

一种基于 GML 实现互操作 WebGIS 的新机制

方颖, 曹加恒, 曾承

(武汉大学计算机学院, 武汉 430079)

摘要: 提出了一种基于 GML 实现互操作 WebGIS 的新机制。该机制以 GML 实现多源数据存储与传输为基础, 利用 WFS 实现地理数据要素层存取、检索与更新, 在客户端以 SVG 方式输出高质量矢量图形。与传统的 WMS 实现 Web 服务相比较, 该机制能实时动态地实现异构、跨平台下地理数据要素层互操作。

关键词: GML; WFS; SVG; WMS; WebGIS

New Implementation Mechanism of Interoperable WebGIS Based on GML

FANG Ying, CAO Jiaheng, ZENG Cheng

(School of Computer, Wuhan University, Wuhan 430079)

【Abstract】 A new implementation method of interoperable WebGIS is proposed based on GML. The basic of it is using GML to encode geospatial data and transport geospatial data over the Web. This paper uses WFS as a data query mechanism to interactively access and retrieve geospatial data at the feature level on the Web, and SVG is used to output superior quality and flexibility vector maps on Web browser. Compared with WebGIS implementation using WMS, this new method can acquire and integrate geospatial data from heterogeneous sources, and dynamically implement interoperable WebGIS at the feature level in real time.

【Key words】 Geography markup language(GML); Web feature service(WFS); Scalable vector graphics(SVG); WMS; WebGIS

随着GIS技术和Internet技术的迅猛发展, WebGIS的应用越来越受关注。在实际应用中, 人们需要得到不同数据源的数据, 这些数据往往有着不同的数据模型、不同的数据格式与存储方式, 为了实现这些异构数据的共享, 跨平台进行交互操作, 一种基于XML的地理标记语言——Geography Markup Language(GML)得到广泛的研究和应用^[1]。

目前大部分基于GML的WebGIS互操作应用主要支持Web Map Service(WMS)功能, WMS定义的GetCapabilities、GetMap和GetFeatureInfo操作使客户端用户可以同时获取本地或远程不同地图服务器上多层地理数据, 并在客户端叠置生成需要的地图^[2]。

但是WMS只提供了一定的WebGIS互操作支持, 它通常只能提供标准的PNG、GIF、JPEG或SVG等格式的静态图像, 而且无法实现地理数据要素层的互操作。为了实现WebGIS异构、跨平台的要素层互操作, 同时在客户端得到高质量矢量地图, 本文以GML为基础, 提出了一种采用Web Feature Service(WFS)实现地理数据要素层的查询和提取, 同时以Scalable Vector Graphics(SVG)方式输出可视化结果的WebGIS互操作实现新机制。

1 互操作 WebGIS 的基础——GML

GML是开放式地理信息系统联盟(OGC)为了解决不同来源、不同内容、不同数据模型以及不同数据格式的空间数据之间的共享与互操作, 推出的一个基于OpenGIS“简单要素”(Simple Feature)数据模型和XML的空间数据交换格式, 主要用于空间数据的编码、存储和传输。

GML为不同格式数据之间的共享和转换提供了方便。在

实际应用中, 要求不同部门、不同行业、不同地区的空间数据以统一的格式存储往往是不可能的, 要想交互使用这些跨平台、异构的空间数据, 数据转换必不可少。通常空间数据以Shapfile文件形式存放, 或者存放在Oracle Spatial、PostGIS等数据库中, 它们的格式各不相同, 但是本质上都是由元数据和空间数据本身组成, 都是基于OpenGIS“简单要素”数据模型。“简单要素”数据模型又是GML存储空间数据的基础, 这保证了GML实现不同格式空间数据的共享和转换。

GML分层组织空间数据, 根据具体的应用形成树型结构。它采用“简单要素”数据模型对现实世界建模, 其中包括空间要素和非空间要素。空间要素有两个类型:

(1)简单要素, 包括点(point)、线(linestring)和面(polygon);

(2)要素集合是简单要素的集合, 包括多点(multipoint)、多线(multistring)、多面(multipolygon)以及点线面组成的复合类型(multigeometry)^[1,3]。

