

## **Estudo do XML, GML, SVG e WEBSERVICES (WMS e WFS) para formatação e divulgação de informações geográficas.**

Fábio Brandão<sup>1</sup>  
João Araújo Ribeiro<sup>1</sup>

<sup>1</sup>Universidade do Estado do Rio de Janeiro (UERJ), PÓS-GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO - GEOMÁTICA. Rua São Francisco Xavier, 524, 5º andar, - Cep: 20550-013- Rio de Janeiro – RJ-Brasil.  
fabios77@ig.com.br  
araujo@eng.uerj.br

**Abstract.** The technologies XML, GML, WMS, WFS and SVG became an active research area with many promising applications in Geotechnologies solutions. This paper gives a concrete contribution to the adoption of those open formats and standardized in GIS, what enable the use of geographical information on a Web browser and fosters data reuse throughout the Internet.

**Palavras-chave:** estudo, xml, gml, webservices, wms, wfs.

### **1. Introdução**

Com o aumento do uso e desenvolvimento da internet, os profissionais de GEOMÁTICA viram possibilidades de efetuar negócios via rede mundial e com isso agilizar e reduzir custos nos processos de tratamento e divulgação de informações geográficas. Entretanto, as trocas desses dados por meios digitais encontravam barreiras, já que estes não possuíam um padrão comum, o que desencadeava um ciclo de incompatibilidades, a nível de ambiente operacional e a nível de plataforma. O presente estudo baseia-se nestas dificuldades de comunicação de dados, para apresentar o uso de tecnologias multi-plataforma e multi-ambiente, as quais possam permitir a interoperabilidade de dados geográficos, para que sejam disseminados por toda a rede.

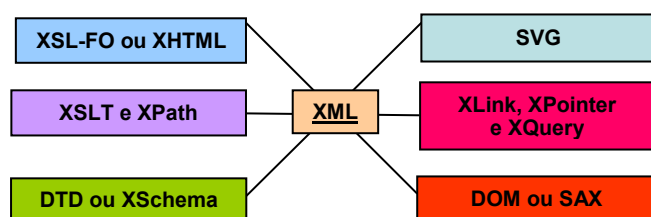
Essas tecnologias devem conter as informações necessárias para descrever os tipos de dados que carregam, assim sendo, este estudo teve como base de desenvolvimento a linguagem de marcação XML. Este caminho foi definido devido às características peculiares dessa linguagem, que provê meios para que dados geográficos sejam padronizados, permitindo sua divulgação independente do ambiente e da plataforma na qual sejam consultados, para tanto foram pesquisadas tecnologias adicionais como GML, SVG, WEBSERVICES (WMS e WFS) que são derivadas da linguagem XML, com o intuito de criar sistemas de informações geográficas voltados para WEB, e o importante diferencial de poderem se comunicar com outros sistemas, independente do formato ou da fonte do dado.

### **2. XML**

O eXtensible Markup Language é um padrão W3C (world wide web consortium) grupo responsável por diversos padrões existentes na internet. O uso de XML como formato para a troca de dados busca uma linguagem de padrão aberto que tenha a facilidade de ser convertida para outros formatos, inclusive formatos proprietários, além disso, o XML é texto, o que contribui para sua fácil leitura e processamento, gerando menos incompatibilidades. Como estamos falando de dados geográficos, devemos levar em conta as necessidades destes, serem descritos por relacionamento semântico, esta é a principal característica da linguagem, que foi definitivamente um dos principais pontos na escolha do seu estudo, possibilitando mecanismos de busca muito mais eficientes que outras linguagens e por sua vez aumentando o reaproveitamento dessas informações.

Para que possa ser manipulado como uma árvore semântica, o documento XML deve ser bem formado, seguindo uma gramática que corresponde um apanhado de regras para um conjunto de documentos XML. Essas regras são definidas por métodos como, o XML Schema e o DTD e devem ser os primeiros argumentos a serem estudado, só depois, vamos seguir para o processo de validação dos dados, que estará dentro das nossas aplicações de sistemas de informações geográficas. Nesse momento o XML será limitado por um vocabulário, onde os atributos terão tipos e valores previamente definidos, obedecendo a uma estrutura hierárquica, padronizada pela linguagem GML e utilizados por Webservices voltados para divulgação de dados geográficos.

