

应用 REST 技术的 GML 数据管理

张山山^{1,2}, 吴宝佑^{1,2}, 隋宁宁^{1,2}, 秦 瑞^{1,2}

- (1. 中南大学 有色金属成矿预测教育部重点实验室, 长沙 410083;
2. 中南大学 地球科学与信息物理学院, 长沙 410083)

摘 要: 地理标记语言(GML)作为使用 XML 信息编码来表达地理数据的标准, 其主要目的是实现数据共享与互操作。随着 GML 广泛应用, GML 数据存储管理成为研究热点。面向服务架构(SOA)已经广泛应用于地理信息服务, 表述性状态转移(REST)作为一种专门为分布式调用而设计的轻量级架构技术, 在构建 SOA 方面有着明显的优势。论文采用对象数据库对 GML 进行存储管理, 探讨如何利用 REST 架构技术实现 GML 空间数据的管理与共享。分析了 REST 架构的关键技术, 设计了 GML 数据管理服务框架。详细论述了基于微软 WCF 框架的 REST API 的实现过程, 最后实现了一个调用 GML 数据管理服务的数据管理系统。通过客户端实例验证表明 REST 适合于 GML 地理数据的管理与共享。

关键词: 面向服务架构; 表述性状态转移; 地理标记语言; 数据管理

中图分类号: P208

文献标志码: A

GML data management using REST

ZHANG Shan-shan^{1,2}, WU Bao-you^{1,2}, SUI Ning-ning^{1,2}, QIN Rui^{1,2}

- (1. Key Laboratory of Metallogenic Prediction of Nonferrous Metals, Ministry of Education,
Central South University, Changsha 410083, China;
2. School of Geosciences and Info-Physics, Central South University, Changsha 410083, China)

Abstract: It is necessary that geography markup language (GML) which uses XML as the encoding information to express the geographic data provides data sharing and interoperability. As the application of GML is more extensive, GML data storage and management is one of the hot spots. With the development of the network technology and GIS technology, service oriented architecture (SOA) was widely used in geographic information services. Representational state transfer (REST) is a lightweight architecture technology for distributed call, which has a distinct advantage in building SOA. The object database is used in GML data storage, and how to use REST architecture technology to realize the management and sharing of GML spatial data is mainly discussed. First, the key REST architecture technology and the design guidelines of GML data management service were introduced. Then the process of realizing the API based on the WCF framework was introduced. Finally, a data management system that is called GML data management service is achieved. This system shows that REST is ideal for GML geographic data management.

Key words: service oriented architecture (SOA); representational state transfer (REST); geography markup language (GML); data management

地理标记语言(Geography markup language, GML)
是开放式地理空间信息联盟(Open geospatial

consortium, OGC)推荐的用以描述地理数据的标准格式, 在实际中有着广泛的应用。随着越来越多的空间

数据使用 GML 来表达,以及各种地理信息服务的不断涌现,如何有效地管理这些数据,并提供这些数据的服务,是当前在 GML 领域亟待解决的问题之一^[1]。在 GML 的数据存储方面,目前并没有一个真正意义上的空间数据库或者空间数据库引擎能够提供对 GML 空间数据进行有效的存储与管理。由于 GML 数据存在着不同层次上的嵌套关系,其特殊的树形结构同对象数据库的嵌套对象模型非常吻合,所以将 GML 空间数据对象化存储是管理 GML 文档不错的选择。在日益复杂的分布式网络环境下,面向服务架构(Service oriented architecture, SOA)^[2]作为一个新型的企业级分布式软件架构思想,完全可以为地理信息服务的构建提供帮助,并且给用户提供了简单易懂的地理信息服务接口。目前实现 SOA 架构的方法有 CORBA、EJB、Web Service 和 REST 等,其中 REST 技术是为通过 HTTP 协议来进行分布式调用量身定制的架构。随着 GIS 技术和 REST 架构方法的不断成熟,国内外学者结合二者作了不少研究。DZENANA 等^[3]在分析了空间数据及其服务的基础上,利用 REST 技术实现了可以利用标准的 HTTP 协议访问空间数据的简单服务。MAZZETTI 等^[4]从 REST 架构的角度分析了现有的地理信息服务,并探讨了可能的 REST 的实现。毛峰等^[5]结合 REST 与 OGC 规范,设计了一种面向资源的地理信息服务,从而实现信息共享与功能的互操作。张利利等^[6]提出一种基于 REST 架构模式的遥感影像服务方案,并结合 WCF 框架实现了该遥感影像服务。如上所述,国内外的研究现状主要集中在对 REST 技术应用于地理信息服务的探讨和研究,且应用的 GIS 数据格式为不同的 GIS 数据生产厂商提供。实际上,REST 特别适合于处理 GML 这种 XML 编码格式的数据,本文作者在此基础上提出一种基于 REST 架构模式的 GML 土地利用数据管理服务方案,并详细介绍 REST API 的设计与实现过程。