GML格式描述的空间数据与其它格式存储的空间数据通过“简单要素”数据模型建立了映射关系, 例如: 在SQL数据库(如PostGIS)中存放有Road类型的地理要素, 其模式Road(R_name:varchar(50), Geom: LineString), 则用GML描述如下:

```
<?xml version="1.0" ?>
<schema
```

基金项目: 国家自然科学基金资助项目(60573095)

作者简介: 方颖(1974-), 女, 讲师、博士生, 主研方向: 高级数据库; 曹加恒, 教授、博导; 曾承, 博士生

收稿日期: 2006-05-27 **E-mail:** whelf2005@hotmail.com

```

...>
<element name="Road"
type="myns:Road_Type"
substitutionGroup="gml:_Feature"/>
<complexType name=" Road_Type ">
<complexContent>
<extension base="gml:AbstractFeatureType">
<sequence>
<element name="R_name" nillable="true" minOccurs="0">
<simpleType>
<restriction base="string">
<maxLength value="50"/>
</restriction>
</simpleType>
</element>
<element name="Geom"
type="gml:LineStringPropertyType" nillable="false"/>
</element>
</sequence>
</complexContent>
</complexType>
</schema>

```

使用上述的 GML 模式描述，可以得到相应的地理要素实例。

2 WFS 实现地理数据要素层互操作

GML 是 WebGIS 互操作的基础，基于 XML 以及“简单要素”数据模型的特点使它能够对空间数据进行编码、存储和传输。但是 GML 只是空间数据的一个载体，它本身无法实现用户对空间数据交互操作，为此 OGC 和 ISO/TC211 推出了基于 GML 的地理信息服务规范，包括 Web 地图服务规范 WMS、Web 要素服务规范 WFS 和 Web 覆盖服务规范 WCS 等。

2.1 WFS 基本功能

目前商业 WebGIS 软件普遍支持 WMS，WMS 提供了一定的 WebGIS 互操作支持。但是由于 WMS 静态图像输出以及无法实现基于地理要素层的互操作，仅仅基于 WMS 实现地理数据 Web 服务无法满足 WebGIS 互操作应用的要求。

WFS 是 WMS 的发展和扩充。WFS 返回的是要素级的 GML 编码，并提供对要素的增加、修改、删除等事务操作。它允许客户端从多个 Web 要素服务中取得使用 GML 编码的地理空间数据。

WFS 支持 5 个操作：(1)GetCapabilities 返回描述 Web 要素服务性能描述文档；(2)DescribeFeatureType 返回描述可以提供服务的任何结构的 XML 文档；(3)GetFeature 请求可以获取要素实例；(4)Transaction 为事务请求提供服务；(5)LockFeature 处理在一个事务期间对一个或多个要素类型实例上锁的请求^[4]。

这 5 个操作可以实现 3 类 WFS：Basic WFS，XLink WFS 以及 Transaction WFS，其中 Transaction WFS 可支持事务操作。

2.2 WFS 的应用

在 WFS 应用中，客户端应用可以向 WFS 请求要素层的 GML 空间数据，WFS 服务器得到请求并分析请求，最后返回 GML 格式空间数据，WFS 的处理过程如图 1 所示。客户端可以同时向多个不同的空间数据库服务器(包括本地和远程)请求一个或多个空间要素。

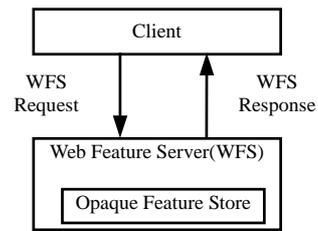


图 1 Web 要素服务体系结构

下列实例中的 GetFeature 请求查询来自不同服务器，包围在矩形盒[-50,40,100,260]内的 Road 类型的属性 R_name 以及 Tree 类型的属性 T_Geom。

```

<?xml version="1.0" ?>
<GetFeature
...
<Query typeName="myns: Road "/>
<wfs:PropertyName>myns: R_name </wfs:PropertyName>
<ogc:Filter>
<ogc:Within>
<ogc:PropertyName> myns: Geom </ogc:PropertyName>
<gml:Envelope srsName="EPSG:63266405">
<gml:lowerCorner>-50 40</gml:lowerCorner>
<gml:upperCorner>100 260</gml:upperCorner>
</gml:Envelope>
</ogc:Within>
</ogc:Filter> </Query>
<Query typeName=yourns: Tree "/>
<wfs:PropertyName>yourns: T_ Geom </wfs:PropertyName>
<ogc:Filter> <ogc:Within>
<ogc:PropertyName>yourns: T_ Geom </ogc:PropertyName>
<gml:Envelope srsName="EPSG:63266405">
<gml:lowerCorner>-50 40</gml:lowerCorner>
<gml:upperCorner>100 260</gml:upperCorner>
</gml:Envelope>
</ogc:Within>
</ogc:Filter>
</Query> </GetFeature>