## 2.1. Mapeamento do XML



**Figura 1 - Mapeamento do XML**

Mapeando o XML, encontramos várias estruturas que serão úteis ao nosso estudo, como é o caso do XSL-FO ou XHTML necessário para formatar dados para impressão ou visualização na tela (PDF, WORD, WEB), o XSLT e XPATH, para transformar os dados em outros formatos, o DTD ou XSCHEMA, que especifica o modelo de dados validando as informações. O XLINK, XPOINTER e XQUERY são responsáveis por criar vínculos lógicos entre os documentos e localizar seus componentes, já o DOM ou SAX geram e extraem dados dos documentos, além de ler e gravar em bancos de dados e por último o SVG, para gerar informações e formato de gráficos vetoriais.

## 3. GML

A Geographic Markup Language (GML) é uma codificação XML para transporte e armazenamento de informação geográfica, incluindo suas propriedades espaciais e não espaciais. O objetivo da GML é oferecer um conjunto de regras com as quais um usuário passa a definir sua própria linguagem para descrever seus dados. A GML é codificada a partir de XML Schemas baseados em modelos geométricos, de feições e de superfícies. Os principais esquemas são:

O *BasicTypes*, que engloba uma série de componentes simples e genéricos para representação arbitrária de atributos, nulos ou não. O *Topology*, o qual especifica as definições do esquema geométrico dos dados, bem como sua descrição. O *Coordinate Reference Systems*, para sistemas de referência de coordenadas. O *Temporal Information and Dynamic Feature*, este esquema estende aos elementos características temporais dos dados geográficos e suas funções dinamicamente definidas. O *Definitions and Dictionaries*, definições das condições de uso dentro de documentos com certas propriedades ou informações referentes à propriedade padrão. O *Metadata*, este esquema é utilizado para definir as propriedades dos pacotes de dados que podem ser utilizados através de outros dados já existentes.

### 3.1. Regras do GML:

Desenvolvedores de esquemas de aplicação devem assegurar que seus tipos são subtipos dos correspondentes tipos da GML:

*gml:AbstractFeatureType* ou  
*gml:AbstractFeatureCollectionType* para feições

*gml:AbstractGeometryType* ou  
*gml:GeometryCollectionType* para a geometria.

Um esquema de aplicação não pode mudar o nome, definição ou tipo de dado dos elementos obrigatórios da GML. Definições de tipos abstratos podem ser livremente estendidas ou restritas. Esquema de aplicação deve estar disponível a qualquer um que receba o dado estruturado por aquele esquema. Os esquemas relevantes devem especificar um “namespace” que não deve ser: <http://www.opengis.net/gml>.

### 3.2. Definição Inicial do GML

```
<schema targetNamespace="http://www.opengis.net/gml"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  elementFormDefault="qualified"
  version="2.06">
  <import namespace="http://www.w3.org/1999/xlink"
    schemaLocation="xlinks.xsd">
```

Figura 2 – Definição Inicial do GML

### 3.3. O Framework

Define os objetos abstratos *Feature* e *Geometry* que iniciam a hierarquia de tipos de GML. Todas as aplicações têm seus principais objetos derivados desses tipos.

### 3.4. Codificando Geometrias

Implementa geometrias, com as seguintes classes:

- *Point*;                               - *LineString*;
- *LinearRing*;                       - *Polygon*;
- *MultiPoint*;                       - *MultiLineString*;
- *MultiPolygon*                      - *MultiGeometry*;

Possui ainda os elementos *<coordinates>* e *<coord>* para codificação de coordenadas.

*LineString* - possui as coordenadas do primeiro e último ponto;

*LinearRing* - a última coordenada deve coincidir com a primeira;

*Polygon* - o limite do *Polygon* é um conjunto de *linearRings*;

*MultiGeometry*-contém todos os elementos geométricos, possui a propriedade *geometryMember* que retorna o próximo elemento geométrico na coleção.

