

科学数据共享工程技术标准

SDS/T 2221.2—2004

数据访问服务接口规范 ——WEB 要素服务规范

（征求意见稿）

（本稿完成日期：2005 年 5 月）

中华人民共和国科学技术部 发布

目 次

目 次.....	I
前 言.....	II
引 言.....	III
WEB 地物要素实现规范	1
1 范围.....	1
2 规范性引用文件.....	2
3 术语和约定.....	3
4 基本服务元素.....	5
5 公共元素.....	8
6 DescribeFeatureType 操作	15
7 GetFeature 操作	24
8 LockFeature 操作	36
9 Transaction 操作	44
10 GetCapabilities 操作	60
11 名一值” 对编码.....	68
附 录 A XML 模式定义（规范性附录）	82
附 录 B 一致性测试（规范性附录）	104

前 言

本规范修改采用国际标准“OpenGIS Web Feature Service Implementation Specification”。

附录A和B是规范性的。

本标准第一次制定。

本标准由中华人民共和国科学技术部基础研究司提出。

本标准由中华人民共和国科学技术部基础研究司归口。

本标准主要起草单位：国家信息中心。

本标准主要起草人：徐枫、宦茂盛等。

引 言

本标准描述了使用HTTP协议通过网络访问要素服务的规范，是数据访问服务接口之一。

网络地图服务允许客户端重叠显示来自Internet的多个网络地图服务的地图图像。类似的，WEB要素服务（网络要素服务）允许客户端从多个WEB要素服务（网络要素服务）中取得使用地理标记语言（GML）编码的地理空间数据。

实现WEB要素服务（网络要素服务）的必要条件：

接口必须用XML定义。

GML必须被用来在接口中表达要素。

一个最小化的WFS必须可以使用GML表达要素。

谓词或过滤语言将使用XML定义。

用来存储要素的数据库对客户端应用应该是不透明的，它们对数据的唯一查看方法应通过WFS接口。

使用XPath表达式子集引用特性。

本文档的目的是描述WEB要素服务（网络要素服务）接口，如图1所示。

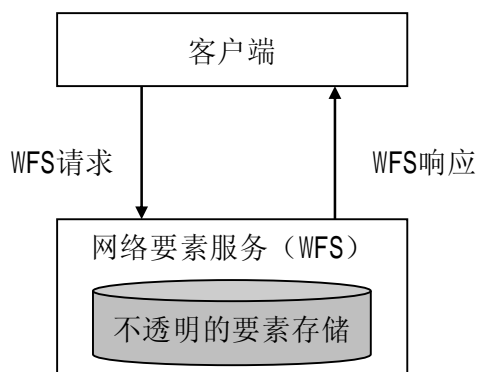


图1 WEB 要素服务（网络要素服务）

本规范是科学数据共享工程从国家863中引入的标准（中间成果）。在科学数据共享工程中的用途：（加图）

WEB 地物要素实现规范

1 范围

本规范描述了WEB要素服务（网络要素服务）(WFS)操作。WFS操作支持在使用HTTP协议的分布式计算平台上对要素进行插入、更新、删除、查询、发现操作。

在科学数据共享工程中，本规范的作用是提供给用户访问数据所需遵循的接口规范，该接口规范规定了的数据访问所使用的语言、参数、命名域，服务响应、数据返回的格式等多方面的内容，指导科学数据共享工程数据访问系统的开发与实现。

在本文档的上下文环境中，事务是一个逻辑工作单元，是由一个或多个数据操作组成的。要素以持久存储的方式保存在本文档中没有做陈述，因此，事务的语义，如原子失败，是假设不存在的。在和持久存储要素的数据存储系统的交互中保证数据改变的一致性WEB要素服务（网络要素服务）的功能。然而，本文也承认许多系统支持标准并发事务语义的事实，因此推荐一些可选的操作，这些操作允许WEB要素服务（网络要素服务）利用这些系统的有利条件（例如基于SQL的关系型数据库）。

地理要素

本文采纳了地理标记语言(GML)实现规范[2]中解释的地理要素的相同概念，也就是说，地理要素的状态是由一组特性来描述的，每个特性可以被想成（名字，类型，值）对，每个特性的名字和类型由地理要素的类型决定。地理要素至少有一个特性是由几何赋值的。这意味着要素是可以定义为根本没有几何特性的。地理要素的几何特性被限定为简单几何特性。一个简单的几何特性的坐标被定义为二维坐标，并且曲线的描述来自线性内插。在二维空间参考系统中定义的传统0,1和2维几何要素表现为点、线和面。另外，几何模型允许一个几何（Geometry）本身是其它多个几何（Geometry）的集合，可以是同构的几何，也可以是异构的几何。最后，GML允许要素具有复合或聚集型的特性。

处理请求

本文的这一部分简要介绍处理WEB要素服务（网络要素服务）请求要遵循的协议，处理请求的过程如下：

客户端应用请求WFS的能力文档。这个文档包含对WFS支持的所有操作的描述和它可以提供服务的要素类型列表。

客户端应用（可选的）向WEB要素服务（网络要素服务）发出请求，请求得到WFS可以服务的一个或多个要素类型的定义。

以要素类型的定义为基础，客户端应用生成本文档指定的请求。

请求，提交给Web服务器。

WFS被调用，去读取请求并为之提供服务。

当WFS完成请求处理，它生成一个状态报告，并把它传回客户端。万一有错误发生，状态报告会将其指出。

操作

为了支持事务和查询操作，以下操作必须定义：

GetCapabilities

一个WEB要素服务（网络要素服务）必须可以描述它的能力，特别地，必须指出它可以为那些要素类型提供服务以及每个要素类型支持那些操作。

DescribeFeatureType

WEB要素服务（网络要素服务）必须可以在请求中描述它可以提供服务的任何要素的结构。

GetFeature

WEB要素服务（网络要素服务）必须可以为一个获取要素实例的请求提供服务。另外，客户端可以规定获取要素的那些特性，并且可以使用空间和非空间的查询限制条件。

Transaction

WEB要素服务（网络要素服务）可以为事务请求提供服务。一个事务请求是由修改要素的操作组成的，即对地理要素的创建、修改和删除操作。

LockFeature

WEB要素服务（网络要素服务）可以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。这就保证了可序列化的事务被支持。

基于对以上操作的描述，可以定义两种类型的要素服务：

基本 WFS

基本WFS可以实现GetCapabilities,DescribeFeatureType和GetFeature操作。它被认为是只读的WEB要素服务（网络要素服务）。

事务 WFS

事务WFS可以支持基本WFS要素服务的所有操作，并且它可以实现事务操作。可选地，事务WFS可以实现LockFeature操作。

图1是一个简化了的协议图，它描述了为了处理一个典型的事务请求，客户端和WEB要素服务（网络要素服务）间来回传递的数据。

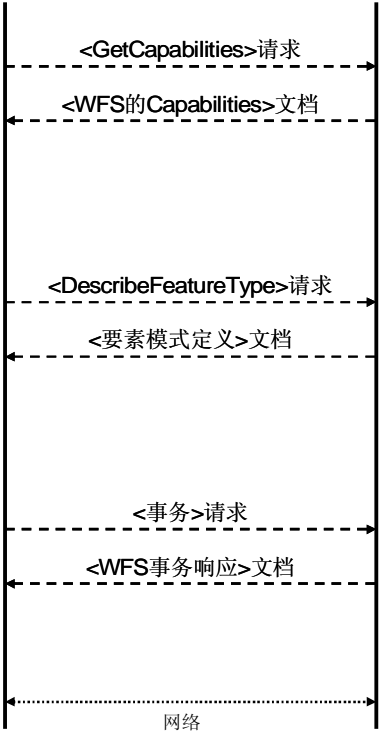


图2 协议图

2 规范性引用文件

[1] Bradner, Scott, "RFC 2119 Key words for use in RFCs to Indicate Requirement Levels," March 1997, <ftp://ftp.isi.edu/in-notes/rfc2119.txt> .

- [2] Cox, S., Cuthbert, A., Lake, R., and Martell, R. (eds.), "OpenGIS Implementation Specification #02-009: OpenGIS® Geography Markup Language (GML) Implementation Specification, version 2.1.1", April 2002
- [3] Vretanos, Panagiotis (ed.), "OpenGIS Implementation Specification #01-067: Filter Encoding Implementation Specification", May 2001
- [4] Percivall, George, ed., "The OpenGIS Abstract Specification, Topic 12: OpenGIS Service Architecture", 2002
- [5] Bray, Paoli, Sperberg-McQueen, eds., "Extensible Markup Language (XML) 1.0", 2nd edition, October 2000, W3C Recommendation, <http://www.w3.org/TR/2000/REC-xml>.
- [6] Beech, David, Maloney, Murry, Mendelson, Noah, Thompson, Harry S., "XML Schema Part 1: Structures", May 2001, W3C Recommendation, <http://www.w3c.org/TR/xmlschema-1>.
- [7] Bray, Hollander, Layman, eds., "Namespaces In XML", January 1999, W3C Recommendation, <http://www.w3.org/TR/2000/REC-xml-names>.
- [8] Clark, James, DeRose, Steve, "XML Path Language (XPath), Version 1.0", November 1999, W3C Recommendation, <http://www.w3c.org/TR/XPath>.
- [9] Fielding et. al., "Hypertext Transfer Protocol - HTTP/1.1," IETF RFC 2616, June 1999, <http://www.ietf.org/rfc/rfc2616.txt>.
- [10] Berners-Lee, T., Fielding, N., and Masinter, L., "Uniform Resource Identifiers (URI): Generic Syntax", IETF RFC 2396, <http://www.ietf.org/rfc/rfc2396.txt>.
- [11] National Center for Supercomputing Applications, "The Common Gateway Interface," <http://hoohoo.ncsa.uiuc.edu/cgi/>.
- [12] Freed, N. and Borenstein N., "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", IETF RFC 2045, November 1996, <http://www.ietf.org/rfc/rfc2045.txt>.
- [13] Internet Assigned Numbers Authority, <http://www.isi.edu/in-notes/iana/assignments/media-types/>
- [14] de La Beaujardière, Jeff (ed.), "OpenGIS Implementation Specification #01-047r2: Web Map Service Implementation Specification", June 2001
- [15] Vretanos, Panagiotis, "OpenGIS Discussion Paper: Transaction Encoding Specification Version 0.0.5", March 2000
- [16] Arctur, D., Pilkington, P., Cuthbert, A., "Spatial Object Transfer Format (SOTF): Initial High-Level Design, Version 1.2", Laser-Scan Inc., November 1999
- [17] Rumbach, James, et al., "Unified Modeling Language Reference Manual", 1999
- [18] Murata, St. Laurent, Kohn, "XML Media Types, January 2001, <http://www.ietf.org/rfc/rfc3023.txt>

3 术语和约定

3.1 约定

3.1.1 标准化动词

在已经标记标准化的章节中，本文档中的关键词“必须”，“必须不”，“必需的”，“将要”，“将要不”，“应该”，“应该不”，“推荐的”，“可能”，和“可选的”在Internet RFC 2119 [1]中进行了解释。

3.1.2 缩写词

CGI	公共网关接口 Common Gateway Interface
DCP	分布式计算平台 Distributed Computing Platform
DTD	文档类型定义 Document Type Definition
EPSG	欧洲石油调查组 European Petroleum Survey Group
GIS	地理信息系统 Geographic Information System
GML	地理标记语言 Geography Markup Language
HTTP	超文本传输协议 Hypertext Transfer Protocol
IETF	Internet 工程任务组 Internet Engineering Task Force
MIME	多目标 Internet 邮件扩展 Multipurpose Internet Mail Extensions
OGC	开放地理信息系统协会 Open GIS Consortium
OWS	OGC Web 服务 OGC Web Service
URL	统一资源定位符 Uniform Resource Locator
WFS	网络要素服务 (WEB 要素服务) Web Feature Service
XML	可扩展置标语言 Extensible Markup Language

3.1.3 示例的使用

本规范广泛的使用了XML示例。它们用来解释本规范讨论的WEB要素服务（网络要素服务）的不同方面。在努力保证范例格式良好并且有效的同时，也在许多方面为清晰作出牺牲。例如，许多例子用高亮的方式显示特别指出的部分，这会使得例子通不过XML验证工具的有效性验证。另外，大多数的例子引用的是虚构的服务器和数据。

因此，本规范并不断言从本文档拷贝的用XML或名值对编码的例子可以正确执行或可以通过特定的验证工具。只有标记为标准化的部分才是有效合法的XML或XML模式。

3.2 术语

3.2.1

XML文件

一个XML文件是一个实际的操作系统文件或任何有效的XML流。

3.2.2

2段式提交

2段式提交协议确保执行数据的同步提交。或者全部提交或者全部退回到每个分布式数据库。

3.2.3

DTD

文档类型定义：一个XML文档模式描述。

3.2.4

XML模式

模式描述语言，与DTD类似，基于XML自身。象DTD，能用于描述XML文档模式，但它可以灵活的描述其他的结构。与DTD不同，XML文档包括类型信息。XML模式定义在<http://www.w3c.org/TR/xmlschema-1/>

3.2.5

Xpath表达

Xpath表达是一个路径表达，与标识操作系统文件的路径类似，用于参考在XML文档中的元素。Xpath表达定义在<http://www.w3c.org/TR/Xpath/>

3.2.6

操作

使得一个对象被调用执行的转换或查询规范[4]。

3.2.7

接口

体现一个实体[4]行为特征的具有名称的操作集。

3.2.8

服务

由实体通过接口提供的独特的功能。

3.2.9

服务实例

一个服务的真正实现，服务实例与服务是同义的。

3.2.10

客户端

可以唤起服务器端操作的软件组件。

3.2.11

请求

客户端操作的一次调用。

3.2.12

响应

服务器端返回给客户端的操作结果。

3.2.13

能力XML

描述一个服务实例可用的操作和内容的服务级元数据。

3.2.14

空间参考系统

同 ISO19111中的定义。

3.2.15

不透明

不可见、不可访问或没有意义。

3.2.16

一致性

测试与本规范的一致性要使用附录B（标准化的）中所有的相关测试来检查。

3.2.17

地物

是指分布于地表之上的人工或自然物体，如建筑物、江河、森林等。

4 基本服务元素

4.1 介绍

本节描述了WEB要素服务（网络要素服务）的行为，这些行为独立于指定的操作，或是一些操作和接口共有的行为。

4.2 版本号和协商

4.2.1 版本号形式

已发布的规范版本号由3个正整数组成，并由小数点分隔，其形式为"x.y.z"。数字"y" 和 "z" 不能超过99。每个规范都独立地编号。

4.2.2 版本变化

一个特定规范的版本号**将会**随每次改版而变化。数字**将会**单调增加，并且**将会**由不多于3个由小数点分隔的正整数组成。第一个整数最有意义。在数字序列中，可能出现断沟，一些数字可能意味着试验的或过渡的版本。服务实例和客户端不需要支持所有的版本，但是**必须**遵守以下的协商规则。

4.2.3 在请求中和在服务元数据中的出现形式

版本号至少出现在2个地方：在描述服务的能力XML中和在客户端对这个服务请求的参数列表中。客户端请求的一个指定服务实例中使用的版本号**必须**等于这个实例已经声明支持的版本号（特别是在下面将要描述的协商中）。一个服务实例可以支持多种版本，客户端根据协商规则可以发现它们的值。

4.2.4 版本号协商

一个客户端会和一个服务实例协商以决定双方都同意的规范的版本号。协商是通过GetCapabilities操作完成的，并依据下面的规则。

所有的能力XML都必须包含协议版本号。为了响应一个包含版本号的GetCapabilities请求，WEB要素服务（网络要素服务）**必须**或者响应输出与这个规范版本一致的版本，**或者**在请求的版本号没有被服务器实现时协商一个双方都同意的版本。如果请求中没有声明版本号，服务器**必须**响应一个它能理解的最高的版本号，并标记相应的响应。

版本号协商如下：

如果服务器实现了请求中的版本号，服务器**必须**发送这个版本号。

如果客户端请求一个未知的版本号，此版本号高于服务器能理解的最低版本号，服务器**必须**发送低于请求版本号的最大版本号。

如果客户端发送的版本号低于服务器所能理解的所有版本号，服务器**必须**发送一个它能理解的最低版本号。

如果客户端不能理解服务器传递的新版本号，它**可能**或者停止与服务器通信，**或者**发送一个带有客户端能理解的新版本号的请求，但这个版本号必须小于服务器发送的版本号（如果服务器已经响应了一个低版本号）。

如果服务器已经响应了一个比较高的版本号（因为请求中的版本号低于任何服务器能理解的版本号），并且客户端不理解这个版本号，客户端**可能**会发送一个新的带有高于服务器发送的版本号的版本号。

这个处理过程会不断重复，直到取得双方都能理解的版本，或者直到客户端认为它不能和这个服务通信。

例1： 服务器理解版本1, 2, 4, 5 和 8。客户端理解版本1, 3, 4, 6, 和 7。客户端请求版本7。服务器响应版本5。客户端请求版本 4。服务器响应版本 4，客户端和服务器都理解版本4，协商成功结束。

例 2： 服务器理解版本4, 5 和 8。客户端理解版本3，客户端请求版本3。服务器响应版本4。客户端不能理解这个版本或任何更高的版本，因此协商失败，客户端停止和服务器通信。

4.3 通用 HTTP 请求规则

4.3.1 介绍

目前，WEB要素服务（网络要素服务）支持的唯一的分布式计算平台是World Wide Web本身，更为特别地，Internet上的主机实现了超文本传输协议（HTTP）[9]。因此，被服务实例支持的每个操作的在线资源是通过一个HTTP统一标识符(URL)定位的。对每个操作，URL可能是不同的，或是相同的，这由服务提供者决定。每个URL**必须**符合[9]中的描述。否则，是依赖实现的。只有构成服务请求的参数本身是由本规范管理的。

HTTP支持两种请求方法：GET和POST。两种方法中的任一种方法都可以用来描述特定的WEB要素服务（网络要素服务）类型，并且由服务实例提供。这两种方法使用在线资源的方法是不同的。

4.3.2 HTTP GET

一个用来构成HTTP GET请求的在线资源URL，事实上，是必须附加额外参数的URL前缀，它构成一个合法的操作指示。一个URL前缀被定义成一个不透明的字符串，包含协议、主机名、可选端口号、路径、问号‘?’以及可选的一个或多个服务器定义的参数，参数间以‘&’号分隔。前缀唯一地定位了一个特定的服务实例，客户端以名值对的形式（"name=value&"）附加必要的请求参数。结果URL必须符合HTTP公共网关接口(CGI)标准[7]，它托管在查询参数前的‘?’和每个参数间的‘&’。正如所有的CGI应用，查询URL[10]是被编码的，用以保护特殊字符。

URL前缀**必须**或者以一个‘?’结尾（在没有其它的服务器声明参数时），或者以一个‘&’结尾。实际上，客户端**应该**在增加本规范定义的操作参数之前，准备必要的‘?’或‘&’。

表 1 总结了一个操作请求URL的组成。

表1 一个通用的 Web 服务请求

URL 组成	描述
http://host[:port]/path?{name[=value]&}	服务操作的URL前缀。[]表示可选部分出现0次或1次；{ }表示出现0次或任意次。前缀完全由服务提供者控制。
name=value&	由WEB要素服务（网络要素服务）定义的一个或多个标准请求参数名/值对。必须的或可选的参数组成了实际参数列，由相应的规范的每个操作控制。

4.3.3 HTTP POST

一个用以构成HTTP POST的在线资源URL是一个完全的、合法的URL，客户端在POST文档体中提交经编码的请求。一个WFS**必须**不能有额外的参数附加在URL中构成对合法目标的操作请求。

