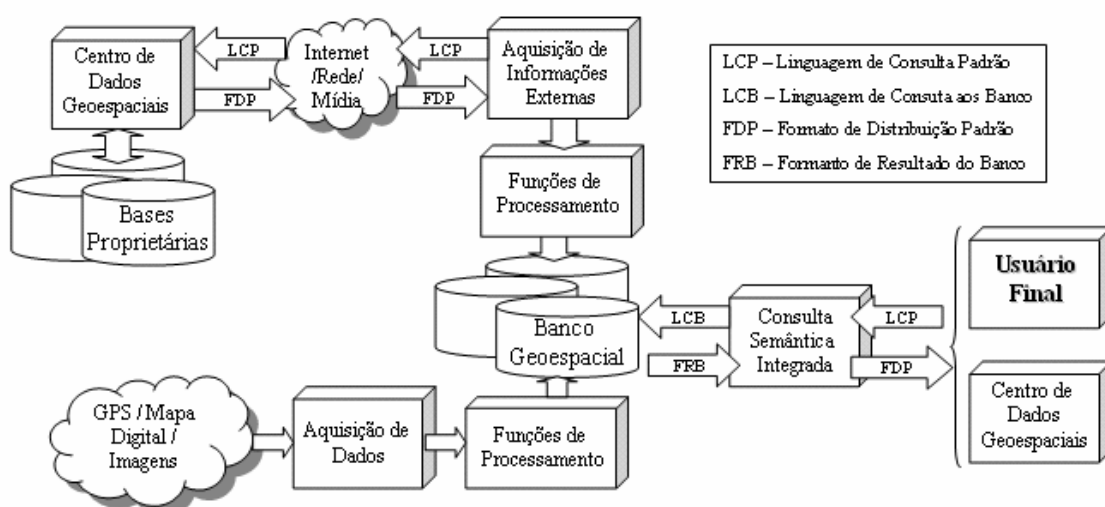


Figura 6.1 – Cenário utilizando a recomendação de integração semântica.



### 6.6.1. Consulta Semântica Integrada

- **Descrição:** é o processo responsável por consultar a informação geoespacial. Tanto a Informação Geográfica Local quanto a Informação Geográfica Distribuída serão consultadas usando a mesma Linguagem de Consulta ao Banco (LCB), que retornará no Formato de Resultado do Banco (FRB). A consulta será feita usando a Linguagem de Consulta Padrão (LCP) pelo usuário final ou agora, pela possibilidade de integração, por outro Centro de Dados Geoespaciais. O resultado será entregue no Formato de Distribuição Padrão (FDP).

- **Caracterização:** os dados são solicitados diretamente dos Centros de Dados ou dos resultados armazenados no banco de dados geoespaciais. Com os metadados armazenados junto com o formato de distribuição padrão, é possível manter consultas com maior significado dos dados.

- **Exemplo:** no novo Sistema de Monitoração Imobiliária, os dados do DNIT serão recebidos no formato padrão GML, que poderão trazer a estrutura semântica RDF internamente e armazená-los no SGBD Oracle utilizando a opção Oracle Spatial. A consulta a essas informações será feita usando as extensões espaciais do SQL3. O usuário, através de uma interface gráfica, consultará por informações sobre imóveis próximos a estrada. O módulo de consulta inferirá, através de regras armazenadas em RDF, que postos de gasolina não precisarão fazer parte dessa resposta. A consulta será realizada utilizando a XQuery com extensão geoespacial e receberá o resultado no formato GML. Por ter interesse nessa região, o Estado de São Paulo também consultará informações do Sistema de Monitoração Imobiliária diretamente com a XQuery estendida. Com essa linguagem, há como armazenar o critério de pesquisa e as informações sobre a consulta.

## CAPÍTULO 7 - CONSIDERAÇÕES FINAIS

*“Uma idéia não aparece uma ou duas vezes  
no mundo, mas milhões de vezes”.*

Aristóteles

### 7.1. Conclusão

Os estudos aqui realizados mostraram que a principal recomendação para alcançar a integração semântica entre a Informação Geográfica Local e a Informação Geográfica Distribuída é pensar antecipadamente neste problema e padronizar o sistema para melhor integração futura.

Este trabalho oferece um panorama geral do problema da integração semântica e serve como obra de referência. Seu público-alvo são usuários de Geotecnologias, em todos os setores de aplicação, bem como desenvolvedores de sistemas e aplicações, que quiserem ter uma visão do estado da arte dos problemas para a integração semântica.