### 3.5. Feature (feição)

Feição é o átomo da informação geográfica. É uma representação abstrata dos fenômenos do

mundo real e está associada a uma localização relativa à Terra.

As feições geográficas contêm informações de sua posição em relação às coordenadas da Terra. A técnica mais comum para descrever a forma e localização de uma feição é através de sua geometria.

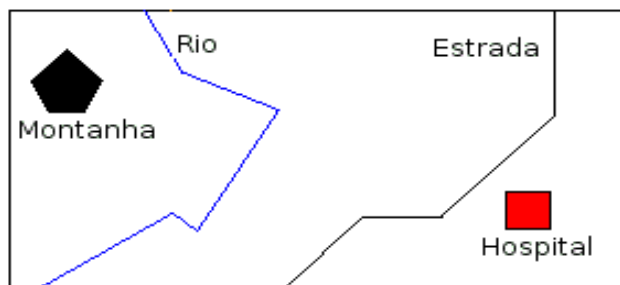
```
...
1.<complexType name="hidrografia">
2.  <complexContent>
3.    <extension base="gml:AbstractFeatureType">
4.      <sequence>
5.        <element ref="gml:centerLineOf"/>
6.      </sequence>
7.    </extension>
8.  </complexContent>
9.</complexType>
...
```

**Figura 3 – Arquivo exemplo, xsd**

A Figura 2 é um fragmento de um arquivo “exemplo.xsd” que define um esquema de aplicação, mostrando a criação de um tipo, no caso hidrografia. Seguindo as regras, a linha 3 faz com que hidrografia seja subtipo de *gml:AbstractFeatureType*. Este tipo pode ser usado na criação de uma tag, por exemplo:

```
...
<element name="rio" type="ex:hidrografia" substitutionGroup="gml:_Feature"/>
...
```

**Figura 4 – Tag**



**Figura 5 - Representação Geométrica de uma cidade.**

- A cidade é uma extensão do tipo GML *AbstractFeatureCollectionType*;
- Rio, rodovia, montanha e hospital são do tipo GML *AbstractFeatureType*;
- A delimitação da cidade deve conter o sistema de referência espacial (SRS) e as informações geométricas de todas as *features* devem encontrar-se também nesse sistema;
- O rio é uma propriedade geométrica do tipo *LineString*;
- A estrada é uma propriedade geométrica do tipo *LineString*;
- A montanha é uma propriedade geométrica do tipo *LineRing*;
- O hospital é uma propriedade geométrica do tipo *Point*.

#### 4. SVG

Muitas vezes os dados em GML precisam ser convertidos para o padrão SVG nas aplicações, para fazer a apresentação gráfica dos dados espaciais. Geralmente utiliza-se XSLT para fazer

a conversão.

