

基于 GML 的空间数据转换服务研究^①

Implementation of GML Based Spatial Data Transformation and its Web Service

兰小机 (南京师范大学地理信息科学江苏省重点实验室 210097)
(赣州市江西理工大学环境与建筑工程学院 341000)

闫国年 张书亮 施苗苗 殷丽丽 威海峰 (南京师范大学地理信息科学江苏省重点实验室 210097)

摘要:系统分析了 GML 3.0 实现规范,并基于 GML 3.0 的核心模式创建了用于空间数据转换的 GML 应用模式;利用 JavaBeans 技术实现了 ARC/INFO E00、中国地球空间数据交换格式 VCT 与 GML 数据的双向转换,及 GML 数据到 SVG 格式的转换;利用 Web Services 技术,在 JBuilder X 环境下实现了基于 GML 的空间数据转换服务。

关键词:GIS GML 3.0 空间数据转换 Web 服务

地理信息系统经过近四十年的发展,几乎在所有与空间信息相关的领域得到了广泛应用,同时也积累了大量的空间数据。但是,长期以来由于对地理世界的描述缺乏统一的标准,不同的 GIS 软件厂商对同一地理实体或地理现象的描述不一致,而空间数据又宿主于不同的 GIS 软件平台,导致不同格式的空间数据无法无损的共享。尽管各 GIS 软件厂商或其他组织提供了公开的交换文件格式或数据转换标准来进行空间数据的转换,但由于缺少相应的支持工具,而没有被广泛接受。开放地理信息系统协会(OpenGIS Consortium, OGC)提出的地理标记语言(Geography Markup Language, GML)实现规范,为 GIS 空间数据的建模、传输和存储提供了统一的框架。GML 使用 XML 对地理数据进行编码,为开发商和用户提供了一种开放的、中立于任何厂商的地理数据建模框架,使地理信息能够在不同领域、不同部门进行语义共享。由于 GML 建模具有易扩展性、可选择性和 XML 工具的广泛性等特点,基于 GML 的空间数据转换克服了以往任何数据转换存在的缺陷。本文主要研究基于 GML 的空间数据转换及其 Web 服务的有关问题。

1 GML 概述

1GML 是一种用于建模、传输和存储地理及与地理相关信息的 XML 编码语言,它包括了地理要素的空间与非空间特征^[1]。GML 建立在 W3C 系列标准之上,以一种互联网上容易共享的方式来描述、表达地理信息,是第一个被 GIS 界广泛接受的元标记语言。在这之前开发的各种用于存储和交换地理信息的文件格式或标准,由于缺少相应的支持工

具,而没有被广泛接受。GML 一个重要优势就是它能够让开发者或用户非常灵活地使用已被广泛接受的 XML 技术,它建立在 XML、XML Schema、XLink 和 XPointer 基础之上, GML 数据能够很容易地与非空间数据集成使用。GML 主要采用要素(Feature)来描述地理实体和地理现象, GML 要素是通过其属性(property)来描述的。

OGC 先后推出了 1.0、2.x、3.0 三个版本的 GML, GML 3.0 增加了对复杂的几何实体、拓扑、空间参照系统、元数据、时间特征和动态要素等的支持,使其更加适合描述现实世界问题。GML 提供了一套核心模式和一个基于要素/属性(Features/properties)的简单语义模型。GML 核心模式定义了构建地理要素的基本组件,使用 GML 模型及其模式组件,用户可以在自己的应用模式中定义其问题领域中的地理要素。

GML 最常用的核心模式有 3 个:要素模式(Feature Schema)、几何模式(Geometry Schema)和拓扑模式(Topology Schema)。

1.1 要素模式

地理要素包含一系列的空间与非空间属性。要素模式 feature.xsd 为创建 GML 要素和要素集合提供了一个框架,它定义了抽象和具体的要素元素及类型,和以前版本相比,增加了一些新的要素类型及属性,如 BoundedFeatureType、FeatureArrayPropertyType、EnvelopeWithTimePeriodType 等,并通过 <include> 元素引入了几何模式 geometryBasic2d.xsd 和时态模式 temporal.xsd 中的定义和声明。

1.2 几何模式

① 基金项目:863 资助项目(编号:2002AA131030)