4.4 通用 HTTP 响应规则

在取得一个合法的请求后，服务**必须**发送一个与这个请求相对应的响应。这个请求在适当的规范中描述。只有在版本协商时（如上文所述），服务器可能提供不同的结果。

若取得不合法的请求，服务**必须**产生一个服务异常。

注：作为一种特殊的方式，在WWW环境中客户端应准备接收合法的结果或什么也没有或者任何其他结果。这是因为客户端本身形成一个非一致请求，它可能在不经意间出发一个不是WEB要素服务（网络要素服务）的内容，因为服务本身是非一致的。

响应对象**必须**与此对象相应的多用途的网际邮件扩充协议（MIME）类型[9]相伴。

为了扩展的可能性，响应对象**应该**与其它实体头相伴。特别的Expire和Last-Modified头信息提供了高速缓存的重要信息。Content-Length可以被客户端用来了解什么时候传输结束，并且可以有效的分配结果空间，Content-Encoding或Content-Transfer-Encoding的正确解释结果可能是必要的。

4.5 对请求编码

本文档提供了两种方法对WFS请求进行编码，第一种方法使用XML作为编码语言。第二种方法使用“名值对”对请求中的可变参数编码。一个“名值对”的例子如下：

"REQUEST=GetCapabilities"

其中"REQUEST"是关键字，"GetCapabilities"是值。两种编码方法的请求得到的响应或是异常报告都是相同的。

表2列出了本规范中定义的WFS操作和它们的编码语义。

表2 对操作请求编码

操作	Request 编码
GetCapabilities	XML & KVP
DescribeFeatureType	XML & KVP

GetFeature / GetFeatureWithLock	XML & KVP
LockFeature	XML & KVP
Transaction	XML & limited KVP

KVP = keyword-value pair

本文档托管了使用GML进行对地理要素状态的XML编码。对编码的完全描述可以在文档[2]中找到。

4.6 名字空间

名字空间 (17) 是用来区分不同XML文档中的词汇的。WFS有3个标准化的名字空间定义, 即:

(<http://www.opengis.net/wfs>) - WFS 接口词汇

(<http://www.opengis.net/gml>) - GML 词汇

(<http://www.opengis.net/ogc>) - OGC 过滤词汇

一个给定的WFS实现将会用到一个或多个GML应用模式, 这些模式又会用到一个或多个名字空间(例如: <http://www.someserver.com/myns>)。本文档中许多例子使用的是单一名字空间, 但是如11.2.6节所示, 多名字空间是可以使用的。

5 公共元素

5.1 要素标识符

本文档假设一个特定的WFS实现可以操作的每个要素实例都是可以唯一标识的, 也就是说, 这个要素标识符在服务器上唯一的, 并且可以多次引用同一个要素实例(假设这个要素没有删掉)。更进一步的假设是要素标识符以OpenGIS® 过滤器编码实现规范[3]所述的方式来进行编码。一个要素标识符可以在任何需要引用要素的时候被使用。为了引用的目的从过滤编码规范中摘选了定义要素标识符的XML模式片断:

```
<xsd:element name="FeatureId" type="ogc:FeatureIdType"/>
<xsd:complexType name="FeatureIdType">
  <xsd:attribute name="fid" type="xsd:anyURI" use="required"/>
</xsd:complexType>
```

使用要素标识符的目的是使数据库操作成为可能。

5.1.1 全局唯一标识符 (资料性的)

如果仅是实现WEB要素服务(网络要素服务), 本地标识符已经足够了。但是, 需要对WEB要素服务(网络要素服务)中的所有种类的对象提供唯一的标识符, 为此, 要使用独立的范围和要素id组合的方法引用对象, 范围是指提供此要素服务的服务器的URL, 要素id是指要素的本地标识符。这种方法在其它上下文环境的传输和使用中会感到笨拙, 例如在注册时如果只想创建一个单一存储数据的实例(如一幅卫星图像)。

本节的目的是指出一个单一的全局唯一标识符字符串在上下文环境中使用是很方便。这样的字符串可以由一个WEB要素服务(网络要素服务)URL和本地标识符合并而成。

这个字符串在很多环境中可以被不透明的使用。但是如果它是一个真正的URL或是URN的话会更有用, 因为URL或URN可以用来直接访问用本地格式定义的对象。URL或URN是完全由规范明确的, 使用URN要注意: 没有多少实现能真正解析并获取数据对象, 它大多数时候只是用作唯一的标识符字符串。

对一个只需简单访问一个接口信息无需知道的对象的应用来说, 使用URL或URN是很有帮助的。这种访问/标识方法, 对于与高层次的XML技术(如RDF或XSLT)集成、甚至用于调试都是很有帮助的。

5.2 要素状态

由WFS提供服务的要素的定义是由GML应用模式提供的。第8节描述了客户端是怎样请求包含GML应用模式的XML文档的。GML应用模式中定义了一个或多个由WFS提供服务的要素类型定义。这种应用模式定义将会与OpenGIS 地理置标语言(GML)实现规范, 版本2.1.1 [2]一致。

一个客户端应用程序使用要素类型的GML应用模式定义引用这种地物类型的要素实例，以及引用这些要素实例的所有特性的名字和类型。一个要素实例的所有特性构成了这个要素的状态。一个客户端应用程序通过要素类型的名字和要素特性的名和值来引用要素实例。一个客户端应用程序通过插入、更新和删除操作请求使事务WFS改变地物的状态。

5.3 特性名

一个WEB要素服务（网络要素服务）引用GML应用模式中定义好的要素特性名。既然要素的状态必须是用GML即XML表达，WEB要素服务（网络要素服务）中使用的特性名也必须是合法的XML元素和属性名。另外，特性名也得符合XML[7]中对名字空间的描述。以下的定义是从可扩展置标语言(XML)1.0 [5]规范中的第2和3小节节选的：

```
[4] NCName ::= (Letter | '_' ) (NCNameChar)* /* An XML Name, minus the ":" */
[5] NCNameChar ::= Letter | Digit | '.' | '-' | '_' | CombiningChar | Extender
[6] QName ::= (Prefix ':')? LocalPart
[7] Prefix ::= NCName
[8] LocalPart ::= NCName
```

Letter, Digit, CombiningChar 和 Extender 的定义在文档 [5]的附录中。

例子

合法的特性名：

Age, Temperature, _KHz, myns:INWATERA_1M.WKB_GEOM

不合法的特性名：

+Domain, 123_SomeName

5.4 特性引用

5.4.1 介绍

正如在引言中提到的，GML允许地理要素具有复合或聚合的非几何特性。因此引出了问题，即这些特性怎样能在不同的地方被引用（例如查询或过滤表达式中）。正如在本文档中定义的，一个WFS必须使用XPath表达式去引用一个编码为XML元素或属性的要素的特性和子特性。

5.4.2 XPath 表达式

XML路径语言[8]规范是一种用来定位XML文档的语言，或是像在本规范中那样，通过XML元素或属性引用要素的特性。

本规范不要求一个WFS实现支持全部的XPath语言。为了保证实现入口代价的最小化，本规范要求一个WFS实现必须支持以下XPath语言的子集：

一个WFS实现必须支持缩写相对定位 路径。

相对定位路径由一个或多个步组成，并由路径分割符 ‘/’ 分割。

相对定位路径的第一个步 可能 与被引用的要素特性的根元素相对应，或者是与要素类型的根元素相对应，下一个步与被引用的要素特性的根元素相对应。

路径中的每个子步必须 由抽象形式的child:: 轴描述符组成，而且要素特性的名字被编码成元素基本结点类型。child:: 轴描述符只是省略了定位路径中描述符。

路径中的每个步可选的包含由谓语界定符 ‘[‘ 和 ‘]’ 组成的谓语，其中的数字表明上下文环境中哪个结点被选中。这就允许要素特性可以重复的引用。

路径中的最后一个步可以可选地由 attribute:: 轴描述符的抽象形式组成，‘@’ 和要素特性名被编码为属性的基本结点。

例子

从实践的角度解释如何使用XPath表达式引用复杂要素的特性（编码为元素或属性）。考虑一个假想的复杂地物Person，由以下的XML模式定义：

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

```

<schema
  targetNamespace="http://www.someserver.com/myns"
  xmlns:myns="http://www.someserver.com/myns"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  version="1.0">

  <import namespace="http://www.opengis.net/gml"
    schemaLocation="../gml/2.1/feature.xsd"/>

  <element name="Person" type="myns:PersonType"
    substitutionGroup="gml:_Feature"/>
  <complexType name="PersonType">
    <complexContent>
      <extension base="gml:AbstractFeatureType">
        <sequence>
          <element name="LastName" nillable="true">
            <simpleType>
              <restriction base="string">
                <maxLength value="30"/>
              </restriction>
            </simpleType>
          </element>
          <element name="FirstName" nillable="true">
            <simpleType>
              <restriction base="string">
                <maxLength value="10"/>
              </restriction>
            </simpleType>
          </element>
          <element name="Age" type="integer" nillable="true"/>
          <element name="Sex" type="string"/>
          <element name="Spouse">
            <complexType>
              <attribute name="sin" type="xsd:anyURI" use="required" />
            </complexType>
          </element>
          <element name="Location"
            type="gml:PointPropertyType"
            nillable="true"/>
          <element name="Address" type="myns:AddressType" nillable="true"/>
          <element name="Phone" type="xsd:string"

```

```

        minOccurs="0" maxOccurs="unbounded" />
    </sequence>
    <attribute name="sin" type="xsd:anyURI" use="required" />
</extension>
</complexContent>
</complexType>

<complexType name="AddressType">
    <sequence>
        <element name="StreetName" nillable="true">
            <simpleType>
                <restriction base="string">
                    <maxLength value="30" />
                </restriction>
            </simpleType>
        </element>
        <element name="StreetNumber" nillable="true">
            <simpleType>
                <restriction base="string">
                    <maxLength value="10" />
                </restriction>
            </simpleType>
        </element>
        <element name="City" nillable="true">
            <simpleType>
                <restriction base="string">
                    <maxLength value="30" />
                </restriction>
            </simpleType>
        </element>
        <element name="Province" nillable="true">
            <simpleType>
                <restriction base="string">
                    <maxLength value="30" />
                </restriction>
            </simpleType>
        </element>
        <element name="PostalCode" nillable="true">
            <simpleType>
                <restriction base="string">
                    <maxLength value="15" />
                </restriction>
            </simpleType>
        </element>
    </sequence>
</complexType>

```

```

        <element name="Country" nillable="true">
            <simpleType>
                <restriction base="string">
                    <maxLength value="30"/>
                </restriction>
            </simpleType>
        </element>
    </sequence>
</complexType>
</schema>

```

注意特性Address是一个AddressType类型的复杂特性，一个要素Person的实例如下：

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

```

<myns:Person
    sin="111222333"
    xmlns:myns="http://www.opengis.net/myns"
    xmlns:gml="http://www.opengis.net/gml"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.opengis.net/myns Person.xsd">

    <myns:LastName>Smith</myns:LastName>
    <myns:FirstName>Fred</myns:FirstName>
    <myns:Age>35</myns:Age>
    <myns:Sex>Male</myns:Sex>
    <myns:Spouse sin="444555666" />
    <myns:Location>
        <gml:Point><gml:coordinates>15,15</gml:coordinates></gml:Point>
    </myns:Location>
    <myns:Address>
        <myns:StreetName>Main St.</myns:StreetName>
        <myns:StreetNumber>5</myns:StreetNumber>
        <myns:City>SomeCity</myns:City>
        <myns:Province>SomeProvince</myns:Province>
        <myns:PostalCode>X1X 1X1</myns:PostalCode>
        <myns:Country>Canada</myns:Country>
    </myns:Address>
    <myns:Phone>416-123-4567</myns:Phone>
    <myns:Phone>416-890-1234</myns:Phone>
</myns:Person>

```

使用XPath [8]表达式，Person要素的每个特性可以被引用如下（为了清楚的缘故，省略了名字空间描述符）：

LastName

FirstName
 Age
 Sex
 Source
 Location
 Address
 Address/StreetNumber
 Address/StreetName
 Address/City
 Address/Province
 Address/Postal_Code
 Address/Country
 Phone[1]
 Phone[2]

在这个实例中，每个相对定位路径是从被引用的要素特性的根元素开始的，它只是简单地与要素特性的名字相对应。可选地，每个要素特性可以通过相对定位路径被引用，这个相对定位路径是由要素的根元素开始的（例如要素类型的名字），因此LastName特性可以被引用为Person/LastName，City特性可以被引用为Person/Address/City等等。

路径中的每个step由抽象的child::轴描述符（例如child::轴描述符被省略）组成，并且指定特性的名字是element结点类型。

元素Phone出现了多次，指示符[1] 和[2]用来指定某个元素。指示符[1]通常用来指示第一个出现Phone元素，指示符[2]通常用来指示第二个出现Phone元素

5.5 <native>元素

很清楚，一个开放的接口只能支持一定的公共能力集。<Native>元素的目的是允许访问由提供商提供的任何特殊的WEB要素服务（网络要素服务）或数据存储的特定能力。

<Native>元素在下面的XML模式中被定义：

```

<xsd:element name="Native" type="wfs:NativeType"/>
<xsd:complexType name="NativeType">
  <xsd:any />
  <xsd:attribute name="vendorId" type="xsd:string" use="required"/>
  <xsd:attribute name="safeToIgnore" type="xsd:boolean" use="required"/>
</xsd:complexType>
  
```

<Native>元素包含了提供商提供的特殊命令或操作。

vendorId 属性用来认证识别包含在<Native>元素中的命令和操作的提供商。这个属性提供了一种方式允许一个WEB要素服务（网络要素服务）决定它是否可以处理这个命令。

当<Native>中的命令或操作不被识别时，safeToIgnore属性被用来指导WEB要素服务（网络要素服务）采取行动。safeToIgnore有两个可能值：True 或 False。他们有如下含义：

safeToIgnore=False

值 False 表明<Native>元素不能被忽略，否则如果WEB要素服务（网络要素服务）不能处理操作，<Native> 元素中关联的操作将失败。

safeToIgnore=True

值 True 表明 <Native> 元素可以被安全的忽略。

实例

这个例子说明使用<Native>元素可以使使用一个基于SQL关系数据库的特殊要素成为可能。在这个例子中，元素表明Oracle命令可以被安全地忽略。

```
<Native vendorId="Oracle" safeToIgnore="True">
  ALTER SESSION ENABLE PARALLEL DML
</Native>
```

5.6 过滤器

过滤器是用来定义一组要被施加操作的地物实例。这个集合可以由一个或多个枚举要素组成，也可以由对要素类型的几何和标量的特性的空间和而非空间的限制条件定义的要素集组成。过滤器规范**将会**按过滤器编码实现规范[3]中描述的那样定义。

5.7 异常报告

当网络要素处理请求时遇到错误或是收到一个不可识别的请求，它**将会**生成一个XML文档，指明有错误发生。XML错误响应的格式由在A.2节定义的异常响应响应模式指定，并且必须经过有效性验证。

<ServiceExceptionReport> 元素可以包含一个或多个WFS处理异常。强制属性version用来表示服务异常报告模式的版本，对于这个版本的规范，它的值固定为1.2.0。

单个的异常信息包含在<ServiceException>元素中。可选的code属性用来把异常代码与相应的信息联系在一起。可选的locator属性可以用来指出异常是在生成异常的请求中的哪个部分发生的。本文档中定义了许多元素，其中包括handle属性。handle属性被用来联系元素和它的助记名。如果handle属性存在，它的值将被<ServiceException>元素的locator属性报告，如果handle属性没有被指定，web服务规范将尝试用其他的方法来定位错误，如用行号等方法。

实例

以下是一个异常报告的例子。这个异常报告表明第一个插入语句失败，因为在请求中缺少了XML关闭标签。

```
<?xml version="1.0" ?>
<ServiceExceptionReport
  version="1.2.0"
  xmlns="http://www.opengis.net/ogc"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/ogc ../wfs/1.0.0/OGC-exception.xsd">
  <ServiceException code="999" locator="INSERT STMT 01">
    parse error: missing closing tag for element WKB_GEOM
  </ServiceException>
</ServiceExceptionReport>
```

应该注意这个例子的输出验证了前面介绍的异常报告模式。

5.8 通用 XML 属性

5.8.1 版本属性

所有用XML编码的WFS请求都包含version属性。强制属性version用来指明被编码的请求遵循的WFS规范的版本号。这个版本号用于版本的协商，这在6.2.4节中已作了说明。version属性的默认值是1.0.0，与本文档的版本号相同。

5.8.2 服务属性

用XML编码的WFS请求都包含service属性。强制属性service用来指明一个特定服务实例能提供服务的可用的服务类型。当调用一个WEB要素服务（网络要素服务）service属性的值应该是WFS。

5.8.3 Handle 属性

handle属性的目的是允许客户端应用程序出于错误处理的目的将助记名和请求联系起来。如果handle被指定，并且遇到了异常，WEB要素服务（网络要素服务）可以使用handle去识别出错的元素。

6 DescribeFeatureType 操作

6.1 介绍

DescribeFeatureType操作的功能是生成一个模式描述WFS实现所能提供服务的要素类型。模式定义了如何编码WFS实现所期望的在输入要素实例以及输出时如何生成一个要素实例。

6.2 请求

DescribeFeatureType元素包含0个或1个TypeName元素。TypeName元素用来对将要描述的要素类型名进行编码。如果DescribeFeatureType元素是空的，它就被解释为请求所以WFS可以提高服务的要素类型。对DescribeFeatureType请求的XML编码用以下XML模式片段定义。

```
<xsd:element name="DescribeFeatureType" type="wfs:DescribeFeatureType"/>
<xsd:complexType name="DescribeFeatureType">
  <xsd:sequence>
    <xsd:element name="TypeName" type="xsd:QName"
      minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="version"
    type="xsd:string" use="required" fixed="1.0.0"/>
  <xsd:attribute name="service"
    type="xsd:string" use="required" fixed="WFS"/>
  <xsd:attribute name="outputFormat"
    type="xsd:string" use="optional" default="XMLSCHEMA"/>
</xsd:complexType>
```

outputFormat属性被用来指明用来描述要素类型模式描述语言。响应DescribeFeatureType操作的唯一的强制性输出格式是XML模式，用outputFormat属性的XML模式值表示。其他提供商指定格式也可行，但必须在能力文档中作出声明。

由GML [2]指定，要素模式定义完全由特定的WFS实现决定，WFS实现可以描述它的要素类型，仅有的限制条件是：

要素的几何信息必须用GML几何描述表达(geometry.xsd)。

空间参考系统必须如OpenGIS® 地理置标语言(GML)实现规范，版本2.1.1 [2]中定义的那样表达。

要素模式必须与要素模型一致。这意味着要素模式定义要素的特性。这个陈述的GML解释为：嵌套在要素类型根元素下的元素定义这个要素的特性。

6.3 响应

在响应一个DescribeFeatureType请求时，其outputFormat属性值已经被设为XML模式，WFS实现必须能提供一个合法GML[2]应用Schema的XML[6]模式，并且定义列在请求列表中的要素类型。DescribeFeatureType请求提交的文档可以被用来以要素集合的形式验证WFS输出生成的要素实例的合法性，或者，用来进行事务操作的由输入指定的要素实例。

使用其它schema描述语言（如DTD）的schema描述也是允许的，只要其在能力文档中被声明。

6.3.1 支持多名字空间

一个XML模式[6]文档只能描述属于单一名字空间的元素。这意味着一个WEB要素服务（网络要素服务）不能用一个单一的XML模式文档描述多名字空间的要素，为了克服这个局限性，WFS可以生成一个“打包”的XML模式文档，它根据需求引入不同名字空间的模式。例如，考虑以下请求：

```

<?xml version="1.0" ?>
<DescribeFeatureType
  version="1.0.0"
  service="WFS"
  xmlns="http://www.opengis.net/wfs"
  xmlns:ns01="http://www.server01.com/ns01"
  xmlns:ns02="http://www.server02.com/ns02"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-basic.xsd">
  <TypeName>ns01:TREESA_1M</TypeName>
  <TypeName>ns02:ROADL_1M</TypeName>
</DescribeFeatureType>