1 GML 数据存储管理

本研究采用开源的纯面向对象数据库引擎 db4o (database for objects)^[7]来实现 GML 空间数据存储,该数据库引擎为存储任何复杂的对象提供了持久性的解决方案。

1.1 存储粒度

在基于 db4o 的 GML 空间数据库中,存储粒度表示记录与结点的对应关系,记录的粒度主要有 3 种,

分别为:节点级、子树级和文档级^[8]。顾名思义,节点级表示每一个结点对应一个记录(对象),子树级表示一个子树对应一个记录(对象),文档级则表示将整个文档作为一个记录(对象)。

由于 GML 文档由一个或者多个要素集组成,要素集由一个或多个要素组成,要素由其属性(包括非空间属性、空间属性和拓扑属性等)组成。要素是描述一个具体的地物对象的最小单位,任何 GML 空间数据都是由要素组成的一个集合对象。因此本研究选择的存储粒度为子树级存储粒度,该子树大小为一个要素。

1.2 映射方式

德克萨斯大学的 LEONIDAS^[9]提出了两种基于对象数据库的 XML 映射技术。第一种是针对无模式的 XML 文档,采用固定的对象数据管理组(Object data management group, ODMG)对象定义语言(Object define language, ODL)模式实例化 XML 对象。第二种是针对有模式的 XML 文档,该方法对每一种 XML 元素类型映射为一个对象类型,对象之间的关系使用 ODMG 提供的类之间的联系来实现。

由于 GML 文件提供了模式,适合采用第二种对象映射方法,但根据 GML 的特点作者在此映射方法上作了改进。具体映射过程是将作为存储粒度的要素映射为对象,要素的属性结点则映射为对象的属性,而不是也映射为对象,这样做的目的是防止大量对象的产生,从而影响系统的性能。

GML 模式具体的映射规则如下:

- 1) 将类型为 complexType 作为存储粒度的要素(GML 中的元素)映射为对象;
- 2) 将类型为 simpleType 的要素子节点元素映射为对象的属性;
- 3) 内置的元素简单数据类型(如 int, string 等)映射为数据库中相应的简单数据类型;
- 4) 几何类型(如 pointPropertyType, curver-PropertyType, surfacePropertyType 等)统一映射为 IList<PointF>类型,并设置要素对象的 featureType 属性为具体的几何类型(点,线,面等)。