几何模式定义了地理要素的几何类型和属性。GML 2 的几何模式支持的几何基元仅有: Point、LineString、LinearRing、Box、Polygon 及相应的聚合类型: MultiPoint、MultiLineString、MultiPolygon。GML 3.0 支持包括 points、curves、surfaces 及 solids 在内的三维几何模型,在其几何模式中增加许多新的类型,包括: Arc、Circle、CubicSpline、Ring、OrientableCurve、OrientableSurface 及 Solid,还有聚合类型如 MultiPoint、MultiCurve、MultiSurface、MultiSolid 和复合类型如 CompositeCurve、CompositeSurface、CompositeSolid 等。GML 3.0 将其几何模式分为 5 个文件模块: geometryBasic0d1d. xsd、geometryBasic2d. xsd、geometryAggregates. xsd、geometryPrimitives. xsd、geometryComplex. xsd,前 3 个模式中包含了最常用的、与 GML 2 兼容的线性几何组件,后 2 个模式中包含了最新的非线性几何组件。

1.3 拓扑模式

拓扑模式定义了描述几何元素之间拓扑关系的类型和属性,它使用拓扑基元 Node、Edge、Face、TopoSolid 以及这些基元之间的关系描述来构建拓扑关系,拓扑基元通常用来表达几何基元 Point、Curve、Surface、Solid。拓扑基元之间的连接关系主要有:边的公共结点、面的公共边以及三维实体的公共面等。GML 3.0 在拓扑模式 topology. xsd 中对相关的拓扑类型和属性进行了定义,并通过<include>元素引入了复合几何模式 geometryComplexes. xsd 中的定义和声明。

2 GML 应用模式设计

GML 应用模式设计是基于 GML 空间数据转换的关键问题之一,设计时应综合考虑各种 GIS 数据格式所包含的几何对象和拓扑关系,一般 GIS 都把空间数据分为点、线、面三类几何元素,有的考虑了拓扑关系,有的则没有考虑。在我们的研究过程中,考虑到 GML 应用模式的通用性,设计了同时包括几何、拓扑属性的三类要素类型 ZeroDimensionFeatureType、OneDimensionFeatureType、TwoDimensionFeatureType,分别用来表示点、线、面类要素,模式文档具体代码略。

3 空间数据转换软件设计

在我们的研究实例中,利用上节设计的 GML 应用模式和 JavaBeans 技术,在 JBuilder X 环境下,设计了 5 个 Java 组件,来实现 VCT、E00 与 GML 之间的相互转换以及 GML 到 SVG 的转换,这 5 个 Java 组件分别是: CGML2e00. java,实现 GML 到 e00 的转换; e002CGML. java,实现 e00 到 GML 的转换; CGML2VCT. java,实现 GML 到 VCT 的转换; VCT2CGML. java,实现 VCT 到 GML 的转换; CGML2SVG.

java,实现 GML 到 SVG 的转换。

在具体程序设计时,我们对不同的数据格式设计了相应的存储结构,如对 e00 格式,我们设计了以下存储结构:

```
//E00 头文件
class E00Head
{
    int ComPressType; //压缩类型
    String FileName; //输出文件名
};
//E00 弧段结构
class E00Arc //E00 弧段结构
{
    int system_ID; //系统 ID
    int esri_ID; //ESRI ID
    int start_ID; //起点 ID
    int end_ID; //终点 ID
    int left_PolyID; //左多边形 ID
    int right_PolyID; //右多边形 ID
};
//E00 点结构
class E00Lab
{
    int label_ID; //LABEL ID
    int aim_ID; //目标 ID
    Point point1;
    Point point2;
};
//E00 多边形结构
class E00PalInfo //弧段信息
{
    int id; //弧段 ID
    int start_ID; //弧段起点 ID
    int polygon_ID; //另一多边形 ID
};
class E00Pal //PARCEL 体
{
    int count; //弧段数
    Point point1; //闭包最小值
    Point point2; //闭包最大值
    E00PalInfo[] palInfoArray; //PARCEL 弧段序列
};
//E00Cnt 结构
class E00Cnt
```

```

{
    int label_num;
    Point position; //点位置
};
//E00Prj 结构
class E00Prj
{
    String projection; //投影值
    String zone;
    String datum;
    String zunits; //Z 向单位
    String units; //数据单位
    String spheroid; //椭球体
    double Xshift; //X 向偏移量
    double Yshift; //Y 向偏移量
    String parameters; //投影参数
};
//E00INFO 段数据结构
class E00InfoHead
{
    int fieldNum1; //字段数
    int fieldNum2; //字段数
    int recordWidth; //记录总宽度
    int recordNum; //记录个数
};
class E00InfoStruct
{
    String fieldName; //字段名称
    int fieldWidth; //字段宽度
    int fieldPos; //字段起始位置
    String format; //输出格式
    String fieldType; //字段类型
    int fieldID; //字段 ID
};
class E00Info
{
    String name; //文件名 & 层名
    E00InfoHead head; //INFO 头
    E00InfoStruct[] structure; //INFO 字段
    String[][] dataArray; //存放 INFO 数据
};
class E00Anno
{