```

一个WFS对这个请求的如下响应:

```

<?xml version="1.0" ?>
<schema
  xmlns="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">

  <import namespace="http://www.server01.com/ns01"
    schemaLocation="http://www.myserver.com/wfs.cgi?
      request=DescribeFeatureType&type=ns01:TREESA_1M"/>

  <import namespace="http://www.server02.com/ns02"
    schemaLocation="http://www.yourserver.com/wfs.cgi?
      request=DescribeFeatureType&type=ns02:ROADL_1M"/>

</schema>

```

在这个例子中, WFS使用DescribeFeatureType请求去获得不同名字空间的要素模式, 这只是个例子, 其它获得模式的方法也可以实现(如: 引用静态模式文档)。

6.4 异常

当Web Feature服务在服务一个DescribeFeatureType请求时遇到错误, 它将会生成一个在7.7节中描述的异常。

6.5 示例

例1

考虑一下定义在SQL数据库中的TREESA_1M和ROADL_1M两个地理要素类型, 数据库对这两个要素类型的描述报告如下:

```

SQL> describe TREESA_1M

```

Name	Null?	Type
-----	-----	-----
WKB_GEOM	NOT NULL	LONG RAW
ID		NUMBER(10)
TREE_TYPE		VARCHAR2(80)

SQL> describe ROADL_1M

Name	Null?	Type
-----	-----	-----
WKB_GEOM	NOT NULL	LONG RAW
DESIGNATION		VARCHAR2(30)
SURFACE_TYPE		VARCHAR2(30)
NLANES		NUMBER(2)

对DescribeFeatureType型请求的响应:

```
<?xml version="1.0" ?>
<DescribeFeatureType
  version="1.0.0"
  service="WFS"
  xmlns="http://www.opengis.net/wfs"
  xmlns:myns="http://www.myserver.com/myns"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-basic.xsd">
  <TypeName>myns:TREESA_1M</TypeName>
  <TypeName>myns:ROADL_1M</TypeName>
</DescribeFeatureType>
```

一个WEB要素服务（网络要素服务）会产生如下XML模式文档:

```
<?xml version="1.0" ?>
<schema
  targetNamespace="http://www.someserver.com/myns"
  xmlns:myns="http://www.someserver.com/myns"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:gml="http://www.opengis.net/gml"
  elementFormDefault="qualified" version="0.1">

  <import namespace="http://www.opengis.net/gml"
    schemaLocation="../gml/2.1/feature.xsd"/>

  <!-- =====
    define global elements
    ===== -->

  <element name="TREESA_1M"
    type="myns:TREESA_1M_Type"
    substitutionGroup="gml:_Feature"/>

  <element name="ROADL_1M"
    type="myns:ROADL_1M_Type"
    substitutionGroup="gml:_Feature"/>
```

```

<!-- =====
      define complex types (classes)
      ===== -->
<complexType name="TREESA_1M_Type">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <element name="WKB_GEOM"
          type="gml:PolygonPropertyType" nillable="false"/>
        <element name="ID" nillable="true" minOccurs="0">
          <simpleType>
            <restriction base="integer">
              <totalDigits value="10"/>
            </restriction>
          </simpleType>
        </element>
        <element name="TREE_TYPE" nillable="true" minOccurs="0">
          <simpleType>
            <restriction base="string">
              <maxLength value="80"/>
            </restriction>
          </simpleType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<complexType name="ROADL_1M_Type">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <element name="WKB_GEOM"
          type="gml:LineStringPropertyType"
          nillable="false"/>
        <element name="DESIGNATION" nillable="true" minOccurs="0">
          <simpleType>
            <restriction base="string">
              <maxLength value="30"/>
            </restriction>
          </simpleType>
        </element>
        <element name="SURFACE_TYPE" nillable="true" minOccurs="0">

```

```

    <simpleType>
      <restriction base="string">
        <maxLength value="30"/>
      </restriction>
    </simpleType>
  </element>
  <element name="NLANES" nillable="true" minOccurs="0">
    <simpleType>
      <restriction base="integer">
        <totalDigits value="2"/>
      </restriction>
    </simpleType>
  </element>
</sequence>
</extension>
</complexContent>
</complexType>
</schema>

```

客户端使用这个模式描述,可以表达TREESA_1M要素实例和/或ROADL_1M要素实例的状态,如下例所示:

```

<?xml version="1.0" ?>
<wfs:FeatureCollection
  xmlns="http://www.someserver.com/myns"
  xmlns:myns="http://www.someserver.com/myns"
  xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-basic.xsd
    http://www.someserver.com/myns ex07.xsd">
  <gml:boundedBy>
    <gml:Box srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
      <gml:coordinates>-180.0,-90.0 180.0,90.0</gml:coordinates>
    </gml:Box>
  </gml:boundedBy>
  <gml:featureMember>
    <TREESA_1M>
      <WKB_GEOM>
        <gml:Polygon>
          <gml:outerBoundaryIs>
            <gml:LinearRing>
              <gml:coordinates decimal="." cs="," ts=" ">-120.000000,65.588264
-120.003571,65.590782 -120.011292,65.590965 -120.022491,65.595215 -120.031212,65.592880
-120.019363,65.586121 -120.030350,65.585365 -120.045082,65.581848 -120.059540,65.584938

```

```

-120.067284,65.590500 -120.067284,65.595436 -120.067337,65.613441 -120.067337,65.613777
-120.060997,65.606346 -120.045517,65.605545 -120.022675,65.599777 -120.003975,65.601036
-120.000000,65.602081 -120.000000,65.602081 -120.000000,65.588264</gml:coordinates>
    </gml:LinearRing>
  </gml:outerBoundaryIs>
</gml:Polygon>
</WKB_GEOM>
<ID>0000000002</ID>
<TREE_TYPE>Maple</TREE_TYPE>
</TREESA_1M>
</gml:featureMember>
<gml:featureMember>
  <ROADL_1M>
    <WKB_GEOM>
      <gml:LineString srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
        <gml:coordinates decimal="." cs="," ts=" " >-59.478340,-52.226578
-59.484871,-52.223564 -59.488991,-52.198524 -59.485958,-52.169559 -59.480400,-52.152615
-59.465576,-52.141491 -59.462002,-52.136417 -59.447968,-52.127190 -59.422928,-52.120701
-59.411915,-52.117844 -59.397972,-52.116440 -59.371311,-52.121300</gml:coordinates>
      </gml:LineString>
    </WKB_GEOM>
    <DESIGNATION>HYW 401</DESIGNATION>
    <SURFACE_TYPE>ASPHALT</SURFACE_TYPE>
    <NLANES>12</NLANES>
  </ROADL_1M>
</gml:featureMember>
</wfs:FeatureCollection>

```

例2

这个例子描述了一个集合类型：People，它是由要素类型Person组成，Person中包含复杂特性Address。

对DescribeFeatureType型请求的响应

```

<?xml version="1.0" ?>
<DescribeFeatureType
  version="1.0.0"
  service="WFS"
  outputFormat="XMLSCHEMA"
  xmlns="http://www.opengis.net/wfs"
  xmlns:myns="http://www.myserver.com/myns"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-basic.xsd">
  <TypeName>myns:People</TypeName>
</DescribeFeatureType>

```

一个WEB要素服务（网络要素服务）会产生如下XML模式文档：

```

<?xml version="1.0" ?>
<xsd:schema
  targetNamespace="http://www.someserver.com/myns"
  xmlns:myns="http://www.someserver.com/myns"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" version="0.1">

  <xsd:import namespace="http://www.opengis.net/gml"
    schemaLocation="../../gml/2.1/feature.xsd"/>

  <xsd:element name="Person"
    type="myns:PersonType"
    substitutionGroup="gml:_Feature"/>

  <xsd:complexType name="PersonType">
    <xsd:complexContent>
      <xsd:extension base="gml:AbstractFeatureType">
        <xsd:sequence>
          <xsd:element name="LastName" nillable="true">
            <xsd:simpleType>
              <xsd:restriction base="string">
                <xsd:maxLength value="30"/>
              </xsd:restriction>
            </xsd:simpleType>
          </xsd:element>
          <xsd:element name="FirstName" nillable="true">
            <xsd:simpleType>
              <xsd:restriction base="string">
                <xsd:maxLength value="10"/>
              </xsd:restriction>
            </xsd:simpleType>
          </xsd:element>
          <xsd:element name="Age"
            type="integer"
            nillable="true"/>
          <xsd:element name="Sex"
            type="string"/>
          <xsd:element name="Location"
            type="gml:PointPropertyType"
            nillable="true"/>
          <xsd:element name="Address"

```

```

        type="myns:AddressType"
        nillable="true"/>
    </xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="AddressType">
    <xsd:sequence>
        <xsd:element name="StreetName" nillable="true">
            <xsd:simpleType>
                <xsd:restriction base="string">
                    <xsd:maxLength value="30"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="StreetNumber" nillable="true">
            <xsd:simpleType>
                <xsd:restriction base="string">
                    <xsd:maxLength value="10"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="City" nillable="true">
            <xsd:simpleType>
                <xsd:restriction base="string">
                    <xsd:maxLength value="30"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="Province" nillable="true">
            <xsd:simpleType>
                <xsd:restriction base="string">
                    <xsd:maxLength value="30"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="PostalCode" nillable="true">
            <xsd:simpleType>
                <xsd:restriction base="string">
                    <xsd:maxLength value="15"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>

```

```

    <xsd:element name="Country" nillable="true">
      <xsd:simpleType>
        <xsd:restriction base="string">
          <xsd:maxLength value="30"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
</xsd:schema>

```

符合这个模式的实例文档如下:

```

<?xml version="1.0" ?>
<wfs:FeatureCollection
xmlns="http://www.someserver.com/myns"
xmlns:myns="http://www.someserver.com/myns"
xmlns:wfs="http://www.opengis.net/wfs"
xmlns:gml="http://www.opengis.net/gml"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-basic.xsd
http://www.someserver.com/myns ex10.xsd">
  <gml:boundedBy>
    <gml:Box>
      <gml:coord>
        <gml:X>10</gml:X>
        <gml:Y>10</gml:Y>
      </gml:coord>
      <gml:coord>
        <gml:X>20</gml:X>
        <gml:Y>20</gml:Y>
      </gml:coord>
    </gml:Box>
  </gml:boundedBy>
  <gml:featureMember>
    <Person>
      <myns:LastName>Smith</myns:LastName>
      <myns:FirstName>Fred</myns:FirstName>
      <myns:Age>35</myns:Age>
      <myns:Sex>Male</myns:Sex>
      <myns:Location>
        <gml:Point><gml:coordinates>15,15</gml:coordinates></gml:Point>
      </myns:Location>
      <myns:Address>
        <myns:StreetName>Main St.</myns:StreetName>

```

```

        <myns:StreetNumber>5</myns:StreetNumber>
        <myns:City>SomeCity</myns:City>
        <myns:Province>SomeProvince</myns:Province>
        <myns:PostalCode>X1X 1X1</myns:PostalCode>
        <myns:Country>Canada</myns:Country>
    </myns:Address>
</Person>
</gml:featureMember>
</wfs:FeatureCollection>

```

7 GetFeature 操作

7.1 介绍

GetFeature操作允许从WEB要素服务（网络要素服务）取得要素。GetFeature请求是由WFS处理的，并且一个包含结果的XML文档会被返回客户端。

7.2 请求

由XML编码的GetFeature请求由下面的XML模式片段定义：

```

<xsd:element name="GetFeature" type="wfs:GetFeatureType"/>
<xsd:complexType name="GetFeatureType">
    <xsd:sequence>
        <xsd:element ref="wfs:Query" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="version"
        type="xsd:string" use="required" fixed="1.0.0"/>
    <xsd:attribute name="service"
        type="xsd:string" use="required" fixed="WFS"/>
    <xsd:attribute name="handle"
        type="xsd:string" use="optional"/>
    <xsd:attribute name="outputFormat"
        type="xsd:string" use="optional" default="GML2"/>
    </xsd:attribute>
    <xsd:attribute name="maxFeatures" type="xsd:positiveInteger"
        use="optional"/>
</xsd:complexType>
<xsd:element name="Query" type="wfs:QueryType"/>
<xsd:complexType name="QueryType">
    <xsd:sequence>
        <xsd:element ref="ogc:PropertyName" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="ogc:Filter" minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attribute name="handle"
        type="xsd:string" use="optional"/>
    <xsd:attribute name="typeName"
        type="xsd:QName" use="required"/>
    <xsd:attribute name="featureVersion"

```

```

        type="xsd:string" use="optional"/>
    </xsd:complexType>

```

〈GetFeature〉元素包含一个或多个〈Query〉元素，每个〈Query〉元素包含一个对查询的描述。GetFeature请求中的所有查询结果连结在一起组成结果集。

outputFormat属性定义了生成结果集时使用的格式。默认值GML2，表示GML[2]应被使用。提供商提供的指定格式（包括非XML形式和二进制形式），只要在能力文档中作出声明，也是可用的。

maxFeatures属性用来限制GetFeature请求取得的要素数，一旦MaxFeature限制达到了，结果就会在这点返回。

打包在GetFeature请求中的每个个体查询使用〈Query〉元素定义。〈Query〉元素定义查询什么样的要素类型，取得哪些特性以及对这些特性施加什么限制（空间的和非空间的）。

typeName属性用来表明查询的要素类型或类。

FeatureVersion属性是用来适应区分要素版本的系统的。值ALL表明应取回一个要素的所有版本，否则，整数n表示应取回n版本的要素。版本号从1开始，1是最老的版本，如果版本值大于指定的最大版本号，那么最近的版本将被返回。查询默认的操作是返回最近的版本。不支持分版本的系统可以忽略此参数，并返回它所有的唯一版本。

〈PropertyName〉元素用来列举查询中被选择的要素特性。这些特性的值将被包含进对GetFeature请求的响应中，客户端应用程序在构成GetFeature请求之前可以通过DescribeFeatureType决定要素特性。DescribeFeatureType操作将产生一个定义要素类型模式的GML应用模式。客户端由此可以选择需取回的特性。另外，客户端可以决定哪些特性是WFS生成要素类型实例时验证GML应用模式必要的必须要取回的特性。在WFS遇到没有选择要素的全部必要特性的查询时，WFS会在内部扩大特性列表以保证其包含所有必要特性名。因此，一个WFS客户端必须要准备处理这种情况：收到的特性值比请求的多。

如果没有〈PropertyName〉元素被指定，要素的所有特性将被取回。

〈Filter〉元素用被用来定义查询限制条件。无论空间的或非空间的限制条件都可以被指定，如在Filter编码规范[3]中描述的那样，如果在〈Query〉元素中没有〈Filter〉元素，那么查询没有限制条件，所有的要素实例被取回。

〈GetFeatureWithLock〉元素的功能与〈GetFeature〉元素功能相近，只是它向WEB要素服务（网络要素服务）表明要对被选择的要素上锁，假设要修改要素时。

7.3 响应

GetFeature请求的响应格式由outputFormat属性控制。outputFormat属性的默认值应该是GML2。这表明WFS必须生成结果集的GML文档，该文档应与OpenGIS地理置标语言实现规范一致版本2.1.1 [2]。另外，由DescribeFeatureType操作生成的输出必须是经过GML应用模式验证有效的。

所有（在响应outputFormat是GML2的查询时）由WFS实现生成的GML文档，必须参照相应的GML应用模式，以保证所有输出都是有效的，这可以通过使用[6]中定义的schemaLocation属性完成。这个属性提供一个或多个模式文档的物理定位的提示，这些模式文档被用来本地验证以及模式有效性评估。schemaLocation属性的值由一对值组成，第一个值表示名字空间，第二个值可以通过它描述哪儿可以找到适当的模式文档，模式文档的物理定位由URI [10]指定。

下面的XML片段表明schemaLocation属性在根元素中的使用，它表明的XML模式文档的定位用来验证有效性。

```

<?xml version="1.0" ?>
<wfs:FeatureCollection
  xmlns="http://www.opengis.net/myns"
  xmlns:myns="http://www.opengis.net/myns"
  xmlns:gml="http://www.opengis.net/gml"

```

```

xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation="http://www.opengis.net/myns
http://www.someserver.com/wfs.cgi?
request=DescribeFeatureType&typename=TREESA_1M,ROADL_1M"> ...

```

在这个例子中，相对于myns名字空间的模式文档通过发送DescribeFeatureType请求被动态生成，服务器生成输出和请求的模式，这个DescribeFeatureType操作请求TREESA_1M和ROADL_1M要素类型的模式，都是在myns名字空间。

由每个WFS实现安排GML输出适当的schemaLocation参考，这样的输出可以得到验证。

对于〈GetFeatureWithLock〉请求，WFS必须生成一个包括锁描述符的结果。锁描述符用lockId属性编码，lockId属性定义在〈wfs:FeatureCollection〉元素中，以下XML片段解释了如何在响应中包含lockId属性。

```

<wfs:FeatureCollection lockId="00A01"... >
...
</wfs:FeatureCollection>

```

省略意味着包含在GetFeatureWithLock响应中的所有组件与包含在GetFeature响应中的组件是完全一致的。

7.4 异常

当Web要素为GetFeature请求提供服务时遇到错误，它应该产生一个如7.7节所描述的异常。

7.5 示例

本节包含了GetFeature请求的数个实例，一些实例包括示例输出。

例1

这个示例取回指定的要素类型实例INWATERA_1M,其标识符为“INWATERA_1M.1234”。

```

<?xml version="1.0" ?>
<wfs:GetFeature
  service="WFS"
  version="1.0.0"
  outputFormat="GML2"
  xmlns:myns="http://www.someserver.com/myns"
  xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-basic.xsd">
  <wfs:Query typeName="myns:INWATERA_1M">
    <ogc:Filter>
      <ogc:FeatureId fid="INWATERA_1M.1234"/>
    </ogc:Filter>
  </wfs:Query>
</wfs:GetFeature>

```

例2

这个示例取回指定的要素类型实例INWATERA_1M的特性子集,通过请求取回的指定实例的要素标识符为“INWATERA_1M.1013”。

```

<?xml version="1.0" ?>
<wfs:GetFeature
  service="WFS"
  version="1.0.0"
  xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:myns="http://www.someserver.com/myns"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-basic.xsd">
  <wfs:Query typeName="myns:INWATERA_1M">
    <ogc:PropertyName>myns:WKB_GEOM</ogc:PropertyName>
    <ogc:PropertyName>myns:TILE_ID</ogc:PropertyName>
    <ogc:PropertyName>myns:FAC_ID</ogc:PropertyName>
    <ogc:Filter>
      <ogc:FeatureId fid="INWATERA_1M.1013"/>
    </ogc:Filter>
  </wfs:Query>
</wfs:GetFeature>

```

例3

在这个例子中，要素实例枚举列表中（所有要素类型为INWATERA_1M的）要素实例的特性都被取回。〈FeatureId〉元素用来识别每个取回的要素。

```

<?xml version="1.0" ?>
<GetFeature
  version="1.0.0"
  service="WFS"
  xmlns="http://www.opengis.net/wfs"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:myns="http://www.someserver.com/myns"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-basic.xsd">
  <Query typeName="myns:INWATERA_1M">
    <ogc:Filter>
      <ogc:FeatureId fid="INWATERA_1M.1013"/>
      <ogc:FeatureId fid="INWATERA_1M.1014"/>
      <ogc:FeatureId fid="INWATERA_1M.1015"/>
    </ogc:Filter>
  </Query>
</GetFeature>

```

例4

这个例子和上一个例子很相似，除了取回了枚举要素的部分特性，〈PropertyName〉元素用来列出要取得的特性。

```

<?xml version="1.0" ?>