根据 GML 空间数据的特点,由于要素的属性(空间、非空间属性)不包含子节点元素,所以要素的子节点元素被转换为对象的属性。

1.3 存储顺序

由于本研究采用子树级存储粒度,故这里还涉及到记录的组织顺序问题。记录的存储顺序包括按深度优先存储、按广度优先存储和按同类记录聚集存储 3

种方案。

为了使得存储 GML 顺序同 GML 文档的原始顺序保持一致,同时也是为了实现起来更加方便,本研究采用深度优先的存储方案。

2 REST 架构关键技术

表述性状态转移(Representational state transfer, REST)并不是一种标准,而是一种面向资源的软件架构风格,它使用包括 HTTP、URI、XML、MIME 等一系列标准来实现^[10]。REST 将网络看成是资源的集合,网络中的每一个资源都是通过一个唯一的 URI 来标识,如一个 GML 图层文档(行政区图层)可以标识为 `http://gisland/REST/GMLServices.svc/gmlDoc/xzq`,这种将网络看成资源集合的观点非常贴切于我们对 GML 空间数据的认识,用于标识资源的 URI 也更容易理解和记忆。REST 中所有的资源都是通过表述和 HTTP 标准方法(包括 GET、POST、PUT 和 DELETE 等)对其进行操作,如对一个 GML 图层文档(行政区图层)进行 XQuery 查询,可以构造一个 XQuery 语句,并将该 XQuery 语句以 POST 方式发送给 `http://gisland/REST/GMLServices.svc/gmlDoc/xzq/xquery`,最后得到返回的结果,这种操作网络资源的方式可以使得组件间的耦合度减小,并且在客户端的实现难度大大降低。

在 REST 中,分布式是通过对资源的操作来实现的,并且资源被映射到相应的一套 URI 规则上,也就是说资源只和 URI 相关,与具体实现并无关联,因此 REST 具有很好的解耦性。GML 作为一种 OGC 推荐的空间数据的标准数据,同时其基于的数据格式 XML 也是网络数据交换的标准格式,将 GML 运用 REST 技术构建空间数据网络服务是一个不错的选择。

3 GML 数据管理服务 REST API 设计

3.1 设计总线

由于 REST 是一种对抽象为资源的对象进行标识、操作的网络架构风格,结合 GML 数据文档包含有 XML Schema 模式文档和 GML 实例文档的特点^[11],可以将该 REST 设计为如图 1 所示的框架。

对于 GML 的 XML Schema 模式文档,可以针对不同的模式文档设定不同的资源服务标识,以便用户方便快速地查看 GML 文档对应的模式文件。同时由于 GML 模式文档是对 GML 实例文档的格式定义,其包含了 GML 实例文档中的要素包含的一些属性字段,故在模式文档 REST 服务中还提供了查询该模式文档对应的 GML 实例文档要素中所含有的属性字段服务。

对于 GML 实例文档,REST API 提供了资源标识

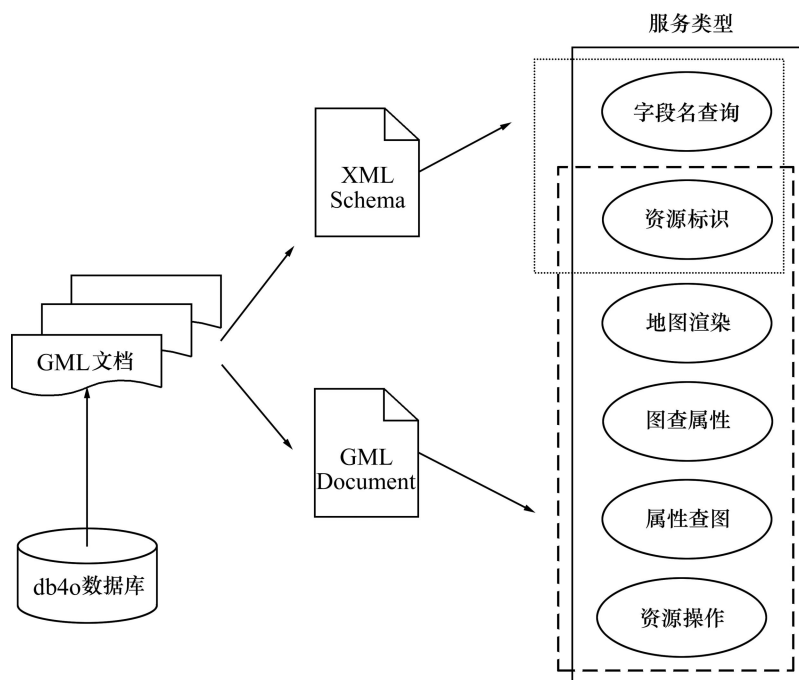


图 1 GML 数据管理服务框架

Fig. 1 Framework of GML data management service

服务、地图渲染服务、简单的地图查询(包括图查属性和属性查图)服务和资源操作服务等。其中资源操作服务主要是对 GML 实例文档进行 XQuery 查询、更新等,这也是 REST 服务对 GML 数据操纵的关键。