Em pesquisa às tendências e soluções do mercado para a integração semântica da informação geoespacial concluiu-se:

- a interoperabilidade semântica é o arcabouço da integração semântica e, para isso, é preciso conseguir a integração sintática e a estruturação da informação.
- a XML permite que as informações geográficas sejam distribuídas com maior eficiência com documentos abertos, já que isso não ocorre só entre computadores, mas entre programas e pessoas;
- a GML torna-se uma ontologia para informações geoespaciais que juntamente com a RDF permitirão a estruturação semântica;
- uma vez recebido os documentos com a GML e RDF, o armazenamento em banco de dados torna-se imprescindível devido ao volume típico dos dados geográficos;
- os bancos de dados evoluíram, e, com isso, surgiu as extensões espaciais, porém não há padrão para consulta espacial e a SQL não é suficiente para responder aos requisitos dos SIG(S) com integração semântica;
- tanto o Oracle Spatial, quanto o PostgreSQL com PostGIS e a extensão espacial do MySQL são utilizados para armazenamento da GML, sendo a solução Oracle a mais completa, porém a mais cara.

- a XQuery estendida permitirá tanto consultar a informação local quanto a distribuída.

Essas tendências foram pesquisadas e analisadas a partir de dois cenários discutidos no capítulo 4, baseados na evolução da geomática, e um terceiro, proposto no capítulo 6, que apresenta um modelo da recomendação proposta.

Entendeu-se no trabalho a importância de usar a GML/XML como formato de distribuição da Informação Geográfica Distribuída, e respostas às consultas e de explorar as funcionalidades geoespaciais dos bancos de dados para viabilizar essas consultas a grandes volumes de dados.

## **7.2. Sugestões e recomendações**

Como propostas para trabalhos futuros, espera-se que esse trabalho se torne uma nova linha de investigação e seja o primeiro de uma gama de estudos feitos a partir da pesquisa e implementação das recomendações propostas.

Segue a lista de sugestões abaixo:

- Implementar um módulo que permita a consulta a banco de dados geoespacial utilizando a XQuery estendida para atender as operações espaciais, que ofereça como resultado as informações geográficas no formato de um documento GML.
- Implementar um módulo que receba o documento GML e armazene em um banco utilizando as funcionalidades geoespaciais.
- Estender a XQuery para tratar não só a consulta como a atualização da informação.
- Estudar a performance de consultas quando o armazenamento da GML é realizado em banco de dados.

## REFERÊNCIAS

- [1] FONSECA F., EGENHOFER M., BORGES K. A. V.: “Ontologias e Interoperabilidade Semântica entre SIGs”, II Workshop Brasileiro de GeoInformática, 2000.
- [2] KAVOURAS M., "A Unified Ontological Framework for Semantic Integration", International Workshop on Next Generation Geospatial Information, Cambridge (Boston), Massachusetts, USA, 19-21, 2003.
- [3] CÂMARA G., THOMÉ R., FREITAS U., *et al*, “Interoperability In Practice: Problems in Semantic Conversion from Current Technology to OpenGIS”, 2nd International Conference on Interoperable Operating Systems, Zurique, 1999.
- [4] OGC - OPEN GEOSPATIAL CONSORTIUM. GML Geographic Markup Language. Disponível em: <http://www.opengeospatial.org/specs/?page=specs.>, Acesso em: 20 mar. 2004.
- [5] OGC - OPEN GEOSPATIAL CONSORTIUM. Vision & Mission. Disponível em: <http://www.opengeospatial.org/about/?page=vision>. Acesso em: 20 mar. 2004.
- [6] CÂMARA, G. et al. *Anatomia de Sistemas de Informação Geográfica*. Campinas, Instituto de Computação, Unicamp, 1996. p.193.
- [7] W3C - WORLD WIDE WEB CONSORTIUM. Extensible Markup Language (XML) 1.0 (Third Edition), Disponível em: <http://www.w3.org/TR/2004/REC-xml-20040204/>, Acesso em: 4 ago. 2004.
- [8] THOMÉ, R. Interoperabilidade em Geoprocessamento: “Conversão entre Modelos Conceituais de Sistemas de Informação Geográfica e Comparação com o Padrão OpenGIS”. São José dos Campos, SP: [s.n], 1998.
- [9] OUKSEL A. M., SHETH A., “Semantic Interoperability in Global Information System. A Brief Introduction to the Research Area and the Special Section.”, SIGMOD, vol 28, p 5-12, 1999.
- [10] W3C - WORLD WIDE WEB CONSORTIUM. The About the World Wide Web Consortium (W3C), Disponível em: <http://www.w3.org/Consortium>, Acesso em: 13 jun 2004.
- [11] KEVIN D., XML: A Manager’s Guide, 2 ed, Massachusetts, Addison-Wesley, 2000. 184p.
- [12] LIGHT R., *Iniciando em XML*, 1ª ed., São Paulo, Makron Books, 1999. 404p.