```

上述实例的 GetFeature 实现了从两个不同空间数据库服务器查询和提取一定空间范围的数据要素实例，而不需要下载整个数据集。

图 2 描述了借助于 WFS 实现 GML 数据在 Web 上传输和共享：(1)在数据层，不同格式的数据首先要转换为 GML 格式；(2)在 WFS 服务器上，客户端用户请求要被生成对 GML 数据的请求描述，同时 WFS 服务器能将请求到的结果返回给客户端用户。

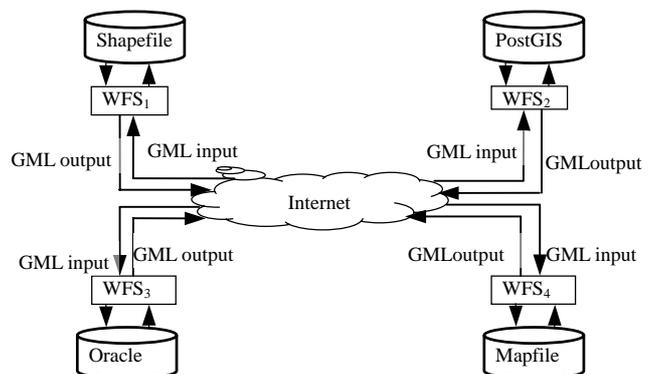


图 2 WFS 实现 GML 数据在 Web 上传输和共享

由于WFS服务于GML数据，GML的XML特征使得一个数据源中数据的更新可以自动地传播到相关联的其它数据源中。例如下例中，city由street和building通过XML连接组成，则当更新street或者building时，city会自动的更新^[5]。

```
<gml:FeatureCollection>
<gml:featureMember xlink:type="simple"
xlink:href="http://../street.xml"/>
<gml:featureMember xlink:type="simple"
xlink:href="http://../building.xml"/>
</gml:FeatureCollection>
```

这样，不同数据源的数据通过 WFS 联系在一起，实现跨平台、异构要素层数据共享。

3 SVG 实现矢量空间数据可视化

GML在Web上实现空间数据的存储和传输，但是GML文档本身不能以地图的形式显示，而客户端用户通过WFS服务器查询的数据是GML数据，为了显示GML数据，需要采用XSLT将GML文档转换为另一种XML文档，如：VML，X3D以及SVG等矢量格式，才能在客户端显示输出^[6]。

SVG 是一个开放的二维图形格式，也是 XML 的一个应用。GML 数据与 SVG 之间可以建立如下的映射关系：GML 中的根元素 featureCollection、子元素 featureMember 对应 SVG 中的标记 <g>，用于组织具有相同结构的地理要素；GML 中的类型为 gml:Point 的要素对应 SVG 中的标记 <circle>；类型为 gml:LineString 的要素对应 SVG 中的标记 <polyline> 或 <path>；类型为 gml:Polygon 的要素可用 SVG 中的 <rect>、<path> 以及 <polygon> 标记来表示；GML 中注记要素对应 SVG 中的文本 <text> 标记。文献[6]详细地介绍了采用 XSLT 转换 GML 数据为 SVG 的原理和方法。图 3 显示了 GML 数据在 Web 客户端以 SVG 格式输出的转换过程。第 1 节中用 GML 模式描述得到的某地理要素实例可以采用 XSLT 转换为如下的 SVG 文档：

```
<?xml version="1.0" encoding="utf-16" ? >
<svg width="200" height="200">
...
<polyline points="50,250 100,225 300,225 "
stroke="blue" fill="none" stroke-width="4">
<text x="150" y="175" style="fill:red">武珞路</text>
<GeoAttribute>
< R_name > 武珞路 </ R_name >
</GeoAttribute>
...
</svg>
```