3.2 XML Schema 模式服务

GML 模式服务采用资源标识的方式实现,每个模式都对应一个全局唯一的 URI 地址。由于 URI 是客户端连接 REST 服务的接口,故 URI 的命名规则要做到顾名思义,并且具有一定的符合逻辑的层次感。

为了使得设计的 REST 资源标识服务遵循一致的规则,本文为模式服务设计的 URI 模板见表 1。

其中资源标识服务的 URI 模板中第一个网络层次(xmlSchema)代表的是该目录为 XML Schema 模式服务,第二个网络层次({tdlyType})代表的是具体的土地利用类型(行政区、宗地、地类图斑等)。由此可以根据不同的土地利用类型来查询其对应的 GML 模式文档,如 xmlSchema/xzq 表示的是行政区的模式文档 URI,xmlSchema/zd 表示的是宗地的模式文档 URI,GML 要素属性字段名查询服务可以依此类推。

REST 服务向客户端输出的资源内容的格式可以有很多种,可以是标准的 XML 和 JSON 等格式,也可以是自定义格式。由于 XML Schema 资源标识服务向客户端输出的资源内容需要为原始的 XML Schema 文档,故其输出的资源内容格式设置为 XML 格式;GML 要素属性字段名查询服务输出的资源内容格式为 JSON 格式。

3.3 GML 实例文档服务

GML 实例文档包含的服务较多,具体的资源标

识、地图渲染、地图查询和资源操作等 REST 服务对应的 URI 模板如表 2。

所有服务类型的服务 URI 模板都是以/gmlDoc 为根目录,表示这些服务都是 GML 实例文档提供的 REST 服务,{tdlyType}的含义同 XML Schema 资源标识服务,表示的是土地利用类型。

GML 资源标识服务是为不同的土地利用类型返回对应 GML 实例文档的 REST 服务,该资源标识服务向客户端返回的数据类型为 XML 格式。

GML 地图渲染服务是对某一土地利用类型的 GML 实例文档进行地图渲染的服务。本研究实现一个简易的网络地图服务(Web Map Service, WMS),采用的地图渲染方式是根据用户输入的地图图片宽度(w)、高度(h)以及地图范围(minx, miny, maxx, maxy)在服务器端解析 GML 数据动态生成地图。由于该服务向客户端返回的数据为对应动态渲染地图的临时 URL 地址,故其返回类型为字符串格式。

图查属性服务是根据在地图上查询选择到的图形要素(集)的属性,其中{queryType}是指查询的类型(包括点选、线选、矩形选择和多边形选择),{coors}表示的是 GML 地图坐标(对),该服务向客户端返回的数据类型为 JSON 格式。

属性查图服务是 REST API 中提供的一种比较简单的通过单一字段值查询地图的服务,其中{field}、{operator}和{value}分别代表的是待查询的字段名、比较运算符和字段值,该服务向客户端返回的数据类型是表示所有查询到的要素(集)坐标对及其所有非空间属性的 JSON 格式数据。

GML 资源操作服务是 REST API 中提供的较为全面的对 GML 实例文档进行各种操作的服务,主要包

表 1 XML Schema 模式服务对应的 URI 模板

Table 1 URI templet of XML Schema service

服务类型	服务 URI 模板	返回格式
XML Shema 资源标识	/xmlSchema/{tdlyType}	XML
GML 要素属性字段名查询	/xmlSchema/{tdlyType}/fields	JSON

表 2 GML 实例文档服务对应的 URI 模板

Table 2 URI templet of GML instance document service

服务类型	服务 URI 模板	返回格式
GML 资源标识	/gmlDoc/{tdlyType}	XML
GML 地图渲染	/gmlDoc/{tdlyType}/map/w/h/minx/miny/maxx/maxy	String
图查属性	/gmlDoc/{tdlyType}/queryM2P/{queryType}/{coors}	JSON
属性查图	/gmlDoc/{tdlyType}/queryP2M/{field}/{operator}/{value}	JSON
GML 资源操作	/gmlDoc/{tdlyType}/XQuery/{xqueryExp}	XML