- [13] MURRAY P., “An introduction to Structured Documents”, Disponível em: <http://www.venus.co.uk/omf/cml/doc/tutorial/xml.html>., Acesso em: 20 set. 2003.
- [14] W3C - WORLD WIDE WEB CONSORTIUM. “XQuery 1.0: An XML Query Language”, Disponível em: <http://www.w3.org/TR/2005/WD-xquery-20050915/>., Acesso em: 20 jun. 2005.
- [15] BAKKER B., WIDARTO I., “An Introduction to XQuery” Disponível em: <http://www.perfectxml.com/articles/xml/xquery.asp> Acesso em: 20 mar. 2005.
- [16] MOELLER A., “XQuery Tutorial.” Disponível em: <http://www.brics.dk/~amoeller/XML/querying/>, Acesso em: 20 mar. 2005.
- [17] W3C - WORLD WIDE WEB CONSORTIUM. Resource Description Framework (RDF). Disponível em: <http://www.w3.org/RDF/> Acesso em: 15 fev. 2005.
- [17a] MCBRIDE B., “Jena: Implementing the RDF Model and Syntax Specification” Disponível em: <http://www.hpl.hp.com/personal/bwm/papers/20001221-paper/>, Acesso em: 20 mar 2005.
- [18] GOAD C., “Describing Computation within RDF” Disponível em: <http://fabl.net/papers/swws.html>, Acesso em: 7 jul. 2005.
- [19] RFC - REQUEST FOR COMMENTS – IETF INTERNET ENGINEERING TASK FORCE, Uniform Resource Identifiers (URI): Generic Syntax. Disponível em: <http://www.ietf.org/rfc/rfc3986.txt>., Acesso em: 11 jan 2006.
- [20] LAKE R., “Introduction to GML – Geography Markup Language”, Disponível em: <http://www.w3.org/Mobile/posdep/GMLIntroduction.html>, Acesso em: 30 out. 2005.
- [21] LAKE R., “Making Maps With Geography Markup Language (GML)” Disponível em: <http://www.galdosinc.com/files/MakingMapsInGML2.pdf>, Acesso em: 20 mar. 2005.
- [22] SFA - Implementation Specification for Geographic information - Simple feature access Part 1:Common architecture, Disponível em: <http://www.opengeospatial.org/specs/?page=specs>., Acesso em: 20 mar. 2005.
- [23] W3C - WORLD WIDE WEB CONSORTIUM. The W3C Abstract Specification – Disponível em <http://www.w3.org>. Acesso em: 12 ago. 2004.
- [24] SVG – Scalable Vector Graphics (SVG), Disponível em: <http://www.w3.org/Graphics/SVG/>, Acesso em: 20 mar 2005.

- [25] VML - The Vector Markup Language, Disponível em: <http://www.w3.org/TR/1998/NOTE-VML-19980513>, Acesso em: 20 mar 2005.
- [26] VRML Virtual Reality Modeling Language, Disponível em: <http://www.w3.org/MarkUp/VRML>, Acesso em: 20 mar. 2005.
- [27] ORACLE SPATIAL, User Guide and Reference, Disponível no CD de Documentação da Oracle Release .9.2.
- [28] MYSQL, “Spatial Extensions in MySQL”, MySQL AB. Disponível em <http://dev.mysql.com/doc/refman/5.1/en/spatial-extensions.html>, Acesso em: 20 mar 2005.
- [29] POSTGRESQL, “Tutorial do PostgreSQL 7.3.2”, Disponível em: <http://www.postgresql.org>., Acesso em: 20 mar 2005
- [30] POSTGIS, “PostGis Manual”, Disponível em: <http://postgis.refrations.net>., Acesso em: 20 mar 2005.
- [31] W3C - WORLD WIDE WEB CONSORTIUM. Semantic Web. Disponível em: <http://www.w3.org/2001/sw/>, Acesso em: 20 mar 2005.
- [32] BERNERS-LEE T., MILLER E., LASSILA O., “The Semantic Web”, Scientific American, vol. 184, no. 5, pp. 34-43, 2001. Disponível em: <http://www.scientificamerican.com/>. Acesso em: 20 mar. 2005.
- [33] BERNERS-LEE T., MILLER E., “The Semantic Web lifts off”, Disponível em: [http://www.ercim.org/publication/Ercim\\_News/enw51/bernner-lee.html](http://www.ercim.org/publication/Ercim_News/enw51/bernner-lee.html)., Acesso em: 20 mar. 2005.
- [34] MATTHEWS B., WILSON M., BRICKLEY D., “Semantic Web Advanced development in Europe”, Disponível em: [http://www.ercim.org/publication/Ercim\\_News/enw51/matthews.html](http://www.ercim.org/publication/Ercim_News/enw51/matthews.html), Acesso em: 20 mar 2005.
- [35] BERNERS-LEE T., CONNOLLY D., SWICK R.R., “Web Architecture – Describing and Exchanging Data” Disponível em: <http://www.w3.org/1999/04/WebData.html>., Acesso em: 20 mar 2005.
- [36] MACKAY D. S., “Semantic Integration of Envirommental Models for Application to Global Information Systems and Decision-Making” SIGMOD, vol 28, p 13-19, 1999.