```

```

<GetFeature
  version="1.0.0"
  service="WFS"
  xmlns="http://www.opengis.net/wfs"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:myns="http://www.someserver.com/myns"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-basic.xsd">
  <Query typeName="myns:INWATERA_1M">
    <ogc:PropertyName>myns:WKB_GEOM</ogc:PropertyName>
    <ogc:PropertyName>myns:TILE_ID</ogc:PropertyName>
    <ogc:Filter>
      <ogc:FeatureId fid="INWATERA_1M.1013"/>
      <ogc:FeatureId fid="INWATERA_1M.1014"/>
      <ogc:FeatureId fid="INWATERA_1M.1015"/>
    </ogc:Filter>
  </Query>
</GetFeature>

```

例5

取出所有要素类型为 INWATERA_1M 的实例，要素的最大个数是 10000。

```

<?xml version="1.0" ?>
<GetFeature
  version="1.0.0"
  service="WFS"
  maxFeatures="10000"
  xmlns="http://www.opengis.net/wfs"
  xmlns:myns="http://www.someserver.com/myns"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-basic.xsd">
  <Query typeName="myns:INWATERA_1M" />
</GetFeature>

```

例 6

下面这个没有限制条件的请求取回属于指定要素类型的所有要素实例。注意：要素类型并不都在一个名字空间。

```

<?xml version="1.0" ?>
<GetFeature
  version="1.0.0"
  service="WFS"
  xmlns="http://www.opengis.net/wfs"
  xmlns:myns="http://www.someserver.com/myns"
  xmlns:yourns="http://demo.cubewerx.com/yourns"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

```

```

xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-basic.xsd">
  <Query typeName="myns:INWATERA_1M"/>
  <Query typeName="myns:BUILTUPA_1M"/>
  <Query typeName="yourns:ROADL_1M"/>
</GetFeature>

```

例 7

下例选择了Grand Banks地区的HYDROGRAPHY要素的几何信息和深度。Grand Banks的边界矩形为：
[-57.9118,46.2023,-46.6873,51.8145]。

```

<?xml version="1.0" ?>
<GetFeature
  version="1.0.0"
  service="WFS"
  handle="Query01"
  xmlns="http://www.opengis.net/wfs"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:myns="http://www.someserver.com/myns"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-basic.xsd">
  <Query typeName="myns:HYDROGRAPHY">
    <ogc:PropertyName>myns:GEOTEMP</ogc:PropertyName>
    <ogc:PropertyName>myns:DEPTH</ogc:PropertyName>
    <ogc:Filter>
      <ogc:Not>
        <ogc:Disjoint>
          <ogc:PropertyName>myns:GEOTEMP</ogc:PropertyName>
          <gml:Box>
            <gml:coordinates>-57.9118,46.2023 -46.6873,51.8145</gml:coordinates>
          </gml:Box>
        </ogc:Disjoint>
      </ogc:Not>
    </ogc:Filter>
  </Query>
</GetFeature>

```

这个请求的输出结果可能是：

```

<?xml version="1.0" ?>
<wfs:FeatureCollection
  xmlns="http://www.someserver.com/myns"
  xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.someserver.com/myns HYDROGRAPHY.xsd

```

```

http://www.opengis.net/wfs ../wfs/1.0.0/WFS-basic.xsd">
<gml:boundedBy>
  <gml:Box srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
    <gml:coordinates>10,10 20,20</gml:coordinates>
  </gml:Box>
</gml:boundedBy>
<gml:featureMember>
  <HYDROGRAPHY fid="HYDROGRAPHY.450">
    <GEOTEMP>
      <gml:Point srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
        <gml:coordinates>10,10</gml:coordinates>
      </gml:Point>
    </GEOTEMP>
    <DEPTH>565</DEPTH>
  </HYDROGRAPHY>
</gml:featureMember>
<gml:featureMember>
  <HYDROGRAPHY fid="HYDROGRAPHY.450">
    <GEOTEMP>
      <gml:Point srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
        <gml:coordinates>10,11</gml:coordinates>
      </gml:Point>
    </GEOTEMP>
    <DEPTH>566</DEPTH>
  </HYDROGRAPHY>
</gml:featureMember>
<!--
.
. ... more HYDROGRAPHY instances ...
.
-->
</wfs:FeatureCollection>

```

例 8

本例描述了两个查询，取回感兴趣区域内的ROADS和RAILS实例：

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

```

<GetFeature
  version="1.0.0"
  service="WFS"
  handle="Example Query"
  xmlns="http://www.opengis.net/wfs"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:myns="http://www.someserver.com/myns"

```

```

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-basic.xsd">
<Query typeName="myns:ROADS">
  <ogc:PropertyName>myns:PATH</ogc:PropertyName>
  <ogc:PropertyName>myns:LANES</ogc:PropertyName>
  <ogc:PropertyName>myns:SURFACETYPE</ogc:PropertyName>
  <ogc:Filter>
    <ogc:Within>
      <ogc:PropertyName>myns:PATH</ogc:PropertyName>
      <gml:Box>
        <gml:coordinates>50,40 100,60</gml:coordinates>
      </gml:Box>
    </ogc:Within>
  </ogc:Filter>
</Query>
<Query typeName="myns:RAILS">
  <ogc:PropertyName>myns:TRACK</ogc:PropertyName>
  <ogc:PropertyName>myns:GAUGE</ogc:PropertyName>
  <ogc:Filter>
    <ogc:Within>
      <ogc:PropertyName>myns:TRACK</ogc:PropertyName>
      <gml:Box>
        <gml:coordinates>50,40 100,60</gml:coordinates>
      </gml:Box>
    </ogc:Within>
  </ogc:Filter>
</Query>
</GetFeature>

```

两个查询结果被连接起来形成输出的要素集合。

```

<?xml version="1.0" ?>
<wfs:FeatureCollection
  xmlns="http://www.someserver.com/myns"
  xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-basic.xsd
    http://www.someserver.com/myns ROADSRAILS.xsd">
  <gml:boundedBy>
    <gml:Box srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
      <gml:coordinates>0,0 180,360</gml:coordinates>
    </gml:Box>
  </gml:boundedBy>
  <gml:featureMember>

```

```

    <ROADS fid="ROADS.100">
      <PATH>
        <gml:LineString gid="1"
          srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
          <gml:coordinates>10,10 10,11 10,12 10,13</gml:coordinates>
        </gml:LineString>
      </PATH>
      <SURFACE_TYPE>ASPHALT</SURFACE_TYPE>
      <NLANES>4</NLANES>
    </ROADS>
  </gml:featureMember>
</gml:featureMember>
  <ROADS fid="ROADS.105">
    <PATH>
      <gml:LineString gid="2"
        srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
        <gml:coordinates>10,10 10,11 10,12</gml:coordinates>
      </gml:LineString>
    </PATH>
    <SURFACE_TYPE>GRAVEL</SURFACE_TYPE>
    <NLANES>2</NLANES>
  </ROADS>
</gml:featureMember>
<!--
... more ROADS features ....
-->
<gml:featureMember>
  <RAILS fid="RAILS.119">
    <TRACK>
      <gml:LineString gid="n"
        srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
        <gml:coordinates>15,10 16,11 17,12</gml:coordinates>
      </gml:LineString>
    </TRACK>
    <GAUGE>24</GAUGE>
  </RAILS>
</gml:featureMember>
<!--
... more RAILS features ....
-->
</wfs:FeatureCollection>

```

例 9

本例说明了要素的复杂特性是如何通过使用Xpath表达式引用的。要素类型Person定义如下：

This example illustrates how complex properties of features can be referenced using XPath expressions. Consider the feature type Person defined as:

```
<?xml version="1.0" ?>
<schema
  targetNamespace="http://www.opengis.net/myns"
  xmlns:myns="http://www.opengis.net/myns"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  version="1.0">

  <import namespace="http://www.opengis.net/gml"
    schemaLocation="../../gml/2.1/feature.xsd"/>

  <element name="Person" type="myns:PersonType"
    substitutionGroup="gml:_Feature"/>
  <complexType name="PersonType">
    <complexContent>
      <extension base="gml:AbstractFeatureType">
        <sequence>
          <element name="LastName" nillable="true">
            <simpleType>
              <restriction base="string">
                <maxLength value="30"/>
              </restriction>
            </simpleType>
          </element>
          <element name="FirstName" nillable="true">
            <simpleType>
              <restriction base="string">
                <maxLength value="10"/>
              </restriction>
            </simpleType>
          </element>
          <element name="Age" type="integer" nillable="true"/>
          <element name="Sex" type="string"/>
          <element name="Spouse">
            <complexType>
              <attribute name="sin" type="xsd:anyURI" use="required" />
            </complexType>
          </element>
          <element name="Location"
            type="gml:PointPropertyType">
```

```

        nillable="true"/>
        <element name="Address" type="myns:AddressType" nillable="true"/>
    </sequence>
    <attribute name="sin" type="xsd:anyURI" use="required"/>
</extension>
</complexContent>
</complexType>

<complexType name="AddressType">
    <sequence>
        <element name="StreetName" nillable="true">
            <simpleType>
                <restriction base="string">
                    <maxLength value="30"/>
                </restriction>
            </simpleType>
        </element>
        <element name="StreetNumber" nillable="true">
            <simpleType>
                <restriction base="string">
                    <maxLength value="10"/>
                </restriction>
            </simpleType>
        </element>
        <element name="City" nillable="true">
            <simpleType>
                <restriction base="string">
                    <maxLength value="30"/>
                </restriction>
            </simpleType>
        </element>
        <element name="Province" nillable="true">
            <simpleType>
                <restriction base="string">
                    <maxLength value="30"/>
                </restriction>
            </simpleType>
        </element>
        <element name="PostalCode" nillable="true">
            <simpleType>
                <restriction base="string">
                    <maxLength value="15"/>
                </restriction>
            </simpleType>
        </element>
    </sequence>

```

```

</element>
<element name="Country" nillable="true">
  <simpleType>
    <restriction base="string">
      <maxLength value="30"/>
    </restriction>
  </simpleType>
</element>
</sequence>
</complexType>
</schema>

```

Address 特性是一个复杂特性。

下例取回居住在"SomeTown" 镇"Main St."街区10000号并且收入在\$35,000以上的女性居民的名。注意如何在谓语使用Xpath表达式引用复杂特性。

```

<?xml version="1.0" ?>
<GetFeature
  version="1.0.0"
  service="WFS"
  xmlns="http://www.opengis.net/wfs"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:myns="http://www.someserver.com/myns"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-basic.xsd
    http://www.someserver.com/myns Person.xsd">
  <Query typeName="Person">
    <ogc:PropertyName>myns:Person/myns:LastName</ogc:PropertyName>
    <ogc:Filter>
      <ogc:And>
        <ogc:And>
          <ogc:PropertyIsGreaterThanOrEqualTo>
<ogc:PropertyName>myns:Person/myns:Address/myns:StreetNumber</ogc:PropertyName>
          <ogc:Literal>10000</ogc:Literal>
        </ogc:PropertyIsGreaterThanOrEqualTo>
        <ogc:PropertyIsLessThanOrEqualTo>
<ogc:PropertyName>myns:Person/myns:Address/myns:StreetNumber</ogc:PropertyName>
          <ogc:Literal>10999</ogc:Literal>
        </ogc:PropertyIsLessThanOrEqualTo>
      </ogc:And>
      <ogc:And>
        <ogc:PropertyIsEqualTo>
<ogc:PropertyName>myns:Person/myns:Address/myns:StreetName</ogc:PropertyName>
          <ogc:Literal>Main St.</ogc:Literal>

```

```

        </ogc:PropertyIsEqualTo>
        <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>myns:Person/myns:Address/myns:City</ogc:PropertyName>
        <ogc:Literal>SomeTown</ogc:Literal>
        </ogc:PropertyIsEqualTo>
        <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>myns:Person/myns:Sex</ogc:PropertyName>
        <ogc:Literal>Female</ogc:Literal>
        </ogc:PropertyIsEqualTo>
        <ogc:PropertyIsGreaterThan>
        <ogc:PropertyName>myns:Person/myns:Salary</ogc:PropertyName>
        <ogc:Literal>>35000</ogc:Literal>
        </ogc:PropertyIsGreaterThan>
    </ogc:And>
</ogc:And>
</ogc:Filter>
</Query>
</GetFeature>

```

8 LockFeature 操作

8.1 介绍

WEB连接本身是无状态的，因此，序列化事务的语义是不被保存的。为了理解它，我们观察修改操作。

一个客户端取回一个要素实例，接着这个要素在客户端被修改，并且通过Transaction请求提交修改返回数据库，由于没有任何事务可以保证要素在客户端被修改时，没有其他客户修改数据库中的同一要素，所以连续性丢失了。

保证连续性的一个方法是要求访问数据的操作是相互排斥的，即当一个事务访问一个数据项时，其他的事务不能修改这个数据项，这可以通过使用控制数据访问的锁来实现。

LockFeature操作的目的是使用长期要素锁机制来保证一致性，这个锁被认为是长期的，因为网络的延迟使要素锁要比本机的商用数据库中的锁长很多。

LockFeature操作是一个可选项，并不要求一个WFS实现依据本规范实现它，如果一个WFS实现了LockFeature操作，它必须在能力文档中声明。

8.2 请求

8.2.1 模式定义

由XML编码的LockFeature请求如下XML模式片段定义：

```

<xsd:element name="LockFeature" type="wfs:LockFeatureType"/>
<xsd:complexType name="LockFeatureType">
    <xsd:sequence>
        <xsd:element name="Lock" type="wfs:LockType" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="version"
        type="xsd:string" use="required" fixed="1.0.0"/>
    <xsd:attribute name="service"
        type="xsd:string" use="required" fixed="WFS"/>

```

```

<xsd:attribute name="expiry"
               type="xsd:positiveInteger" use="optional"/>
<xsd:attribute name="lockAction"
               type="wfs:AllSomeType" use="optional"/>
</xsd:complexType>
<xsd:complexType name="LockType">
  <xsd:sequence>
    <xsd:element ref="ogc:Filter" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="handle"
                 type="xsd:string" use="optional"/>
  <xsd:attribute name="typeName"
                 type="xsd:QName" use="required"/>
</xsd:complexType>

```

< LockFeature >元素包含一个或多个< Lock >元素，< Lock >元素定义了对一个要素类型的要素实例的锁操作。

expiry属性用来设定WEB要素服务（网络要素服务）对要素实例上锁的时限，用以在事务没有结果时释放该锁，**expiry**属性是用分钟来指定的，一旦指定的分钟数过去了，WEB要素服务（网络要素服务）器将释放这个要素上存在的锁。使用由服务生成的锁标识符所定的其他相背的进一步事务将会失败。本规范并不限制当时限没有被指定时一个锁应保持多长时间，然而对一个WEB要素服务（网络要素服务）来说包含上探测锁和释放锁的方法是谨慎的，这样可以释放一个持续很长一段时间没有被事务释放的锁。

< Lock >元素包含一个< Filter >元素，< Filter >元素被用来定义需要上锁的指定的要素类型的要素实例。使用< Filter >元素一个或多个要素实例可以通过它们的标识符被枚举出来，或者通过指定锁操作的空间或非空间限制来定义要素集合。< Filter >元素在Filter Encoding Implementation Specification [3]中定义。

可选属性**lockAction**是用来控制要素锁是如何被要求的，一个为ALL的锁动作表明WEB要素服务（网络要素服务）试着去获得所有被要求要素实例的锁，如果不是所有的要素实例都被锁住，这个操作就失败了，这些要素实例的锁将被释放。如果锁动作被设为SOME，那么要素服务将试着给请求的要素实例中可以上锁的实例上锁。默认的锁动作是ALL，10.2.2节介绍了LockFeature操作的状态机制。

8.2.2 使用 UML 表达的状态机

动态模型的方法通过UML参考手册描述，主要技术是状态机视图，状态图的UML注释的总结如图3所示。

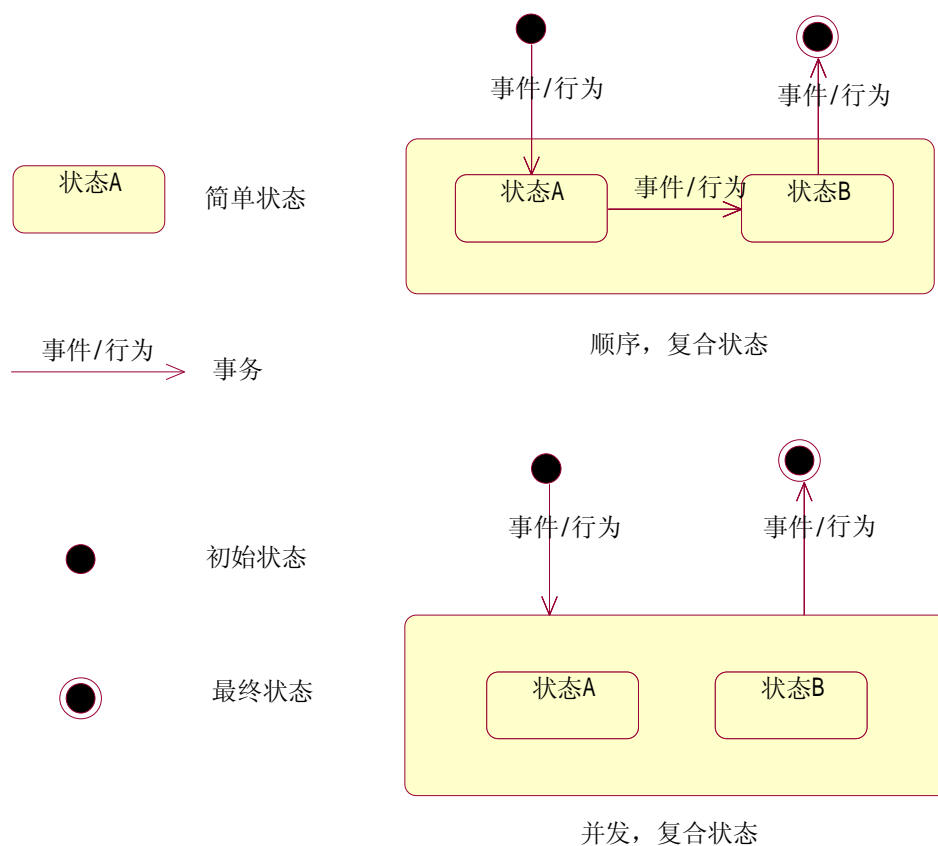


图3 状态图的 UML 注释的总结

8.2.3 WFS 锁的状态机

本节定义了提供WEB要素服务（网络要素服务）接口的服务器的锁状态的状态机，状态图显示了状态间允许的转变，所有其他的状态转变是不允许的，以出错的形式被服务器报告出来。

一个物理的服务器支持多个锁，从WFS规范中定义的服务的角度看，每个锁都是独立的。

在下面的状态模型中，转变是由请求触发的。接下来的消息模型中，WFS请求与WFS响应是成对的。注意当一个请求响应对是活动的则是不能开始，请求可能被HTTP--层的命令取消。