括 XQuery 查询和更新操作, 其中 {xqueryExp} 表示的是 XQuery 查询、更新语句, 该服务向客户端返回的数据类型统一设置为 XML 格式。

4 REST API 实现与应用

4.1 REST 服务实现

随着 REST 设计应用越来越广泛, REST 开发框架也是层出不穷, 目前比较主流的开发框架有: 微软在 .NET Framework 3.5 中引入的 WCF, 采用 Java 语言的 Axis2、CXF、Restlet、Gomba, 采用 Python 语言的 Rails、Django, 采用 PHP 语言的 cakephp 等。本研究设计的 REST API 为在 WCF 框架上采用 C# 语言进行的开发。

4.1.1 资源标识服务

资源标识服务包括有 XML Schema 资源标识服务和 GML 资源标识服务, 按照上一节对这两个服务的设计原则, 并且设置操作响应的 HTTP 协议统一为 GET 方式, 分别对这两个服务进行实现。

以 XML Schema 资源标识服务为例, 该服务接口设置如下:

```
[OperationContract]
```

```
[WebGet(UriTemplate="/xmlSchema/{tdlyType}")]
```

```
System.Xml.XmlElement GetSchema(String tdlyType);
```

4.1.2 GML 要素属性字段名查询服务

要在服务器端查询出 GML 实例文档中的要素含有的属性字段名, 需要对该实例文档对应的 XML Schema 文档进行解析。对比不同的 XML 解析方法, 本研究采用开源、非提取的 XML 文档解析 API——VTD-XML^[12] 2.10 C# 版本对 XML Schema 文档进行解析, 得出 GML 要素属性名。

按照该服务的设计原则, 设置服务器响应协议为 GET, 并调用 VTD-XML API 对该服务进行实现。

4.1.3 XQuery 服务

XQuery 服务主要是对 GML 实例文档进行一些查询和更新操作, 以查询为例, 对普通的 XML 文档进行 XQuery 查询实现, 可以使用微软针对 XQuery 1.0 规范发布的一个 Microsoft.Xml.XQuery.dll 动态链接库^[13]或者 Altova 公司提供的免费组件 AltovaXML^[14]。考虑到 GML 具有空间数据的特殊性, 对其进行 XQuery 查询不仅包括属性查询还包括空间查询, 上面两个 API 对实现 XQuery 空间查询就显得无能为力了。本研究借助第三方开源软件包 Saxon 9.2^[15], 利用该软

件包提供的扩展函数库, 在 XQuery 查询语句中调用 .NET 方法作为外部函数对 GML 文档进行 XQuery 空间查询。下面为 Saxon 调用的 XQuery 空间查询语句的一个示例。

```
declare namespace ext = "clitype:Spatail.Spatial
Relations?from=" + dllPath + "Spatial.dll";
```

```
for $b in doc("xzq.gml")/yf:TDLY/yf:XZQ
```

```
where ext:Contains($b//gml:posList,rectangle)
```

```
order by $b/yf:BSM
```

```
return $b;
```

按照该服务的设计原则, 设置服务器响应协议为 POST, 并调用 Saxon API 对该服务进行实现。

4.1.4 GML 地图渲染服务

由于 WMS 采用在服务器端动态生成 GML 地图的技术, 当用户输入参数向服务器请求图片时, 服务器端在此过程中执行的操作包括: 从 db4o 数据库中查询解析 GML 数据、渲染 GML 数据、将动态生成的临时地图图片 URL 地址发送给客户端。这个过程如图 2 所示。

在 WEB 服务器端生成临时地图时, 首先使用 VTD-XML 解析 GML 实例文档, 然后利用 .NET 提供的 GDI+ 绘图接口对解析出来的地理信息绘制成地图。