- [37] EGENHOFER M. J., "Toward the Semantic Geospatial Web". Disponível em: [http://www.ercim.org/publication/Ercim\\_News/enw51/matthews.html](http://www.ercim.org/publication/Ercim_News/enw51/matthews.html), Acesso em: 20 mar 2005.
- [38] CASEY M. J., AUSTIN M. A., "Semantic Web Methodologies for Spatial Decision Support" Disponível em [http://www.ercim.org/publication/Ercim\\_News/enw51/matthews.html](http://www.ercim.org/publication/Ercim_News/enw51/matthews.html), Acesso em: 20 mar 2005.
- [39] HENDLER J., BERNERS-LEE T., MILLER E., "Integrating Applications on the Semantic Web", Journal of the Institute of Engineers of Japan, Vol122(10), p. 676-680, 2002.
- [40] VATSAVAI R.R., "GML-QL: A Spatial Query Language Specification for GML", Minnesota, UCGIS, 2002. Disponível em: <http://www.cobblestoneconcepts.com/ucgis2summer2002/vatsavai/vatsavai.html>, Acesso em: 10 mar. 2005.
- [41] BOUCELMA O., COLONNA F., "GQuery: a Query Language for GML," 24° Urban Data Management Symposium, Chioggia-Venice, Italy, October, 2004.
- [42] BRINKHOFF T., HORN H., KRIEGEL H. P., "A Storage and Access Architecture for Efficient Query Processing in Spatial Database Systems",
- [43] BRINKHOFF T., KRIEGEL H. P., SCHNEIDER R., "Efficient Spatial Query Processing in Geographic Database Systems", Data Engineering Bulletin, vol 16, n 3, p 10-15, 1993.
- [44] EGENHOFER M., "Spatial SQL: A Query and Presentation Language", IEEE Transaction on Knowledge and Data Engineering, vol. 6, n. 1, pp. 2-4, 1994.
- [45] EGENHOFER M., "Extending SQL for cartographic display," Cartography and Geographic Information System, 1991 (in press). Disponível em: <http://www.spatial.maine.edu/~max/CartSQL.pdf>, Acesso em: 20 mar. 2005.
- [46] TIAN F., DEWITT D. J., CHEN J., *et al*, "The Design and Performance Evaluation of Alternative XML Storage Strategies", 2000, Disponível em: <http://www.csd.uwo.ca/courses/CS853a/storage.pdf>, Acesso em: 20 mar 2005.
- [47] RUBERG G., VIERA H., MATTOSO M.: "Xverter: Armazenamento e Consulta de Dados XML em SGBDs". SBBD – XVII Simpósio Brasileiro de Banco de Dados, 2002.
- [48] KLETTKE M., MEYER H., "Managing XML documents in object-relational databases", Lecture Notes in Computer Science, vol 1997, p 151, 2001.

- [49] DOM – Document Object Model Requirements, Disponível em <http://www.w3.org/DOM/drafts/requirements.html>. Acesso em: 20 mar 2005.
- [50] RUNAPONGSA K., PATEL J. M., “Storing and Querying XML Data in Object-Relational DBMSs”, Disponível em: <http://citeseer.ist.psu.edu/cache/papers/cs/28940/http:zSzzSzwww-personal.umich.edu/~krunaponzSzresearchzSzXORator.pdf/runapongsa02storing.pdf>, Acesso em 20 mar. 2005.
- [51] MANOLESCU, I., FLORESCU, D e KOSSMANN, D.:”Answering XML Query Over Heterogeneous Data Sources”. WebDB, pp. 1-25, 2000.
- [52] LIMA F., MELO R. N. “Consultas em Sistemas Gerenciadores de Bases de dados Heterogêneos”, Disponível em: [ftp://ftp.inf.puc-rio.br/pub/docs/techreports/99\\_19\\_lima.pdf](ftp://ftp.inf.puc-rio.br/pub/docs/techreports/99_19_lima.pdf), Acesso em: 20 mar 2005.
- [53] FERREIRA K., QUEIROZ G., PAIVA J., *et al*: “Arquitetura de Software para construção de Banco de Dados Geográficos com SGBD Objeto-Relacionais”. SBBD – XVII Simpósio Brasileiro de Banco de Dados, 2002.