```

```

    Point Annopoint;
    String height;
    String width;
    String Annotext;
};

```

4 空间数据转换服务的实现

为实现空间数据的转换服务,我们设计了两个类 CGMLFrame.java 和 CGMLServices.java, CGMLFrame.java 类将上节实现的空间数据转换功能都放在同一个可视化界面中,其主界面如图 1 所示。在 CGMLServices.java 类中,对 CGMLFrame.java 类进行实例化,并设计了一个方法 DataTransformation 来调用空间数据转换功能。

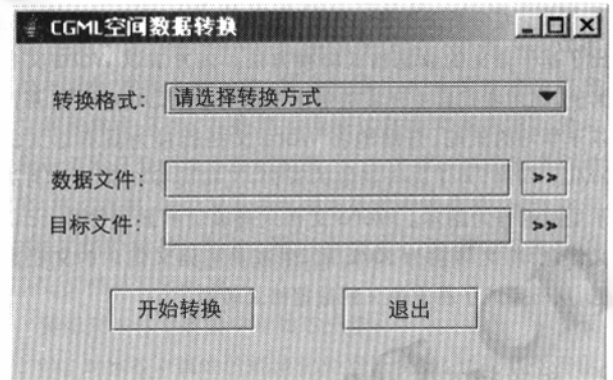


图 1 基于 GML 的空间数据转换服务主界面

4.1 空间数据转换服务的实现

Jbuilder X 集成开发环境中内置了 Web 服务工具包 Apache AXIS 1.1 和 Servlet 引擎 Tomcat 4.1 来支持 Web 服务的实现。在 Jbuilder X 环境下,利用现有的 JavaBeans 创建 Web 服务包括四个主要步骤^[2]:

- (1) 使用 Web Services Configuration wizard 为 Web 服务配置用户工程;
- (2) 使用 Web Services Designer 创建服务;
- (3) 编译并启动 Web 服务;
- (4) 测试 Web 服务。

在 JBuilder X 环境下,我们实现了空间数据转换服务 CGMLServices。

4.2 Web 服务的发布

Web 服务创建后,为了能让 Internet 上任何用户调用,需要到公用 UDDI 注册中心注册。Jbuilder X 提供了 Web Services Explorer,方便用户查询、发布 Web 服务。目前 Internet 上有一些公用的 UDDI 注册中心,如:Microsoft UDDI

(下转第 23 页)

(上接第 39 页)

注册中心, <http://uddi.microsoft.com/>, Microsoft UDDI 测试注册中心, <http://test.uddi.microsoft.com/>. 笔者将本文研究开发的基于 GML 的空间数据转换服务发布在 Microsoft 的 UDDI 测试注册中心(商业名为 landcom), 用户可以直接调用或在自己的应用程序中调用基于 GML 的空间数据转换服务, 完成相应的功能, 实现空间数据的共享。

5 结束语

基于 GML 的空间数据转换服务为空间数据共享提供了一种行之有效的途径。目前, GML 已经成为事实上的空间

数据编码、传输、存储、发布的国际标准, 其应用开发已得到了许多系统和软件厂商以及政府机构、学术机构的支持。基于 GML 的 GIS 将成为下一代 GIS 的主流。

参考文献

- 1 OpenGIS Consortium, Inc., 2003, OpenGIS Geography Markup Language (GML) Implementation Specification 3.0.
- 2 Borland Software Corporation, 2003, Web Services Developer's Guide.