按照该服务的设计原则, 设置服务器响应协议为 GET, 并调用 VTD-XML API 对该服务进行实现。

4.1.5 图文互查服务

图文互查服务包括图查属性服务和属性查图服务。

1) 图查属性服务

该服务是通过与地图交互得到地图信息的服务, 在服务器端, 通过 GML 地图坐标(对)构造标准的 XQuery 空间查询语句, 然后通过此 XQuery 语句对 GML 文档进行查询, 最后将属性值返回。在浏览器客户端, 用户与地图交互的屏幕坐标(对)需要转换为 GML 地图坐标(对)方能调用此服务。由于该服务最后返回的属性值是以 JSON 的形式表达的, 这样使得用户可以很方便地在浏览器客户端通过 JavaScript 脚本语言将其以某种方式显示出来。

按照该服务的设计原则, 设置服务器响应协议为 POST, 并调用 Saxon API 对该服务进行实现。

2) 属性查图服务

该服务是对单个字段进行属性查询的服务, 本研究根据用户输入的 {field}、{operator} 和 {value} 3 个参数构造一标准的 XQuery 非空间查询语句, 然后通过此 XQuery 语句对 GML 文档进行查询, 最后将查询到的要素(集)坐标对和所有非空间属性以 JSON 形式返回。

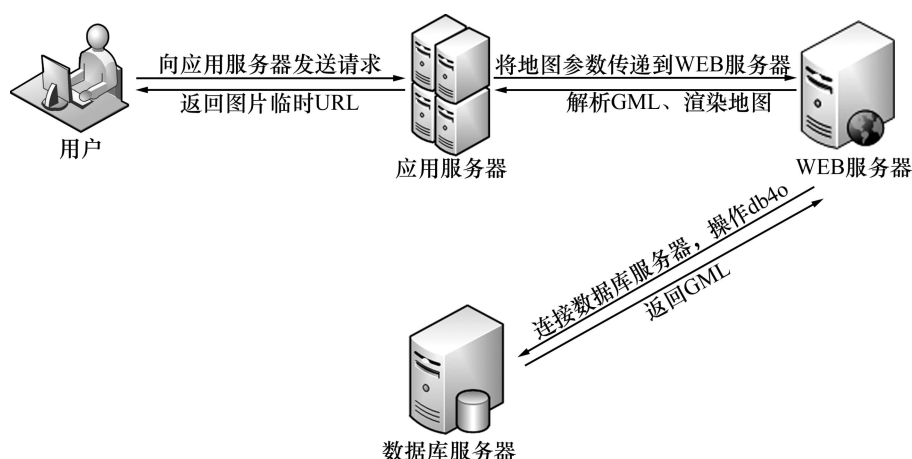


图 2 客户端调用地图渲染服务的过程

Fig. 2 Process of client call WMS

在得到坐标(对)以后, 在客户端可以通过客户端代码(如 JavaScript)以传统的矢量式地图(VML 和 SVG 等)的方式将其绘制到浏览器上。按照该服务的设计原则, 设置服务器响应协议为 GET, 并调用 Saxon API 对该服务进行实现。

4.2 GML 数据管理系统

本研究在根据已经发布的 REST 服务, 利用 JavaScript/Ajax 技术在客户端调用 GML 地理数据服务, 实现了一个简易的 GML 数据管理系统。该系统提供了 GML 地理数据查看、GML 地图数据浏览、图文互查、XQuery 查询与更新等基本功能。图 3 所示为进行图查属性以及执行了 XQuery 空间查询后的系统界面图。

通过此客户端 GML 地图管理系统的实例验证, 本研究设计实现的 REST 服务能够很方便地对 GML 文档数据进行查看与管理, 系统的具有较快的响应速度。



图 3 GML 数据管理系统界面

Fig. 3 Interface of GML data management system

5 结语

1) 首先分析 REST 架构的关键技术, 然后根据 GML 文档的特点结合 REST 风格设计了 GML 数据管理服务 REST API, 接着实现了 GML 数据文档的 Web 服务, 并详细介绍了 REST API 的实现过程中的关键处理技术, 最后搭建了一简易的基于 REST 的 GML 数据管理系统。