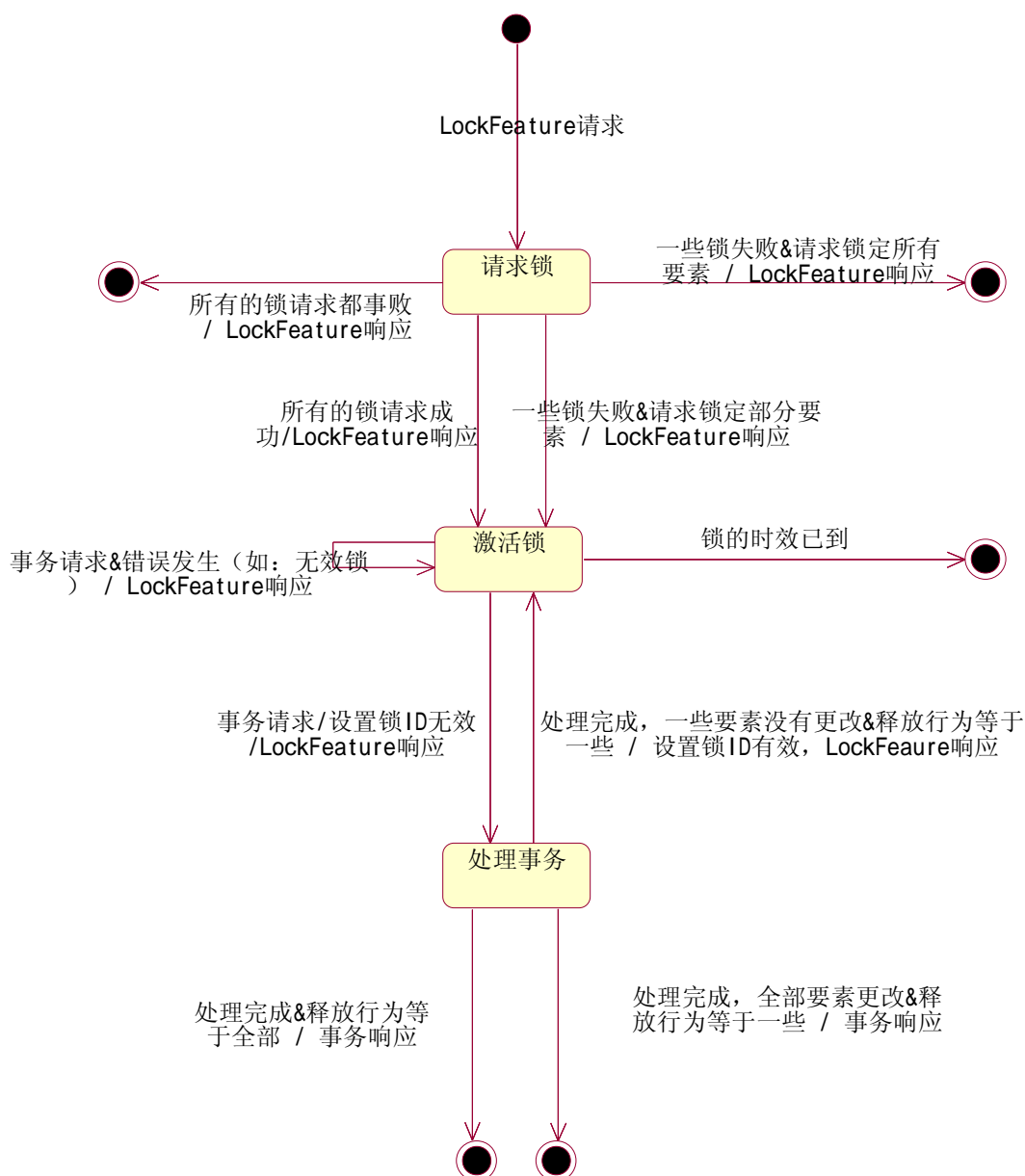


图4 WFS 锁状态图

8.3 响应

以XML编码的LockFeature请求的响应由下面的XML模式片段定义。

```

<xsd:element name="WFS_LockFeatureResponse"
  type="wfs:WFS_LockFeatureResponseType"/>
<!-- RESPONSE TYPES -->
<xsd:complexType name="WFS_LockFeatureResponseType">
  <xsd:sequence>
    <xsd:element ref="wfs:LockId"/>
    <xsd:element name="FeaturesLocked"
      type="wfs:FeaturesLockedType" minOccurs="0"/>
    <xsd:element name="FeaturesNotLocked"
      type="wfs:FeaturesNotLockedType" minOccurs="0"/>
  
```

```

</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="FeaturesLockedType">
  <xsd:sequence maxOccurs="unbounded">
    <xsd:element ref="ogc:FeatureId"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="FeaturesNotLockedType">
  <xsd:sequence maxOccurs="unbounded">
    <xsd:element ref="ogc:FeatureId"/>
  </xsd:sequence>
</xsd:complexType>

```

为了响应LockFeature请求，一个WEB要素服务（网络要素服务）应生成一个XML文档，这个文档应包含一个锁标识符，客户端应用可以在后继的WFS操作中使用这个锁标识符来操作被锁的要素实例集，响应中可以包含可选的元素< FeaturesLocked >和< FeaturesNotLocked >元素，这取决于lockAction属性值。

如果锁动作被指定为SOME，那么< WFS_LockFeatureResponse >元素必须包含< FeaturesLocked >和< FeatureNotLocked >元素。< FeaturesLocked >元素应列出被LockFeature请求上锁的所有要素实例的要素标识符，< FeatureNotLocked >元素应包含不能被WEB要素服务（网络要素服务）上锁的要素实例的要素标识符列表（可能是因为他们已经被其他人上锁）。

没有对锁标识符的格式作任何假定，唯一的要求是它可以用事务请求的字符集表达。

8.4 异常

如果一个WFS没有完成LockFeature操作，它应产生一个异常，表示这个操作不被支持，如果遇到这样一个请求。

在WEB要素服务（网络要素服务）不支持LockFeature操作以及在为请求提供服务时遇到错误时，应该如7.7节所述产生一个异常。

8.5 示例

例1

为枚举的要素上锁，在这种情况下，WFS直接尝试为它能上锁的所有要素上锁。

请求：

```

<?xml version="1.0" ?>
<LockFeature
  version="1.0.0"
  service="WFS"
  lockAction="SOME"
  xmlns="http://www.opengis.net/wfs"
  xmlns:myns="http://www.someserver.com/myns"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-transaction.xsd">
  <Lock typeName="myns:INWATERA_1M">
    <ogc:Filter>
      <ogc:FeatureId fid="INWATERA_1M.1013"/>
    </ogc:Filter>
  </Lock>

```

```

    <ogc:FeatureId fid="INWATERA_1M.1014"/>
    <ogc:FeatureId fid="INWATERA_1M.1015"/>
    <ogc:FeatureId fid="INWATERA_1M.1016"/>
    <ogc:FeatureId fid="INWATERA_1M.1017"/>
  </ogc:Filter>
</Lock>
</LockFeature>

```

响应示例:

```

<?xml version="1.0" ?>
<WFS_LockFeatureResponse
  xmlns="http://www.opengis.net/wfs"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-transaction.xsd">
  <LockId>1</LockId>
  <FeaturesLocked>
    <ogc:FeatureId fid="INWATERA_1M.1013"/>
    <ogc:FeatureId fid="INWATERA_1M.1014"/>
    <ogc:FeatureId fid="INWATERA_1M.1016"/>
    <ogc:FeatureId fid="INWATERA_1M.1017"/>
  </FeaturesLocked>
  <FeaturesNotLocked>
    <ogc:FeatureId fid="INWATERA_1M.1015"/>
  </FeaturesNotLocked>
</WFS_LockFeatureResponse>

```

例2

为所有INMATERA_IM类型的要素实例上锁。

Request:

```

<?xml version="1.0" ?>
<LockFeature
  version="1.0.0"
  service="WFS"
  xmlns="http://www.opengis.net/wfs"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:myns="http://www.someserver.com/myns"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-transaction.xsd">
  <Lock typeName="myns:INWATERA_1M"/>
</LockFeature>

```

响应示例:

```

<?xml version="1.0" ?>

```

```

<WFS_LockFeatureResponse
  xmlns="http://www.opengis.net/wfs"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-transaction.xsd">
  <LockId>2</LockId>
</WFS_LockFeatureResponse>

```

例3

在本例使用有空间限制条件中，< Filter >表达式被用来定义上锁的要素实例集合。

请求：

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

```

<LockFeature
  version="1.0.0"
  service="WFS"
  xmlns="http://www.opengis.net/wfs"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:myns="http://www.someserver.com/myns"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-transaction.xsd">
  <Lock handle="Lock1" typeName="myns:INWATERA_1M">
    <ogc:Filter>
      <ogc:Within>
        <ogc:PropertyName>myns:WKB_GEOM</ogc:PropertyName>
        <gml:Polygon gid="1"
          srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
          <gml:outerBoundaryIs>
            <gml:LinearRing>
              <gml:coordinates>-95.7,38.1 -97.8,38.2 ...</gml:coordinates>
            </gml:LinearRing>
          </gml:outerBoundaryIs>
        </gml:Polygon>
      </ogc:Within>
    </ogc:Filter>
  </Lock>
</LockFeature>

```

响应示例：

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

```

<WFS_LockFeatureResponse
  xmlns="http://www.opengis.net/wfs"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-transaction.xsd">
  <LockId>A1014375BD</LockId>

```

```
</WFS_LockFeatureResponse>
```

例4

本例为在指定窗口内的所有BUILTUPA_1M和INWATERA_1M类型的要素上锁, 这个锁使用handle标记为LOCK1。使用handle标记为LOCK2的锁为要素INWATERA_1M.1212, INWATERA_1M.1213和INWATERA_1M.10上锁。

Request:

```
<LockFeature
  version="1.0.0"
  service="WFS"
  expiry="4"
  lockAction="SOME"
  xmlns="http://www.opengis.net/wfs"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:myns="http://www.someserver.com/myns"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-transaction.xsd">
  <Lock handle="LOCK1" typeName="myns:BUILTUPA_1M">
    <ogc:Filter>
      <ogc:Within>
        <ogc:PropertyName>BUILTUPA_1M/WKB_GEOM</ogc:PropertyName>
        <gml:Polygon gid="1"
          srsName="http://www.opengis.net/gml/epsg.xml#4326">
          <gml:outerBoundaryIs>
            <gml:LinearRing>
              <gml:coordinates>-95.7,38.1 -97.8,38.2 ...</gml:coordinates>
            </gml:LinearRing>
          </gml:outerBoundaryIs>
        </gml:Polygon>
      </ogc:Within>
    </ogc:Filter>
  </Lock>
  <Lock handle="LOCK2" typeName="myns:INWATERA_1M">
    <ogc:Filter>
      <ogc:FeatureId fid="INWATERA_1M.1212"/>
      <ogc:FeatureId fid="INWATERA_1M.1213"/>
      <ogc:FeatureId fid="INWATERA_1M.10"/>
    </ogc:Filter>
  </Lock>
</LockFeature>
```

响应示例:

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

```

<WFS_LockFeatureResponse
  xmlns="http://www.opengis.net/wfs"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-transaction.xsd">
  <LockId>LOCK1A</LockId>
  <FeaturesLocked>
    <ogc:FeatureId fid="BUILTUPA_1M.1" />
    <ogc:FeatureId fid="BUILTUPA_1M.10" />
    <ogc:FeatureId fid="BUILTUPA_1M.34" />
    <ogc:FeatureId fid="BUILTUPA_1M.786" />
    <ogc:FeatureId fid="BUILTUPA_1M.3" />
    <ogc:FeatureId fid="BUILTUPA_1M.13" />
    <ogc:FeatureId fid="BUILTUPA_1M.47563" />
    <ogc:FeatureId fid="INWATERA_1M.1212" />
    <ogc:FeatureId fid="INWATERA_1M.1213" />
    <ogc:FeatureId fid="INWATERA_1M.10" />
  </FeaturesLocked>
</WFS_LockFeatureResponse>

```

9 Transaction 操作

9.1 介绍

Transaction操作用来描述施加到可以通过WEB访问的要素实例上的操作，WEB要素服务（网络要素服务）可以直接处理一个Transaction操作，也可以转换成它所联结的数据存储的目标语言，然后让数据存储执行这个事务。当事务执行完毕，WEB要素服务（网络要素服务）会生成XML报告文档，指明事务完成状态。

Transaction操作是可选的，WFS实现不需要支持它与本规范保持一致。如果Transaction操作被支持，那么一定要在能力文档中声明这一事实，能力文档在第12节中描述。

9.2 请求

9.2.1 模式定义

用XML编码的Transaction请求由如下XML模式片段定义：

```

<xsd:element name="Transaction" type="wfs:TransactionType"/>
<xsd:complexType name="TransactionType">
  <xsd:sequence>
    <xsd:element ref="wfs:LockId" minOccurs="0"/>
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
      <xsd:element ref="wfs:Insert"/>
      <xsd:element ref="wfs:Update"/>
      <xsd:element ref="wfs:Delete"/>
      <xsd:element ref="wfs:Native"/>
    </xsd:choice>
  </xsd:sequence>
  <xsd:attribute name="version"
    type="xsd:string" use="required" fixed="1.0.0"/>

```

```

<xsd:attribute name="service"
                type="xsd:string" use="required" fixed="WFS"/>
<xsd:attribute name="handle"
                type="xsd:string" use="optional"/>
<xsd:attribute name="releaseAction"
                type="wfs:AllSomeType" use="optional"/>
</xsd:complexType>
<xsd:element name="LockId" type="xsd:string"/>
<xsd:element name="Insert" type="wfs:InsertElementType"/>
<xsd:complexType name="InsertElementType">
  <xsd:sequence>
    <xsd:element ref="gml:_Feature" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="handle" type="xsd:string" use="optional"/>
</xsd:complexType>
<xsd:element name="Update" type="wfs:UpdateElementType"/>
<xsd:complexType name="UpdateElementType">
  <xsd:sequence>
    <xsd:element ref="wfs:Property" maxOccurs="unbounded"/>
    <xsd:element ref="ogc:Filter" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="handle" type="xsd:string" use="optional"/>
  <xsd:attribute name="typeName" type="xsd:QName" use="required"/>
</xsd:complexType>
<xsd:element name="Delete" type="wfs:DeleteElementType"/>
<xsd:complexType name="DeleteElementType">
  <xsd:sequence>
    <xsd:element ref="ogc:Filter" minOccurs="1" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="handle" type="xsd:string" use="optional"/>
  <xsd:attribute name="typeName" type="xsd:QName" use="required"/>
</xsd:complexType>

<xsd:element name="Property" type="wfs:PropertyType"/>
<xsd:complexType name="PropertyType">
  <xsd:sequence>
    <xsd:element name="Name" type="xsd:QName"/>
    <xsd:element name="Value"/>
  </xsd:sequence>
</xsd:complexType>

```

9.2.2 属性描述

如7、8节所述，**handle**属性被用来给一个元素分配一个助记名，它的使用，是使错误报告对客户端应用更有意义。当错误发生时，**handle**属性被WEB要素服务（网络要素服务）用来在生成异常报告时定

位错误,如果web没有被指定,WEB要素服务(网络要素服务)会使用行号或其它方便的机制报告与当前Transaction相关的异常定位。

假设一个WFS实现支持可选的LockFeature和/或GetFeatureWithLock操作, releaseAction属性是用来控制当一个事务请求完成后如何处理上锁的要素,值ALL表明使用指定< LockId >加在所有要素实例上的锁在事务完成时被释放的,而不管被锁的要素集中的特定要素实例是否真正被操作过。值SOME表明只有被事务改动过的那些上锁的要素实例被释放,其它没被改动过的要素实例应该继续使用相同的< LockId >保持上锁状态,这样后继的事务可以操作这些要素实例,当releaseAction被设为SOME,并且过期时限由< LockFeature >或< GetFeatureWithLock >元素使用expiry属性指定,当每个事务结束后,过期时限计数器必须被置0,除非所有上锁的要素实例都被操作过。releaseAction默认值是ALL。

例如,如果一个客户应用锁住20个要素实例,接着提交了一个事务请求,只处理了其中10个要素实例,如果releaseAction为SOME,意味着当事务终止后,这10个没被改动的要素实例将继续保持上锁状态,由客户端应用提交的后继事务操作可以使用相同的锁符标识符修改剩余的这10个要素实例。

9.2.3 <Transaction>元素

< Transaction >元素可以包含0或多个< Insert >、< Update >或< Delete >元素,这些元素用来描述创建,修改或销毁要素实例的操作,空< Transaction >请求是合法的,但没什么用。

可选元素< LockId >用来指定事务可以使用前面被上锁的要素实例,第10节详细描述了一个要素上锁的机制,如果WFS不支持要素锁,< LockId >将被忽略,如果WFS支持锁,并且在事务中指定了不合法的锁描述符,那么事务将会失败,并且WEB要素会报告错误,如7.7所述。

在事务的最后,WEB要素服务(网络要素服务)会应用恰当的事务语义到持久保存要素的特定系统中,例如,如果数据存储是基于SQL的RDBMS,那么commit将在事务的最后被执行(或当事务失败时rollback)。在事务处理期间由WEB要素服务(网络要素服务)保持的任何锁应根据releaseAction的值被释放。

< Native >元素在7.5节中定义。

9.2.4 <Insert>元素

< Insert >元素被用来创建新的要素实例,被创建要素实例的初始状态用GML表达,并且必须通过由DescribeFeatureType操作生成的GML应用模式有效性验证,在一个单一的事务请求中可以包含多个< Insert >元素,并且多个要素实例可以由一个单一的< Insert >元素创建。

在响应< Insert >操作时,WEB要素服务(网络要素服务)应生成分配给新要素实例的新生成的要素标识符,要素标识符必须按Transaction请求中< Insert >操作顺序排列。

例:

下面的事务创建2个要素类型为INWATERA_1M的要素实例。

```
<?xml version="1.0"?>
<wfs:Transaction
  version="1.0.0"
  service="WFS"
  xmlns="http://www.someserver.com/myns"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.someserver.com/myns
    http://www.someserver.com/wfs/cwwfs.cgi?
    request=describefeaturetype&typename=INWATERA_1M.xsd
```

```

http://www.opengis.net/wfs ../wfs/1.0.0/WFS-transaction.xsd">
<wfs:Insert>
  <INWATERA_1M>
    <WKB_GEOM>
      <gml:Polygon gid="1"
        srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
        <gml:outerBoundaryIs>
          <gml:LinearRing>
            <gml:coordinates>-98.54,24.26 ...</gml:coordinates>
          </gml:LinearRing>
        </gml:outerBoundaryIs>
      </gml:Polygon>
    </WKB_GEOM>
    <ID>150</ID>
    <F_CODE>ABCDE</F_CODE>
    <HYC>152</HYC>
    <TILE_ID>250</TILE_ID>
    <FAC_ID>111</FAC_ID>
  </INWATERA_1M>
  <INWATERA_1M>
    <WKB_GEOM>
      <gml:Polygon gid="1"
        srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
        <gml:outerBoundaryIs>
          <gml:LinearRing>
            <gml:coordinates>-99.99,22.22 ...</gml:coordinates>
          </gml:LinearRing>
        </gml:outerBoundaryIs>
      </gml:Polygon>
    </WKB_GEOM>
    <ID>111</ID>
    <F_CODE>FGHIJ</F_CODE>
    <HYC>222</HYC>
    <TILE_ID>333</TILE_ID>
    <FAC_ID>444</FAC_ID>
  </INWATERA_1M>
</wfs:Insert>
</wfs:Transaction>

```

引用的模式 INWATERA_1M.xsd假定是由DescribeFeatureType操作生成的，在这个例子中，这个模式文档被静态地引用，不过也可以同样简单地被动态引用。

9.2.5 <Update>元素

< Update >元素描述一个被施加到一个要素或一个单一类型要素集合的修改操作，一个 Transaction请求中可以包含多个< Update >操作。

< Update >元素包含一个或多个< Property >元素，它指定属于某一使用强制typeName属性指定的要素类型的特性的名和替换值，< Property >元素包含一个< Name >元素，相应地，包含将被修改的要素特性名和可选的< Value >元素，< Value >元素包含指定要素特性的替换值，缺省< Value >元素意味着这个特性被赋NULL值，当特性值不能为空时，WFS必须产生一个异常，表明NULL值是不允许的。