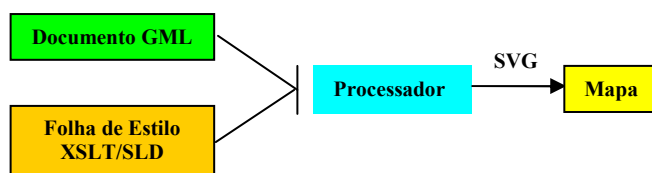


Figura 6 - Transformação de um documento GML para um mapa SVG

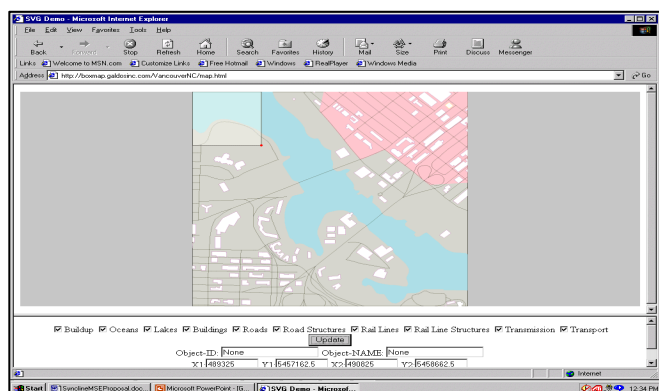


Figura 7 - Visualização de um mapa vetorial num *web browser* utilizando SVG

## 5. Web Services

Os *Web Services* são softwares que oferecem serviços a aplicações remotas, denominadas consumidoras, utilizando a Internet como canal de comunicação. As aplicações consumidoras e os Web Services podem ser desenvolvidos em diferentes plataformas, utilizando diferentes linguagens de programação, o que é permitido pelo uso de um conjunto de padrões, tais como XML, SOAP, HTTP, WSDL. Além disso, uma aplicação pode utilizar simultaneamente, recursos de Web Services localizados em diferentes servidores [2].

### 5.1 Arquitetura *Web Services*

Provedor de serviços - Oferece serviços, alguns dos quais podem ser *Web Services*.

Registro de serviços - Catálogo de endereços: repositório central que contém informações sobre *Web Services*.

Cliente de serviços - Aplicação que descobre um *Web Service*, implementa sua interface de comunicação e usa o serviço.

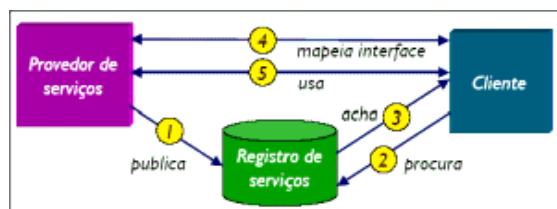
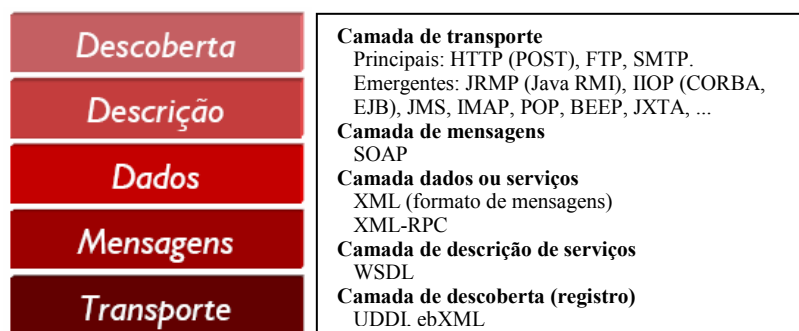


Figura 8 - Arquitetura *WebService*



**Figura 9 – Camadas da arquitetura *Web Service***

As mensagens trocadas são formatadas no protocolo SOAP (*Simple Object Access Protocol*), o que permite a interoperabilidade entre diferentes plataformas, em um processo denominado serialização XML. Porém, antes que as mensagens SOAP sejam trocadas, suas características são explicitadas através de documentos WSDL (*Web Services Description Language*), que descrevem quais dados estarão sendo trocados, e como estes dados estarão organizados nas mensagens SOAP. Adicionalmente, os serviços dos *Web Services* podem ser publicados através de UDDI (*Universal Description, Discovery and Integration*), que é um formato utilizado para seu armazenamento em repositórios disponíveis na Internet. Assim, se um desenvolvedor precisar resolver uma determinada tarefa, pode encontrar o *Web Service* que mais se adequar à sua necessidade [2].

### 5.2. *Web Map Service* (WMS)

A função de WMS é criar uma forma padronizada para recuperação de mapas (imagens matriciais). Neste processo, os parâmetros utilizados para a definição exata de qual mapa recuperar são: a representação do mapa, o tamanho do mesmo e o seu sistema de referência. São definidas três operações :

**GetCapabilities:** provê metadados sobre o serviço, oferecendo informações sobre os valores de parâmetros aceitáveis para as demais operações.

**GetMap:** retorna um mapa para o cliente, com sistema de referência, tamanho, formato e transparências especificados.

**GetFeatureInfo:** operação opcional que retorna informações adicionais sobre pixels determinados pelo cliente, por meio de um mapa recuperado previamente.