2) 通过对该系统可靠性和响应速度的验证, 表明 REST 这种面向资源的架构风格非常适合于对基于 XML 编码的 GML 地理数据的管理。

REFERENCES

- [1] 陈建华. 原生模式 GML 空间数据管理机制研究[D]. 成都: 成都理工大学, 2008: 1-5.
CHEN Jian-hua. Research on native mode GML spatial data management mechanism[D]. Chengdu: Chengdu University of Technology, 2008: 1-5.
- [2] 谭永明, 苏斌. 面向服务架构体系的研究[J]. 计算机技术与发展, 2007, 17(3): 132-134.
TAN Yong-ming, SU Bin. Research of service-oriented architecture [J]. Computer Technology and Development, 2007, 17(3): 132-134.
- [3] DZENANA M, HARIS K. Geospatial SOA using RESTful web services [C]// LUZAR-STIER V, JAREC I, BEKIC Z. Proceedings of the ITI 2009 31st Int Conf on Information Technology Interfaces. Zagreb, Croatia: University of Zagreb, 2009: 199-204.
- [4] MAZZETTI P, NATIVI S, CARON J. RESTful implementation

- of geospatial services for earth and space science applications [J]. *International Journal of Digital Earth*, 2009, 2(1): 40–61.
- [5] 毛 峰, 刘 婷, 刘仁义, 刘 南, 张 丰. 基于 REST 面向资源的地理信息服务设计[J]. *计算机工程*, 2011, 37(8): 238–240.
- MAO Feng, LIU Ting, LIU Ren-yi, LIU Nan, ZHANG Feng. Design of resource-oriented geospatial information service based on REST [J]. *Computer Engineering*, 2011, 37(8): 238–240.
- [6] 张利利, 刘峻明, 张 旭, 于新文. 基于 REST 架构模式的遥感影像数据服务设计[J]. *计算机应用*, 2010, 30(S1): 57–59.
- ZHANG Li-li, LIU Jun-ming, ZHANG Xu, YU Xin-wen. Design of remote sensing image service based on REST [J]. *Journal of Computer Applications*, 2010, 30(S1): 57–59.
- [7] Db4objects, Inc. db4objects [EB/OL]. [2011–02–22]. <http://www.db4o.com>.
- [8] 万常选, 刘喜平. XML 数据库技术(第2版)[M]. 北京: 清华大学出版社, 2008: 72–77.
- WAN Chang-xuan, LIU Xi-ping. XML database technology (2nd edition) [M]. Beijing: Tsinghua University Press, 2008: 72–77.
- [9] LEONIDAS F, RAMEZ E. Query engines for web-accessible XML data [C]// PETER M G, ATZENI A P, CERI S, PARABOSCHI S, RAMAMOCHANARAO K, SNODGRASS R T. *Proceeding of the 27th VLDB Conference*. Roma, Italy: Morgan Kaufmann, 2001: 251–260.
- [10] FIELDNIG R T. Architectural styles and the design of network-based software architectural doctoral dissertation [D]. California: University of California Irvine, 2000: 76–106.
- [11] Open Geospatial Consortium, Inc. OpenGIS geography markup language (GML) encoding Standard 3.2.1 [EB/OL]. [2011–06]. <http://www.opengeospatial.org/standards/gml>.
- [12] ZHANG J M. vtd-xml [EB/OL]. [2011–06]. <http://vtd-xml.sourceforge.net>.
- [13] 胡立辉, 周春华. 基于.NET 的 XQuery 处理器的实现及性能分析[J]. *长沙理工大学学报: 自然科学版*, 2005, 2(2): 57–62.
- HU Li-hui, ZHOU Chun-hua. Implementation and performance analysis of an XQuery processor based on .NET [J]. *Journal of Changsha University of Science and Technology: Natural Science*, 2005, 2(2): 57–62.
- [14] AltovaXML. AltovaXML-XSLT 1.0/2.0 Engine, XQuery Engine, XML Validator [EB/OL]. [2011–06]. <http://www.altova.com/altovaxml.html>.
- [15] KAY M H. The SAXON XSLT and XQuery processor [EB/OL]. [2011–06]. <http://saxon.sourceforge.net>.

(编辑 何学锋)