< Update >元素的范围由< Filter >元素限定，< Filter >元素可以被用来限定修改操作的范围，枚举出要素集或使用空间或非空间限制条件定义的要素集。

<FILTER>元素在FILTER ENCODING IMPEMETATION SPECIFITION中定义。

示例：

下例修改了要素标识符为UILTUPA_1M.1013.的要素的POPULATION特性。

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

```
<wfs:Transaction
```

```
  version="1.0.0"
```

```
  service="WFS"
```

```
  xmlns="http://www.someserver.com/myns"
```

```
  xmlns:ogc="http://www.opengis.net/ogc"
```

```
  xmlns:wfs="http://www.opengis.net/wfs"
```

```
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
  xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-transaction.xsd">
```

```
  <wfs:Update typeName="BUILTUPA_1M">
```

```
    <wfs:Property>
```

```
      <wfs:Name>POPULATION</wfs:Name>
```

```
      <wfs:Value>4070000</wfs:Value>
```

```
    </wfs:Property>
```

```
    <ogc:Filter>
```

```
      <ogc:FeatureId fid="BUILTUPA_1M.10131"/>
```

```
    </ogc:Filter>
```

```
  </wfs:Update>
```

```
</wfs:Transaction>
```

示例

修改一个枚举要素集的POPULATION_TYPE，在本例中，要素由要素标识符指定：

BUILTUPA_1M.1013

BUILTUPA_1M.34

BUILTUPA_1M.24256

它们的POPULATION_TYPE属性值设为“CITY”。

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

```
<wfs:Transaction
```

```
  version="1.0.0"
```

```
  service="WFS"
```

```
  xmlns="http://www.someserver.com/myns"
```

```
  xmlns:ogc="http://www.opengis.net/ogc"
```

```
  xmlns:wfs="http://www.opengis.net/wfs"
```

```
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```

xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-transaction.xsd">
<wfs:Update typeName="BUILTUPA_1M">
  <wfs:Property>
    <wfs:Name>POPULATION_TYPE</wfs:Name>
    <wfs:Value>CITY</wfs:Value>
  </wfs:Property>
  <ogc:Filter>
    <ogc:FeatureId fid="BUILTUPA_1M.1013"/>
    <ogc:FeatureId fid="BUILTUPA_1M.34"/>
    <ogc:FeatureId fid="BUILTUPA_1M.24256"/>
  </ogc:Filter>
</wfs:Update>
</wfs:Transaction>

```

示例

修改枚举要素集的NAME特性，并且修改由限制条件‘FAC_ID特性值大于1000’，限制的要素集的FAC_ID特性。

```

<?xml version="1.0" ?>
<wfs:Transaction
  version="1.0.0"
  service="WFS"
  xmlns:myns="http://www.someserver.com/myns"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-transaction.xsd">
  <wfs:Update typeName="myns:BUILTUPA_1M">
    <wfs:Property>
      <wfs:Name>myns:NAME</wfs:Name>
      <wfs:Value>somestring</wfs:Value>
    </wfs:Property>
    <ogc:Filter>
      <ogc:FeatureId fid="BUILTUPA_1M.1013"/>
      <ogc:FeatureId fid="BUILTUPA_1M.34"/>
      <ogc:FeatureId fid="BUILTUPA_1M.24256"/>
    </ogc:Filter>
  </wfs:Update>
  <wfs:Update typeName="myns:BUILTUPA_1M">
    <wfs:Property>
      <wfs:Name>myns:FAC_ID</wfs:Name>
      <wfs:Value>100</wfs:Value>
    </wfs:Property>
    <ogc:Filter>

```

```

    <ogc:PropertyIsGreaterThan>
      <ogc:PropertyName>BUILTUPA_1M/TILE_ID</ogc:PropertyName>
      <ogc:Literal>1000</ogc:Literal>
    </ogc:PropertyIsGreaterThan>
  </ogc:Filter>
</wfs:Update>
</wfs:Transaction>

```

示例

本例修改了两个要素类，OCEANSA_1M和TREESA_1M。所有深于2400M的OCEANSA_1M要素以及要素TREESA_1M1010被修改。

```

<?xml version="1.0" ?>
<wfs:Transaction
  version="1.0.0"
  service="WFS"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:myns="http://www.someserver.com/myns"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-transaction.xsd">
  <wfs:Update typeName="myns:OCEANSA_1M">
    <wfs:Property>
      <wfs:Name>myns:DEPTH</wfs:Name>
      <wfs:Value>2400</wfs:Value>
    </wfs:Property>
    <ogc:Filter>
      <ogc:PropertyIsGreaterThan>
        <ogc:PropertyName>OCEANSA_1M.DEPTH</ogc:PropertyName>
        <ogc:Literal>2400</ogc:Literal>
      </ogc:PropertyIsGreaterThan>
    </ogc:Filter>
  </wfs:Update>
  <wfs:Update typeName="myns:TREESA_1M">
    <wfs:Property>
      <wfs:Name>myns:TREETYPE</wfs:Name>
      <wfs:Value>CONIFEROUS</wfs:Value>
    </wfs:Property>
    <ogc:Filter>
      <ogc:FeatureId fid="TREESA_1M.1010"/>
    </ogc:Filter>
  </wfs:Update>
</wfs:Transaction>

```

```
</wfs:Update>
```

```
</wfs:Transaction>
```

9.2.6 <Delete>元素

〈Delete〉元素用来表明一个或多个要素实例将被删除，删除操作的范围用〈Filter〉元素限定，〈Filter〉元素在Filter中描述

示例：

删除一个要素。

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

```
<wfs:Transaction
```

```
  version="1.0.0"
```

```
  service="WFS"
```

```
  xmlns="http://www.someserver.com/myns"
```

```
  xmlns:ogc="http://www.opengis.net/ogc"
```

```
  xmlns:wfs="http://www.opengis.net/wfs"
```

```
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
  xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-transaction.xsd">
```

```
  <wfs:Delete typeName="INWATERA_1M">
```

```
    <ogc:Filter>
```

```
      <ogc:FeatureId fid="INWATERA_1M.1013"/>
```

```
    </ogc:Filter>
```

```
  </wfs:Delete>
```

```
</wfs:Transaction>
```

示例

本例删除一个枚举要素实例集。

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

```
<wfs:Transaction
```

```
  version="1.0.0"
```

```
  service="WFS"
```

```
  xmlns:myns="http://www.someserver.com/myns"
```

```
  xmlns:ogc="http://www.opengis.net/ogc"
```

```
  xmlns:wfs="http://www.opengis.net/wfs"
```

```
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
  xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-transaction.xsd">
```

```
  <wfs:Delete typeName="myns:INWATERA_1M">
```

```
    <ogc:Filter>
```

```
      <ogc:FeatureId fid="INWATERA_1M.1013"/>
```

```
      <ogc:FeatureId fid="INWATERA_1M.10"/>
```

```
      <ogc:FeatureId fid="INWATERA_1M.13"/>
```

```
      <ogc:FeatureId fid="INWATERA_1M.140"/>
```

```
      <ogc:FeatureId fid="INWATERA_1M.5001"/>
```

```
      <ogc:FeatureId fid="INWATERA_1M.2001"/>
```

```

    </ogc:Filter>
  </wfs:Delete>
</wfs:Transaction>

```

示例

本例表明删除在由面定义的区域内的 INWATERA_1M 类型的要素集。<Filter> 用来限制操作的类型，GML 用来表达面的几何信息。

```

<?xml version="1.0" ?>
<wfs:Transaction
  version="1.0.0"
  service="WFS"
  xmlns="http://www.someserver.com/myns"
  xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-transaction.xsd">
  <wfs:Delete typeName="INWATERA_1M">
    <ogc:Filter>
      <ogc:Within>
        <ogc:PropertyName>WKB_GEOM</ogc:PropertyName>
        <gml:Polygon gid="pp9"
          srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
          <gml:outerBoundaryIs>
            <gml:LinearRing>
              <gml:coordinates>-95.7,38.1 -97.8,38.2 ...</gml:coordinates>
            </gml:LinearRing>
          </gml:outerBoundaryIs>
        </gml:Polygon>
      </ogc:Within>
    </ogc:Filter>
  </wfs:Delete>
</wfs:Transaction>

```

9.3 响应

在响应事务请求时，WEB要素服务（网络要素服务）应生成XML文档指明事务的最终状态。另外，如果事务请求包含〈Insert〉操作，WEB要素服务（网络要素服务）必须报告所有新创建的要素标识符。当事务失败时，WEB要素服务（网络要素服务）也应在响应中声明。

用XML编码的WFS事务响应应由以下XML模式片段定义：

```

<xsd:element name="WFS_TransactionResponse"
  type="wfs:WFS_TransactionResponseType"/>
<xsd:complexType name="WFS_TransactionResponseType">
  <xsd:sequence>
    <xsd:element name="InsertResult"
      type="wfs:InsertResultType"

```

```

        minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="TransactionResult"
        type="wfs:TransactionResultType"/>
    </xsd:sequence>
    <xsd:attribute name="version"
      type="xsd:string" use="required" fixed="1.0.0"/>
  </xsd:complexType>
  <xsd:complexType name="TransactionResultType">
    <xsd:sequence>
      <xsd:element name="Status" type="wfs:StatusType"/>
      <xsd:element name="Locator" type="xsd:string" minOccurs="0"/>
      <xsd:element name="Message" type="xsd:string" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="handle" type="xsd:string" use="optional"/>
  </xsd:complexType>
  <xsd:complexType name="InsertResultType">
    <xsd:sequence>
      <xsd:element ref="ogc:FeatureId" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="handle" type="xsd:string" use="optional"/>
  </xsd:complexType>
  <xsd:complexType name="StatusType">
    <xsd:choice>
      <xsd:element ref="wfs:SUCCESS"/>
      <xsd:element ref="wfs:FAILED"/>
      <xsd:element ref="wfs:PARTIAL"/>
    </xsd:choice>
  </xsd:complexType>
  <xsd:element name="SUCCESS" type="wfs:EmptyType"/>
  <xsd:element name="FAILED" type="wfs:EmptyType"/>
  <xsd:element name="PARTIAL" type="wfs:EmptyType"/>

```

〈WFS_TransactionResponse〉元素包含 0 个或多个〈InsertResult〉元素和一个〈TransactionResult〉元素。

〈InsertResult〉元素包含一个或多个新创建的要素实例的要素标识符，一个〈InsertResult〉元素报告请求中的一个〈Insert〉元素的结果，〈Insert〉结果的报告顺序是依照〈Transaction〉元素中〈Insert〉的操作顺序。另外，如果handle属性被指定了，它们可以相互关联。

事务请求的所有结果使用〈TransactionResult〉元素指定，〈TransactionResult〉元素必须包含< Status >元素，可以包含< Locator >元素和〈Message〉元素。

<Status>元素用来表示事务完成的状态，事务有以下结束状态：

SUCCESS	事务完全完成;
FAILED	当事务处理请求中的一个或多个元素时发生异常;
PARTIAL	事务部分成功, 数据保持一致性状态, 如果系统不支持原子事务, 这个输出是很可能的。

当事务请求失败, < Locator >元素可以用来指出哪部分事务失败了, 如果失败的那个元素使用了 handle 属性标记, WEB 要素服务 (网络要素服务) 报告它的值用以定位失败, 否则, 要素服务会从事务服务的开始来定位失败, 使用行号或其它方便的机制。

〈Message〉元素被用来报告错误信息

示例

考虑一个事务请求 (标记为 "TX01") 它创建了一些新的要素实例, 要素实例是使用 〈Insert〉元素创建的, 标记为 "STMT1"、"STMT2" 和 "STMT3", 一个典型的对上述请求的响应可能是:

```
<?xml version="1.0" ?>
<wfs:TransactionResponse
  version="1.0.0"
  xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-transaction.xsd">
  <wfs:InsertResult handle="STMT1">
    <ogc:FeatureId fid="SOMEFEATURE.4567"/>
    <ogc:FeatureId fid="SOMEFEATURE.4568"/>
    <ogc:FeatureId fid="SOMEFEATURE.4569"/>
  </wfs:InsertResult>
  <wfs:InsertResult handle="STMT2">
    <ogc:FeatureId fid="FEATURE1.4569"/>
  </wfs:InsertResult>
  <wfs:InsertResult handle="STMT3">
    <ogc:FeatureId fid="FEATURE2.389345"/>
  </wfs:InsertResult>
  <wfs:TransactionResult handle="TX01">
    <wfs:Status>
      <wfs:SUCCESS/>
    </wfs:Status>
  </wfs:TransactionResult>
</wfs:TransactionResponse>
```

9.4 异常

当WEB要素服务 (网络要素服务) 在为Transaction请求服务时遇到错误, 它将产生一个异常, 异常在7.7节描述。

当WEB要素服务 (网络要素服务) 在处理一个Transaction请求中特定元素时遇到错误, WEB要素服务 (网络要素服务) 将排队等候结果, 并在<WFS_TransactionResponse> 元素中报告失败。

示例:

本例中，Transaction请求中第二个元素失败了，WFS会生成如下响应

```
<?xml version="1.0" ?>
<WFS_TransactionResponse
  version="1.0.0"
  xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-transaction.xsd">

  <TransactionResult handle="TX01">
    <Status><FAILED/></Status>
    <Locator>STMT2</Locator>
    <Message>ORA-00942: table or view does not exist</Message>
  </TransactionResult>
</WFS_TransactionResponse>
```

9.5 示例

本例定义了一个复杂的事务，标记为"Transaction 01"，它可以执行插入、修改和删除操作，一些要素类型包含复杂特性和Xpath表达式，它们使用过滤表达式明确需要引用的特性，例子中的注释解释了各种操作。

```
<?xml version="1.0" ?>
<wfs:Transaction
  version="1.0.0"
  service="WFS"
  handle="Transaction 01"
  xmlns="http://www.someserver.com/myns"
  xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.someserver.com/myns
    http://www.someserver.com/wfs/cwwfs.cgi?
    request=DESCRIBEFEATURETYPE&
    typename=ELEVP_1M,ROADL_1M,BUILTUPA_1M
    http://www.opengis.net/wfs ../wfs/1.0.0/WFS-transaction.xsd">

  <!-- Create a new instance of feature type ELEVP_1M -->
  <wfs:Insert handle="Statement 1">
    <ELEVP_1M>
      <ID>167928</ID>
      <F_CODE>CA</F_CODE>
      <ACC>2</ACC>
      <ELA>1</ELA>
      <ZV2>152</ZV2>
      <TILE_ID>250</TILE_ID>
```

```

        <END_ID>111</END_ID>
        <LOCATION>
            <gml:Point gid="e33"
                srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
                <gml:coordinates>-98.5485,24.2633</gml:coordinates>
            </gml:Point>
        </LOCATION>
    </ELEV_1M>
</wfs:Insert>

<!-- Create a new instance of feature type ROADL_1M
      which has complex properties SEGMENT and ROADTYPE. -->
<wfs:Insert handle="ComplexInsert">
    <ROADL_1M>
        <NAME>Highway 401</NAME>
        <SEGMENT>
            <DESIGNATION>SEG_A41</DESIGNATION>
            <GEOMETRY>
                <gml:LineString gid="e3"
                    srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
                    <gml:coordinates>...</gml:coordinates>
                </gml:LineString>
            </GEOMETRY>
        </SEGMENT>
        <ROADTYPE>
            <SURFACE_TYPE>Asphalt</SURFACE_TYPE>
            <NLANS>12</NLANS>
            <GRADE>15</GRADE>
        </ROADTYPE>
    </ROADL_1M>
</wfs:Insert>

<!-- Update the designation of a particular range of segments
      which are now being collapsed into a single segment. The
      The filter uses an XPath expression to reference the
      DESIGNATION property -->
<wfs:Update typeName="ROADL_1M">
    <wfs:Property>
        <wfs:Name>ROADL_1M/SEGMENT/DESIGNATION</wfs:Name>
        <wfs:Value>SEG_A60</wfs:Value>
    </wfs:Property>
    <ogc:Filter>
        <ogc:PropertyIsBetween>
            <ogc:PropertyName>ROADL_1M/SEGMENT/DESIGNATION</ogc:PropertyName>

```

```

    <ogc:LowerBoundary>
      <ogc:Literal>SEG_A60</ogc:Literal>
    </ogc:LowerBoundary>
    <ogc:UpperBoundary>
      <ogc:Literal>SEG_A69</ogc:Literal>
    </ogc:UpperBoundary>
  </ogc:PropertyIsBetween>
</ogc:Filter>
</wfs:Update>

<!-- Create 2 feature instances of feature type BUILTUPA_1M -->
<wfs:Insert handle="Statement 2">
  <BUILTUPA_1M>
    <PLACEID>4070</PLACEID>
    <NAME>Toronto</NAME>
    <POPULATION>4000000</POPULATION>
    <BNDRY>
      <gml:Polygon gid="g3"
        srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
        <gml:outerBoundaryIs>
          <gml:LinearRing>
            <gml:coordinates>-95.7,38.1 -97.8,-38.2 ...</gml:coordinates>
          </gml:LinearRing>
        </gml:outerBoundaryIs>
      </gml:Polygon>
    </BNDRY>
  </BUILTUPA_1M>
  <BUILTUPA_1M>
    <PLACEID>1725</PLACEID>
    <NAME>Montreal</NAME>
    <POPULATION>2000000</POPULATION>
    <BNDRY>
      <gml:Polygon gid="e4"
        srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
        <gml:outerBoundaryIs>
          <gml:LinearRing>
            <gml:coordinates>-89.8,44.3 -89.9,44.4 ...</gml:coordinates>
          </gml:LinearRing>
        </gml:outerBoundaryIs>
      </gml:Polygon>
    </BNDRY>
  </BUILTUPA_1M>
</wfs:Insert>

```

```

<!-- Update the NAME property of the feature instance BUILTUPA_1M.1210 -->
<wfs:Update typeName="BUILTUPA_1M">
  <wfs:Property>
    <wfs:Name>NAME</wfs:Name>
    <wfs:Value>SMALLVILLE</wfs:Value>
  </wfs:Property>
  <ogc:Filter>
    <ogc:FeatureId fid="BUILTUPA_1M.1210"/>
  </ogc:Filter>
</wfs:Update>

<!-- Update the geometry of the feature BUILTUPA_1M.1725. -->
<wfs:Update typeName="BUILTUPA_1M">
  <wfs:Property>
    <wfs:Name>BNDRY</wfs:Name>
    <wfs:Value>
      <gml:Polygon gid="g5"
        srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
        <gml:outerBoundaryIs>
          <gml:LinearRing>
            <gml:coordinates>-89.8,44.3 -89.9,44.4 ...</gml:coordinates>
          </gml:LinearRing>
        </gml:outerBoundaryIs>
      </gml:Polygon>
    </wfs:Value>
  </wfs:Property>
  <ogc:Filter>
    <ogc:FeatureId fid="BUILTUPA_1M.1725"/>
  </ogc:Filter>
</wfs:Update>

<!-- Delete the feature instance BUILTUPA_1M.1013. -->
<wfs:Delete typeName="BUILTUPA_1M">
  <ogc:Filter>
    <ogc:FeatureId fid="BUILTUPA_1M.1013"/>
  </ogc:Filter>
</wfs:Delete>

<!-- Delete all instances of the feature type
      BUILTUPA_1M where:
      1. the geometry is INSIDE a polygonal window
      2. where the POPULATION is between 100 and 2000 -->
<wfs:Delete typeName="BUILTUPA_1M">
  <ogc:Filter>