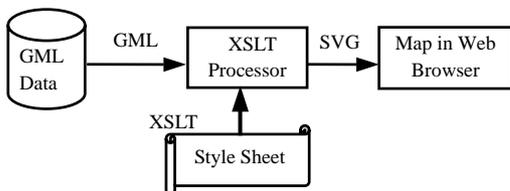


图 3 GML 数据生成地图过程

最终，该 SVG 文档在带有 SVG 插件的客户端浏览器上显示为一条折线。

4 互操作 WebGIS 实现框架

以GML为基础，采用WFS实现地理数据要素层的查询和提取，在客户端以SVG格式输出高质量地图，从而实现要素层异构、跨平台互操作WebGIS。其实现框架如图 4 所示，采

用B/S 4 层结构，依次为客户层、Web服务层、WFS服务层以及空间数据层^[3]。

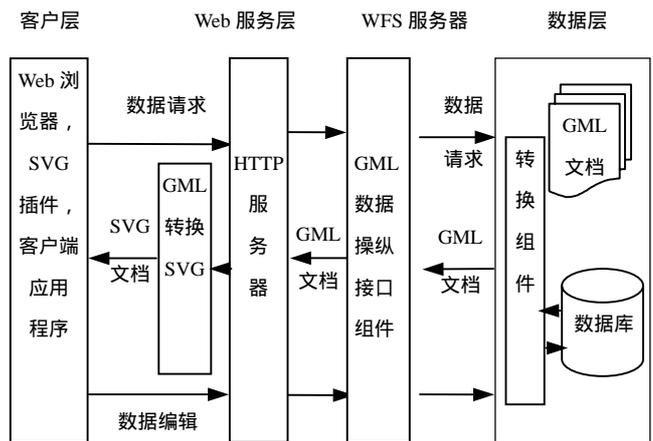


图 4 互操作 WebGIS 实现框架

客户层是用户的使用层，主要负责地图的浏览、显示以及请求等用户接口。

Web 服务层执行 Web 服务，它的 HTTP 服务器实现客户端与 WFS 服务器之间的数据传输。返回的 GML 格式数据需要使用 XSLT 转换为 SVG 格式。

WFS 服务器提供 WFS 服务，支持事务和查询请求，一方面它把客户端的请求转换为目标数据能接收的语言从而把客户请求传输给数据层执行；另一方面将查询结果返回给客户端。

数据层用于提供空间数据，包括 Oracle Spatial、PostGIS 等空间数据库，Shapefile 文件以及 GML 文档。从数据层得到的必须是 GML 格式的数据，非 GML 数据与 GML 数据需要使用转换组件进行转换。

这 4 层结构实现了实时动态交互操作 GML 数据。客户端可以在 Web 上实时地存取和检索要素层地理数据：(1)客户端向服务器发出一系列请求，服务器接收请求并分析这些请求，得到相应数据库中的数据；(2)服务器将 GML 格式的请求结果返回给客户端或者将转换后的 SVG 返回给客户端；(3)客户端显示 SVG 数据并且执行 GIS 分析。

结合 GML、SVG 以及 WFS 的 4 层结构是基于客户端的应用，用户与 GML 或 SVG 数据的交互可以直接在客户端进行，与基于服务器端的实现方法相比减少了网络传输的负担以及服务器的负担，同时又能得到高质量的 SVG 图像；与传统的客户端方法相比，它不需要复杂特有的客户端应用程序，克服了传统客户端方法的不足。

5 结论

以 GML 为基础实现异构跨平台下的互操作 WebGIS 是目前 WebGIS 研究的一大热点。本文提出的 WebGIS 互操作实现新机制以 GML 为基础，结合 WFS 以及 SVG 的功能，通过 4 层结构实现框架实现要素层地理数据跨平台、异构互操作，不仅实现地理数据分布式共享，而且能给用户高质量矢量图形。

根据互操作 WebGIS 实现框架，选用 MapServer 提供 WFS 服务，同时还选用 MapServer 作为 WFS 客户端。安装 Apache HTTP Web 服务器以及 PHP 软件，配置 Mapfile 文档，编写 PHP/Mapscript 脚本语言来实现互操作 WebGIS 功能。具体的实现结果在我们下一步研究中将会给出。

(下转第 52 页)