### 5.3. *Web Feature Service* (WFS)

Esta especificação suporta a comunicação padronizada entre clientes e servidores de dados geográficos, permitindo a execução de consultas, inserções, atualizações e exclusões de feições geográficas. Para tanto, foram definidas as seguintes operações:

**GetCapabilities:** retorna um documento que descreve os tipos de feições suportados, bem como as operações permitidas por cada um destes tipos.

**DescribeFeatureType:** retorna a descrição de um determinado tipo de feição, suportado pelo WFS.

**GetFeature:** é responsável por retornar instâncias de dados oferecidos pelo serviço, representadas por documentos no formato GML.

Além destas, que definem um WFS básico, que é somente-leitura, existem ainda as seguintes operações, que definem um WFS transacional.

**Transaction:** permitem a realização de operações de inserção, atualização e exclusão sobre os dados mantidos sob o WFS.

**LockFeature:** é uma operação opcional, utilizada para bloquear um ou mais ítems de dados na utilização de um WFS transacional, permitindo a serialização.

A seqüência natural na utilização de *Web Services* que implementam a especificação WFS consiste de três passos. O primeiro é a utilização do método *GetCapabilities* pelo cliente, que retorna as capacidades do serviço. Em seguida, o cliente pode invocar o método *DescribeFeatureType* e ter detalhes de um determinado tipo de feição. O terceiro passo consiste de recuperar ou modificar dados, utilizando-se o método *Transaction* ou *GetFeature*.

## 6. Conclusões

A informação é a base para todos os setores da atividade humana, tanto no processo de planejamento como na tomada de decisões. Na área de geotecnologia, essa importância ganha dimensões ainda maiores. A busca de uma linguagem universal está ligada à necessidade de disseminação dessa informação para todos que a queiram usar. No trabalho apresentado podemos ver os bons resultados obtidos com os padrões da OGC (*Open Geospatial Consortium*) e que o GML (XML) desponta como uma linguagem padrão com um ótimo desenvolvimento. A linguagem GML já é uma realidade no atual contexto, entretanto devemos observar que existe a necessidade do Brasil em participar do desenvolvimento desses padrões para soluções de geoprocessamento deixando de ser um mero consumidor de tecnologia passando a ser um produtor da mesma.

## Referências

Bastos, Iochpe, Lima 2004, Transformação de GML para SVG em uma Clearinghouse para disponibilização de dados geográficos na Internet. II Workshop de Tecn. da Inf. aplicada ao Meio Ambiente – CBComp 2004.

FARIA, Rogério Amorim de. **Treinamento avançado em XML**. São Paulo: Digerati Books, 2005. 126 p.

GML specification. Disponível em: <<http://www.opengeospatial.org>>. Acesso em: 15 jul. 2006.

GML Dev Days. Disponível em: <<http://www.gmldev.org>>. Acesso em: 13 jul. 2006.

Brigitte Mathiak, Andreas Kupfer And Karl Neumann. Using Xml Languages For Modeling And Web-Visualization Of Geographical Legacy Data. Institut F`Ur Informationssysteme-TU Braunschweig M`Uhlenpfordtstr. 23, 38100 Braunschweig, Germany {Mathiak, Kupfer, Neumann}@Idb.Cs.Tu-Bs.De, 2003.

Open GIS Consortium, “Geography Markup Language (GML) 3.0,” Open GIS Implementation Specification, [Online]. Disponível em: <<http://www.opengis.org/docs/02-023r4.pdf>>. Acesso em 20 jan. 2006.

ROCHA, Helder da, XML na gestão da informação. Disponível em: <[www.argonavis.com.br](http://www.argonavis.com.br)>. Acesso em: 17 jul. 2006.

Schuenck e Soares. **Mecanismos Para Intercâmbio De Dados Geográficos Através De Web Services**. Departamento de Informática e Matemática Aplicada - UFRN, Caixa Postal 515, 201, Natal, RN, Brasil, michael@ppgsc.ufrn.br, valeria@dimap.ufrn.br,