```

```

<ogc:And>
  <ogc:Within>
    <ogc:PropertyName>BUILTUPA_1M/BNDRY</ogc:PropertyName>
    <gml:Polygon gid="WINDOW"
      srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
      <gml:outerBoundaryIs>
        <gml:LinearRing>
          <gml:coordinates>0,0 0,5 5,5 5,0 ...</gml:coordinates>
        </gml:LinearRing>
      </gml:outerBoundaryIs>
    </gml:Polygon>
  </ogc:Within>
  <ogc:And>
    <ogc:PropertyIsGreaterThanOrEqualTo>
      <ogc:PropertyName>BUILTUPA_1M/POPULATION</ogc:PropertyName>
      <ogc:Literal>100</ogc:Literal>
    </ogc:PropertyIsGreaterThanOrEqualTo>
    <ogc:PropertyIsLessThanOrEqualTo>
      <ogc:PropertyName>BUILTUPA_1M/POPULATION</ogc:PropertyName>
      <ogc:Literal>2000</ogc:Literal>
    </ogc:PropertyIsLessThanOrEqualTo>
  </ogc:And>
</ogc:And>
</ogc:Filter>
</wfs:Delete>
</wfs:Transaction>

```

在响应成功完成的请求时，WEB要素服务（网络要素服务）生成以下响应文档：

```

<?xml version="1.0" ?>
<wfs:TransactionResponse
  version="1.0.0"
  xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-transaction.xsd">
  <wfs:InsertResult handle="Statement 1">
    <ogc:FeatureId fid="ELEVP_1M.1001"/>
  </wfs:InsertResult>
  <wfs:InsertResult handle="ComplexInsert">
    <ogc:FeatureId fid="ROADL_1M.1553"/>
  </wfs:InsertResult>
  <wfs:InsertResult handle="Statement 2">
    <ogc:FeatureId fid="BUILTUPA_1M.509876"/>
    <ogc:FeatureId fid="BUILTUPA_1M.509877"/>
  </wfs:InsertResult>
</wfs:TransactionResponse>

```

```

</wfs:InsertResult>
<wfs:TransactionResult handle="Transaction 01">
  <wfs:Status>
    <wfs:SUCCESS/>
  </wfs:Status>
</wfs:TransactionResult>
</wfs:TransactionResponse>

```

10 GetCapabilities 操作

10.1 介绍

WEB要素服务（网络要素服务）必须具有描述自身能力的本领。这一节定义了一个XML文档，用于WEB要素服务（网络要素服务）描述其自身能力。

在这个规范中定义的能力文档与在web地图服务器实现规范中定义的Map服务器的能力文档[1]相类似。

10.2 请求

<GetCapabilities>元素用于从一个WEB要素服务（网络要素服务）寻求能力文档。定义为下面的XML模式片断：

```

<xsd:element name="GetCapabilities" type="wfs:GetCapabilitiesType"/>
<xsd:complexType name="GetCapabilitiesType">
  <xsd:attribute name="version"
    type="xsd:string" use="optional"/>
  <xsd:attribute name="service"
    type="xsd:string" use="required" fixed="WFS"/>
</xsd:complexType>

```

这个服务的属性在7、8节中已经描述了。与一般情况的情况不同，version属性不是强制的，因为客户应用程序可能不知道服务器支持哪个版本。客户端和服务需要采用在6.2.4节中描述的版本协商规则。

10.3 响应

10.3.1 响应模式

对GetCapabilities请求的响应模式被XML格式标准化的定义了，见附录A.4。

10.3.2 能力文档

能力文档由以下4部分组成：

1) 服务部分

服务部分提供了服务自身的信息。

2) 能力部分

能力部分详细描述了WFS能够处理的一系列请求。基于它们支持的能力的两类WEB要素服务（网络要素服务）在综述中定义了（见I部分）。

3) 要素类型列表

这部分定义了WEB要素服务（网络要素服务）可以提供服务的要素类型列表（以及每个要素类型的操作）。同时提供了关于每个要素类型的附加信息，如SRS。

4) 过滤器能力部分

能力过滤器部分在过滤器编码执行规范中定义[3]。这是可选部分。如果它存在，那么WFS将支持其相应的操作。如果没有定义能力过滤器部分，那么客户端将假设只支持在过滤器编码执行规范中定义的最小缺省过滤器操作[3]。

version属性指定了这个模式所采用的规范修订版本。它的格式是由被分隔开的一个、两个或3个整数组成：“x”，或“x.y”或“x.y.z”，最重要的数字放在第一位。以后的版本保证是单调增加的数字形式，尽管在序列中间会出现间隙。

updateSequence属性是用于管理能力文档内容传播的。例如，一个要素服务加上一些要素类型，它会增加序列数，来告知目录服务器它们以前的缓存版本现在已经失效了。格式是正整数。

10.3.3 服务部分

<Service>元素包括要素服务整体的元数据。下面的内容能够描述**<Service>**元素。

表3 服务部分元素

元素名称	描述
Name	WEB 要素服务（网络要素服务）所提供的服务的名称
Title	<Title> 标题是供人阅读标题，在菜单中的简短标识该服务器
Abstract	<Abstract> 是对更多服务信息的描述
Keyword	<Keyword> 包含帮助目录搜索的简短短词
OnOnlineResource	< OnOnlineResource > 元素定义了该服务的高级 HTTP URL，例如网页的 url。
Fees	<Fees> 标识使用该服务或从 WFS 进行数据搜索时所需进行支付的经费。关键字 None 表示免费的。
AccessConstraints	标识在 WFS 上提供服务时或从服务进行数据检索时所描述的访问限制。关键字 None 表示没有访问限制。

10.3.4 能力部分

能力部分用于定义特定WFS实现提供的一系列WFS操作。一个基本的WFS包括GetCapabilities, DescribeFeatureType和GetFeature操作。一个事务处理的WFS也包括事务处理操作，可能有LockFeature操作和/或GetFeatureWithLock操作。

由WFS所执行的明确的能力由下列元素表示。

表4 WEB 要素服务（网络要素服务）操作

元素名称	描述
GetCapabilities	<GetCapabilities> 元素用来定义该服务可用的分布式计算平台
DescribeFeatureType	<DescribeFeatureType> 元素用来定义服务可用的分布式计算平台并且指出哪一个模式描述语言能用于描述要素类型的模式，当客户端在请求这种描述时。XML 模式是唯一的有效强制语言并且必须可用。SCHEMALANGUAGES 实体能够被重新定义为包含由提供商指定的某种语言。
Transaction	<Transaction> 元素用来定义该服务可用的分布式计算平台
GetFeature	<GetFeature> 元素用来定义服务可用的分布式计算平台和列举表示请求结果的有效格式。RESULTFORMATS 实体定义了 GML 的强制输出格式，也可以重新定义为包括由提供商指定的某种格式。

LockFeature	<LockFeature>锁定要素。
-------------	--------------------

唯一有效的分布式计算平台是HTTP，因此需要定义两种请求方式：GET和POST。onlineResource属性表明用于 HTTP GET 请求的 URL 前缀（在问题标号和查询串前的部分：
http://hostname[:port]/path/scriptname）；对于POST请求，onlineResource是全部URL。

<VendorSpecificCapabilities>元素用来包括提供商指定扩展。

10.3.5 要素类型列表部分

<FeatureTypeList>元素的目的是包含一系列WFS可以提供服务的要素类型列表，并且定义了每个要素类型所支持的事务处理和查询元素。可能的事务处理和查询元素是：

表5 要素的事务处理和查询元素

元素名称	描述
插入	<Insert>元素用于表示 WFS 可以创建新的要素类型实例
更新	<Update>元素用于表示 WFS 能够改变要素的存在状态
删除	<Delete>元素表示 WFS 能够从数据序列中删除或移动要素类型
查询	<Query>元素表示 WFS 能够执行要素类型的查询操作
锁定	<Lock>元素表示 WFS 能够锁定要素类型的实例

Transaction和query元素能够被全局的指定，用来描述所有类型的要素；也可以被本地地指定，用来描述包含在<FeatureTypeList>元素特定要素类型。全局指定的Transaction和query元素可以被包含在<FeatureTypeList>元素中的每个要素类型继承，并且可以被特定的本地Transaction和query元素扩展。例如：<Query>元素可以全局地指定<FeatureTypeList>中所有要素类型，而<Update>元素只能本地地指定一小部分要素类型。如果在任何地方都没有Transaction和query元素的定义，那么缺省的<Query>元素将隐含的定义在<FeatureTypeList>所有的要素类型中。

下面的元素能用于描述包含在<FeatureTypeList>中的每一个要素类型。

表6 描述要素类型的元素

元素名称	描述
Name	要素类型在命名空间中指定的类型。这个元素是强制的。
Title	<Title>标题是供人阅读的标题，它在菜单中对要素类型作了简短标识
Abstract	<Abstract>要素类型更多信息的描述
Keyword	<Keyword>包含帮助目录搜索的简短单词
SRS	<SRS>元素用于表示哪种空间参考系统用于描述要素的状态。SRS 可以采用 POSC 开放软件格式或在 4.3.2 节中定义的 URL 格式。
Operations	<Operations>元素用于定义一个要素类型支持哪种操作。任何本地定义操作优先于全局定义操作。
LatLongBoundingBox	LatLongBoundingBox 元素用于表示在与 SRS 相关的要素类型的一个封闭矩形边界。其目的是通过表明特殊要素类型存在的位置方便地理搜索。由于可以指定多个 LatLongBoundingBox，WFS 可以指出各种数据存在在哪儿。这个信息使得客户端顺速的查询到他们的数据以方便使用。
MetadataURL	WFS 可以使用 0 个或多个<MetadataURL>提供细节，和关于特定要素类型数据的标准化的元数据。类型属性表明元数据遵守的标准；格式属性表示元数据如何构造。目前有两类定义：‘TC211’ = ISO TC211 19115；‘FGDC’ = FGDC CSDGM

10.4 异常

事实上，一个web服务器在为GetCapabilities请求提供服务时遇到错误，它将产生一个异常如7.7章中描述的。

10.5 示例

这个示例指出一个能力文档将看上去像一个基本WEB要素服务（网络要素服务）。为请求一个能力文档，客户端将发出下列请求：

```
<?xml version="1.0" ?>
<GetCapabilities
  service="WFS"
  xmlns="http://www.opengis.net/wfs"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-basic.xsd"/>
```

为响应上述请求，一个web服务器可以产生如下文档：

```
<?xml version="1.0" ?>
<WFS_Capabilities
  version="1.0.0"
  xmlns="http://www.opengis.net/wfs"
  xmlns:myns="http://www.someserver.com/myns"
  xmlns:ogc="http://www.opengis.net/ogc"
  ">

  <!-- The SERVICE section says something about who is providing the -->
  <!-- service and where one can go to obtain more information about -->
  <!-- the service. -->
  <Service>
    <Name>CubeWerx WFS</Name>
    <Title>CubeWerx Web Feature Service</Title>
    <Abstract>Web Feature Server maintained by CubeWerx Inc.</Abstract>
    <OnlineResource>http://www.someserver.com/wfs/cwwfs.cgi?</OnlineResource>
  </Service>

  <!-- The CAPABILITY section defines which WFS operations this -->
  <!-- service instance supports, what distributed computing platform -->
  <!-- is supported for each service and what the entry point is for -->
  <!-- each operation. -->
  <Capability>
    <Request>
      <GetCapabilities>
        <DCPType>
          <HTTP>
            <Get onlineResource="http://www.someserver.com/wfs/cwwfs.cgi?" />
          </HTTP>
```

```

        </DCPType>
        <DCPType>
            <HTTP>
                <Post onlineResource="http://www.someserver.com/wfs/cwwfs.cgi"/>
            </HTTP>
        </DCPType>
    </GetCapabilities>
    <DescribeFeatureType>
        <SchemaDescriptionLanguage>
            <XMLSCHEMA/>
        </SchemaDescriptionLanguage>
        <DCPType>
            <HTTP>
                <Get onlineResource="http://www.someserver.com/wfs/cwwfs.cgi?"/>
            </HTTP>
        </DCPType>
        <DCPType>
            <HTTP>
                <Post onlineResource="http://www.someserver.com/wfs/cwwfs.cgi"/>
            </HTTP>
        </DCPType>
    </DescribeFeatureType>
    <GetFeature>
        <ResultFormat>
            <GML2/>
        </ResultFormat>
        <DCPType>
            <HTTP>
                <Get onlineResource="http://www.someserver.com/wfs/cwwfs.cgi?"/>
            </HTTP>
        </DCPType>
        <DCPType>
            <HTTP>
                <Post onlineResource="http://www.someserver.com/wfs/cwwfs.cgi"/>
            </HTTP>
        </DCPType>
    </GetFeature>
    <GetFeatureWithLock>
        <ResultFormat>
            <GML2/>
        </ResultFormat>
        <DCPType>
            <HTTP>
                <Get onlineResource="http://www.someserver.com/wfs/cwwfs.cgi?"/>

```

```

        </HTTP>
    </DCPType>
    <DCPType>
        <HTTP>
            <Post onlineResource="http://www.someserver.com/wfs/cwwfs.cgi"/>
        </HTTP>
    </DCPType>
</GetFeatureWithLock>
<Transaction>
    <DCPType>
        <HTTP>
            <Get onlineResource="http://www.someserver.com/wfs/cwwfs.cgi?"/>
        </HTTP>
    </DCPType>
    <DCPType>
        <HTTP>
            <Post onlineResource="http://www.someserver.com/wfs/cwwfs.cgi"/>
        </HTTP>
    </DCPType>
</Transaction>
<LockFeature>
    <DCPType>
        <HTTP>
            <Get onlineResource="http://www.someserver.com/wfs/cwwfs.cgi?"/>
        </HTTP>
    </DCPType>
    <DCPType>
        <HTTP>
            <Post onlineResource="http://www.someserver.com/wfs/cwwfs.cgi"/>
        </HTTP>
    </DCPType>
</LockFeature>
</Request>
</Capability>

<!-- The FEATURETYPELIST section defines the list of feature types -->
<!-- that this service instance can operate upon as well as which -->
<!-- operations are supported on each feature type. -->
<FeatureTypeList>
    <Operations>
        <Query/>
    </Operations>
    <FeatureType>
        <Name>myns:BUILTUPA_1M</Name>

```

```

    <SRS>EPSG:4326</SRS>
    <Operations>
      <Insert/>
      <Update/>
      <Delete/>
    </Operations>
    <LatLongBoundingBox minx="-179.1296081543" miny="-53.167423248291"
      maxx="178.44325256348" maxy="70.992721557617"/>
  </FeatureType>
  <FeatureType>
    <Name>myns:COASTL_1M</Name>
    <SRS>EPSG:4326</SRS>
    <LatLongBoundingBox minx="-179.99942016602" miny="-85.582763671875"
      maxx="179.9999" maxy="83.627418518066"/>
  </FeatureType>
  <FeatureType>
    <Name>myns:ELEVP_1M</Name>
    <SRS>EPSG:4326</SRS>
    <Operations>
      <Insert/>
      <Update/>
      <Delete/>
    </Operations>
    <LatLongBoundingBox minx="-179.9984893715" miny="-89.83837892767"
      maxx="179.99234007206" maxy="83.520408603363"/>
  </FeatureType>
  <FeatureType>
    <Name>myns:OCEANSEA_1M</Name>
    <SRS>EPSG:4326</SRS>
    <Operations>
      <Insert/>
      <Update/>
      <Delete/>
    </Operations>
    <LatLongBoundingBox minx="-179.9999" miny="-85.582763671875"
      maxx="179.99996948242" maxy="89.9999"/>
  </FeatureType>
  <FeatureType>
    <Name>myns:RAILRDL_1M</Name>
    <SRS>EPSG:4326</SRS>
    <Operations>
      <Insert/>
      <Update/>
      <Delete/>

```

```

    </Operations>
    <LatLongBoundingBox minx="-165.24467468262" miny="-53.138427734375"
                        maxx="179.60989379883" maxy="78.16796875" />
</FeatureType>
<FeatureType>
  <Name>myns:TREESA_1M</Name>
  <SRS>EPSG:4326</SRS>
  <Operations>
    <Insert/>
    <Update/>
    <Delete/>
  </Operations>
  <LatLongBoundingBox minx="-139.99757385254" miny="25.281270980835"
                        maxx="-52.661720275879" maxy="66.718765258789" />
</FeatureType>
</FeatureTypeList>

<!-- The FILTER_CAPABILITIES section defines the capabilities of the -->
<!-- filter supported by this feature instance. For example, in -->
<!-- this case all spatial operator are supported. Another, simpler -->
<!-- WFS implementation, may only support the BBOX operator. -->
<ogc:Filter_Capabilities>
  <ogc:Spatial_Capabilities>
    <ogc:Spatial_Operators>
      <ogc:BBOX/>
      <ogc:Equals/>
      <ogc:Disjoint/>
      <ogc:Intersect/>
      <ogc:Touches/>
      <ogc:Crosses/>
      <ogc:Contains/>
      <ogc:Overlaps/>
    </ogc:Spatial_Operators>
  </ogc:Spatial_Capabilities>
  <ogc:Scalar_Capabilities>
    <ogc:Logical_Operators/>
    <ogc:Comparison_Operators>
      <ogc:Simple_Comparisons/>
      <ogc:Like/>
      <ogc:Between/>
      <ogc:NullCheck/>
    </ogc:Comparison_Operators>
    <ogc:Arithmetic_Operators>
      <ogc:Simple_Arithmetic/>

```

```

    <ogc:Functions>
      <ogc:Function_Names>
        <ogc:Function_Name nArgs="1">MIN</ogc:Function_Name>
        <ogc:Function_Name nArgs="1">MAX</ogc:Function_Name>
        <ogc:Function_Name nArgs="1">COUNT</ogc:Function_Name>
        <ogc:Function_Name nArgs="1">DISTINCT</ogc:Function_Name>
      </ogc:Function_Names>
    </ogc:Functions>
  </ogc:Arithmetic_Operators>
</ogc:Scalar_Capabilities>
<ogc:Filter_Capabilities>
</WFS_Capabilities>

```

11 “名—值” 对编码

11.1 介绍

这一节介绍如何使用 标准CGI “名—值” 对格式对WFS操作编码。这意味着参数由名字和值以“名字 = 值”的形式成对组成，并且各对间以“&”符号分割。这种形式的编码就如同URL编码。

11.1.1 示例介绍

通常的，URL编码需要某种标识符，例如“&”，当它们没有用于特定的格式时被避免（转义）[10]。然而，在这一部分，这些标识符为了清晰的原因而不能被避免。

另外，在这一部分许多例子包括一个FILTER参数，其值是一个XML编码的过滤器就如同在过滤器编码执行规范[3]中规定的一样。为保证严格的正确性，这些示例将在<Filter>根元素中包含命名空间和模式定位信息，这样使得XML是有效的。因而参数：

```

FILTER=<Filter><Within><PropertyName>INWATERA_1M/WKB_GEOM<PropertyName>
  <gml:Box><gml:<coordinates>10,10 20,20</gml:coordinates></gml:Box>
</Within></Filter>

```

将更正确的指定如下

```

FILTER=<Filter xmlns="http://www.opengis.net/ogc"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/ogc
    ../filter/1.0.0/filter.xsd
    http://www.opengis.net/gml
    ../gml/2.1/geometry.xsd">
  <Within><PropertyName>INWATERA_1M/WKB_GEOM<PropertyName>
  <gml:Box><gml:<coordinates>10,10 20,20</gml:coordinates></gml:Box>
</Within></Filter>

```

然而，为了使基本信息清晰，这一部分的例子中省略了命名空间和模式的定位属性标签。另外，模式定位显示的仅仅是定位示例，它必须替换成正确模式定位。

最后，FILTER的值和其他参数出于清楚的原因被去掉。实际上，FILTER参数值将组合成一个很长的字符串。

11.2 请求参数规则

11.2.1 参数顺序和大小写

参数名称**将不**区分大小写，参数值**将要**区分大小写。在这个文档中，参数名称典型的以大写字母是为了清晰，并不是一定如此。

请求参数**可能**以任何次序给出。

WEB要素服务（网络要素服务）器**必须**做好遇到不属于此规范的参数部分的准备。就这个规范的产生结果而言，WEB要素服务（网络要素服务）器**应该**忽略这些参数。

11.2.2 参数列表

在列表中的参数需要使用逗号“，”作为分隔符。另外，多个列表可以作为一个参数的值，通过将每个列表放入圆括号中“（”，“）”。

例1

一个项目列表的例子：

`parameter=item1,item2,item2`

例2：

多个列表赋予一个参数的例子：

`parameter=(item11,item12,item13)(item21,item22,item23)`

当TYPENAME指定多个要素类型参数时，圆括号也可以用于分隔多个本地过滤器。下面的URL片段指出它是如何工作的：

`typename=FEAT1,FEAT2&filter=(<Filter>... FEAT1 filter...</Filter>)(<Filter>... FEAT2 filter...</Filter>)`

11.3 公共请求参数

11.3.1 版本参数

VERSION参数指定了协议版本号。版本号格式和版本协议在4.2.4节描述。

11.3.2 请求参数

REQUEST参数指出调用哪个服务操作。这些操作名的值**必须**是WEB要素服务（网络要素服务）实例提供的。

11.3.3 边界框

边界框参数，BBOX，包含在这个规范中边界框参数可以作为普通边界框过滤器的速记表示，普通边界框过滤器如果用XML和在[3]中描述的过滤器编码表达会非常长。BBOX适用于请求中列出的所有应用类型。

BBOX是由四个逗号分割的小数集，科学符号或整数（如果需要浮点数，整数可以看作是小数点在数的末尾）。这些值指出最小X，最小Y，最大X，最大Y的范围，在这种次序下，首先询问SRS要素类型的单位。四个边界框的值表示矩形的外边界，如下图，最小X是左边，最大X是右边，最小Y是下面，最大Y是上面。

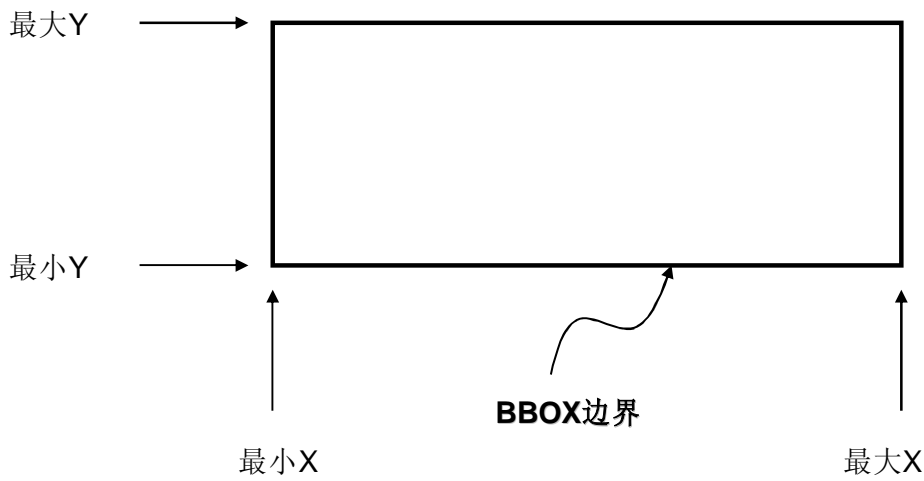


图5 边界框表示

边界框区域面积**不能**为0。

如果请求包括一个无效的边界框（例：最小X大于或等于最大X，或最小Y大于或等于最大Y），服务器**必须**返回一个异常。

如果一个请求所包围的边界框区域与XML能力中记录的LatLongBoundingBox(s)交叠，服务器将返回一个空值（例NULL要素），任何元素必须部分的或全部的包含在边界框中才可能返回一个正确的格式。

在请求中边界框的SRS必须和要素类型的SRS一致。要素类型的SRS记录在WFS的能力文档中。如果在请求中有大于一种的要素类型，要素类型必须具有相同的SRS以及BBOX也必须具有相同的SRS。

如果边界框的值没有在给定的SRS中定义（例：在EPSG：4326中纬度大于90度），服务器将返回在有效的SRS区域范围外的一个空值。

在经度特殊的情况下，下面的方法用于子午线在180度经度的情况下。存在合法的情况地图跨度子午线（例如：地图中心在太平洋）。如果最小X在最西经度并且最大X在最东，那么可采用下列约束：

$$-180 \leq X_{min} < X_{max} < 540$$

例

Xmin, Xmax值和相应的边界框范围：

-180,180 = 地球中心在格林威治

0,360 = 格林威治左边界

120,250 = 太平洋

11.3.4 提供商指定的参数

请求中允许可选的提供商指定参数(VSPs)，这些参数可以增强对请求的响应结果。典型地，它们被用来做非标准功能的私人测试，进而可能会变为标准。一个普通的客户端**不**要求使用这些VSPs。

WEB要素服务（网络要素服务）**必须**生成一个合法的结果，即使VSPs丢失或错误构成的（例如，服务**应该**提供一个缺省值），或者VSPs在不被服务知道的情况下被应用（例如，服务**应该**忽略未知的请求参数）。

WEB要素服务（网络要素服务）可以选择不声明部分或全部。如果VSPs 被包含进能力XML，VendorSpecificCapabilities 元素必须被相应的重定义。其它的 VSPs模式文档可能会引入包含的元素的重定义。

客户端**可以**从能力模式中读提供商指定的定义，并且使用VSPs构成请求。

提供商**应该**小心地选择提供商指定参数名，以避免与标准参数冲突。

11.4 公共参数

下表描述了所有WFS请求的公共参数。后续表将重新定义一个或更多参数的某些方面。

表7 —WFS 请求的通用参数

URL 组成	O/M ²	缺省值	描述
http://server_address/path/script	M		WEB 要素服务（网络要素服务）URL 前缀
VERSION	M ¹	1.0.0	请求版本信息
SERVICE	M	WFS	服务器类型。
REQUEST	M		WFS 请求名称
Additional parameters	0		在这一部分描述的参数
Vendor-specific parameters	0		可选的特殊参数。

强制性VERSION参数指定了协议的版本号，并且考虑到在6.2.4部分描述的协议。

强制性SERVICE参数指定了在特定的服务中被调用的有效的服务类型。WFS值用于表示被调用的web服务器。

也需要包括REQUEST参数，并且指出调用了哪个web服务器操作。REQUEST参数的可能值是：DescribeFeatureType, LockFeature, Transaction, GetFeature, GetFeatureWithLock 或 GetCapabilities。

另外，这一节中描述的GET参数将表达为一对“名—值”。参数名字不区分大小写。参数值区分大小写。请求中的参数可以用任意顺序存放。

WFS必须可以处理该规范中没有定义参数类型。它们被称为供应商指定参数。供应商指定参数允许供应商指定增强请求结果的附加参数。即使供应商指定参数是错误的或是不全的，WFS必须生成有效的结果。WFS可以在它的XML能力文档中声明提供供应商指定参数。WFS可以选择声明一些或全部提供供应商指定参数。客户端可以读取能力模式，并使用声明了的供应商指定参数形成请求。

11.5 响应

任何使用“名—值”对编码的请求将与用XML编码的请求的响应是一样的，在本文档的头几个章节中描述了。

11.6 异常

采用“名—值”对编码的请求的异常报告将与采用XML编码的请求一样。参考5.7节和9.4，详细讨论了异常报告。

11.7 操作

11.7.1 介绍

这一节描述如何用标准CGI类型“名—值”对编码描述WFS请求。大量的用例说明了各种可能的形式。另外，为了清晰，例子中的每个参数规范为分割线上。

11.7.2 描述要素类型操作

11.7.2.1 请求

表8 —要素类型编码描述

URL 组成	O/M	缺省值	描述
REQUEST=DescribeFeatureType	M		请求名字
TYPENAME	0		用逗号分割要素类型列表。如果没有

			指定值就认为是所有要素类型。
OUTPUTFORMAT	0	XML 模式	用于描述要素的输出格式。XML 模式格式必须支持。其他输出格式，例如 DTD 可能支持。

11.7.2.2 例子

例1:

下面例子请求TREESA_1M要素类型的模式描述。

```
http://www.someserver.com/wfs.cgi?
SERVICE=WFS&
VERSION=1.0.0&
REQUEST=DescribeFeatureType&
TYPENAME=TREESA_1M
```

例2:

下面例子请求INWATERA_1M 和 BUILTUPA_1M要素类型的模式描述

```
http://www.someserver.com/wfs.cgi?
SERVICE=WFS&
VERSION=1.0.0&
REQUEST=DescribeFeatureType&
TYPENAME=TREESA_1M,BUILTUPA_1M
```

11.7.3 GetFeature & GetFeatureWithLock 操作

11.7.3.1 请求

表9 – GetFeature & GetFeatureWithLock 编码

URL 组成	O/M	缺省值	描述
REQUEST=[GetFeature GetFeatureWithLock]	M		WFS 请求的名字

PROPERTYNAME	0		特性列表可以指定将查询的要素类型。参见 [13.2.2] 如何形成参数列表。“*” 字符能够表示所有被检索的特性。在每一个要素 ID 或类型名称列表和特性名称列表形成 1: 1 的映射。缺省值表示所有特性都可以获得。
FEATUREVERSION=[ALL N]	0		如果支持版本, 要素版本参数指导 WFS 到响应要素版本中取得。“ALL” 值表示提取所有要素版本信息。整数值表示提起第 N 个要素版本信息。没有值表示提取最近的要素版本信息。
MAXFEATURES=N	0		一个正整数表示 WFS 将对一个请求回答的最大要素数。如果没有值将呈现所有的结果。
TYPENAME (Optional if FEATUREID is specified.)	M		请求的要素类型名称。
FEATUREID (Mutually exclusive with FILTER and BBOX)	0		通过要素标识符来获得要素枚举列表。
FILTER (Prerequisite: TYPENAME) (Mutually exclusive with FEATUREID and BBOX)	0		过滤器规范描述一系列要素操作。过滤器遵照过滤器编码规范定义[3]。如果采用过滤器参数, 对于每一个在类型名称参数列表中的要素类型必须规定过滤器。单个以过滤器参数编码的过滤器以圆括号表示, “(” 和 “)”
BBOX (Prerequisite: TYPENAME) (Mutually exclusive with FEATUREID and FILTER.)	0		在要素 ID 或过滤器的环境下, 客户端可以指定边界框, 如 13.3.3 中描述的那样。

11.7.3.2 示例

在这一节中许多例子包括FILTER参数, 其值遵循过滤器编码执行规范[3]的XML编码过滤器。为保证绝对正确, 这些例子将包括命名空间和In<Filter>元素根结点下的模式定位信息, 使得XML生效。参数如下:

```
FILTER=<Filter><Within><PropertyName>INWATERA_1M/WKB_GEOM<PropertyName>
    <gml:Box><gml:<coordinates>10,10 20,20</gml:coordinates></gml:Box>
</Within></Filter>
```

更正确的写法:

```
FILTER=<Filter xmlns="http://www.opengis.net/ogc"
    xmlns:gml="http://www.opengis.net/gml"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.opengis.net/ogc
```

```

        ../filter/1.0.0/filter.xsd
        http://www.opengis.net/gml
        ../gml/2.1/geometry.xsd">
    <Within><PropertyName>INWATERA_1M/WKB_GEOM<PropertyName>
    <gml:Box><gml:><coordinates>10,10 20,20</gml:coordinates></gml:Box>
    </Within></Filter>

```

然而，为清楚起见，命名空间和模式定位属性标签在这一节的例子中缺省。另外，模式定位仅仅是例子，正确的模式定位需要指定。

最后，为了清晰，FILTER参数的值被几行打断。实际的FILTER参数将是很长的字符串。

例1:

查询所有INWATERA_1M要素类型实例的所有特性。

```

http://www.someserver.com/wfs.cgi&
    SERVICE=WFS&
    VERSION=1.0.0&
    REQUEST=GetFeature&
    TYPENAME=INWATERA_1M

```

例2:

查询INWATERA_1M要素类型实例的某些特性。

```

http://www.someserver.com/wfs.cgi&
    SERVICE=WFS&
    VERSION=1.0.0&
    REQUEST=GetFeature&
    PROPERTYNAME=INWATERA_1M/WKB_GEOM, INWATERA_1M/TILE_ID&
    TYPENAME=INWATERA_1M

```

例3:

查询要素标识符为“INWATERA_1M.1013”的要素的所有特性。

```

http://www.someserver.com/wfs.cgi&
    SERVICE=WFS&
    VERSION=1.0.0&
    REQUEST=GetFeature&
    FEATUREID=INWATERA_1M.1013

```

例4:

查询要素标识符为“INWATERA_1M.1013”的要素的一些特性。

```

http://www.someserver.com/wfs.cgi&
    SERVICE=WFS&
    VERSION=1.0.0&
    REQUEST=GetFeature&
    PROPERTYNAME=INWATERA_1M/WKB_GEOM, INWATERA_1M/TILE_ID&
    FEATUREID=INWATERA_1M.1013

```

例5:

查询要素类型为“INWATERA_1M”的一系列要素的所有特性。

```
http://www.someserver.com/wfs.cgi&
    SERVICE=WFS&
    VERSION=1.0.0&
    REQUEST=GetFeature&
    FEATUREID=INWATERA_1M.1013,INWATERA_1M.1014,INWATERA_1M.1015
```

例6:

查询要素类型为“INWATERA_1M”的列举要素实例的某些特性。在这个例子中，为每个要素实例选择了WKB_GEOM和TILE_ID特性用来。

```
http://www.someserver.com/wfs.cgi&
    SERVICE=WFS&
    VERSION=1.0.0&
    REQUEST=GetFeature&
    PROPERTYNAME=(INWATERA_1M/WKB_GEOM,INWATERA_1M/TILE_ID)
                  (INWATERA_1M/WKB_GEOM,INWATERA_1M/TILE_ID)
                  (INWATERA_1M/WKB_GEOM,INWATERA_1M/TILE_ID)&
    FEATUREID=INWATERA_1M.1013,INWATERA_1M.1014,INWATERA_1M.1015
```

例7:

查询INWATERA_1M要素类型的由限制条件限制的枚举要素实例的所有特性。

```
http://www.someserver.com/wfs.cgi&
    SERVICE=WFS&
    VERSION=1.0.0&
    REQUEST=GetFeature&
    TYPENAME=INWATERA_1M&
    FILTER=<Filter><Within><PropertyName>INWATERA_1M/WKB_GEOM<PropertyName>
          <gml:Box><gml><coordinates>10,10 20,20</gml:coordinates></gml:Box>
          </Within></Filter>
```

例8:

查询 INWATERA_1M 要素类型由限制条件限制的枚举要素的某些特性。

```
http://www.someserver.com/wfs.cgi&
    SERVICE=WFS&
    VERSION=1.0.0&
    REQUEST=GetFeature&
    PROPERTYNAME=INWATERA_1M/WKB_GEOM,INWATERA_1M/TILE_ID&
    TYPENAME=INWATERA_1M&
```

```
FILTER=<Filter><Within><PropertyName>INWATERA_1M/WKB_GEOM<PropertyName><gml:Box>
```

```
<gml:coordinates>10,1020,20</gml:coordinates></gml:Box></Within></Filter>
```

例9:

查询要素类型列表中的所有特性。

```
http://www.someserver.com/wfs.cgi&
SERVICE=WFS&
VERSION=1.0.0&
REQUEST=GetFeature&
TYPENAME=INWATERA_1M,BUILTUPA_1M
```

例10:

查询要素类型列表中的某些特性。在这个例子中，INWATERA_1M 要素类型的WKB_GEOM 和TILE_ID 特性被获得，并且获得BUILTUPA_1M要素类型的所有特性。

```
http://www.someserver.com/wfs.cgi&
SERVICE=WFS&
VERSION=1.0.0&
REQUEST=GetFeature&
PROPERTY=(INWATERA_1M/WKB_GEOM,INWATERA_1M/TILE_ID)(BUILTUPA_1M/*)&
TYPENAME=INWATERA_1M,BUILTUPA_1M
```

例11

查询一系列枚举要素实例的所有特性。

```
http://www.someserver.com/wfs.cgi&
SERVICE=WFS&
VERSION=1.0.0&
REQUEST=GetFeature&
FEATUREID=INWATERA_1M.1013,BUILTUP_1M.3456
```

例12:

查询一系列枚举要素实例的某些特性。在这个例子中，要素"INWATERA_1M.1013"可获得WKB_GEOM 和TILE_ID attributes特性。要素"BUILTUPA_1M.3456"可获得WKB_GEOM属性。

```
http://www.someserver.com/wfs.cgi&
SERVICE=WFS&
VERSION=1.0.0&
REQUEST=GetFeature&
PROPERTYNAME=INWATERA_1M/WKB_GEOM,INWATERA_1M/TILE_ID,BUILTUPA_1M/WKB_GEOM)&
FEATUREID=INWATERA_1M.1013,BUILTUPA_1M.3456
```

例13:

查询INWATERA_1M要素类型的要素实例的所有特性，以及BUILTUPA_1M类型在指定边界框中的所有要素实例。

```
http://www.someserver.com/wfs.cgi&
SERVICE=WFS&
VERSION=1.0.0&
REQUEST=GetFeature&
TYPENAME=INWATERA_1M,BUILTUPA_1M&
FILTER=(<Filter><Within><PropertyName>INWATERA_1M/WKB_GEOM
```

```
<PropertyName><gml:Box><gml:coordinates>10,10 20,20</gml:Box>
</gml:coordinates></Within></Filter>)(<Filter><Within><PropertyName>
BUILTUPA_1M/WKB_GEOM<PropertyName><gml:Box><gml:coordinates>10,10
20,20</gml:Box></gml:coordinates></Within></Filter>)
```

例14:

查询要INWATERA_1M 和 BUILTUPA_1M素类型的所有要素示例的某些特性。

<http://www.SiriusCyberneticsCorp.com/wfs.cgi&>

SERVICE=WFS&

VERSION=1.0.0&

REQUEST=GetFeature&

PROPERTYNAME=(INWATERA_1M/WKB_GEOM,INWATERA_1MTILE_ID)(BUILTUPA_1M/WKB_GEOM)&

TYPENAME=INWATERA_1M,BUILTUPA_1M&

FILTER=(<Filter><Within><PropertyName>INWATERA_1M/WKB_GEOM|INWATERA_1M/WKB_GEOM

<PropertyName><gml:Box><gml:coordinates>10,10 20,20</gml:coordinates>

</gml:Box></Within></Filter>)(<Filter><Within><PropertyName>

INWATERA_1M/WKB_GEOM<PropertyName><gml:Box><gml:coordinates>10,10

20,20</gml:coordinates></gml:Box></Within></Filter>)

11.7.4 LockFeature 操作

11.7.4.1 请求

表10 LockFeature encoding

URL 组成	O/M	缺省值	描述
REQUEST=LockFeature	M		请求的名字
TYPENAME	M		被锁定的一个或多个要素类型的要素实例的名

(Optional if FEATUREID is specified.)			字。
EXPIRY	0		在清除锁前所持续的分钟数。如果没有值，则锁的持续时间是确定的。
LOCKACTION=[<u>ALL</u> SOME]	0		指定如何获得锁。ALL 表示试图获得所有要素锁否则表示失败。SOME 表示获得尽可能多的要素的锁。
FEATUREID (Mutually exclusive with FILTER and BBOX.)	0		通过要素标识符来获得锁定要素枚举列表。
FILTER (Prerequisite: TYPENAME) (Mutually exclusive with FEATUREID and BBOX.)	0		过滤器规范描述一系列要素操作。过滤器遵照过滤器编码规范定义[3]。如果采用过滤器参数，对于每一个在类型名称参数列表中的要素类型必须规定过滤器。单个以过滤器参数编码的过滤器以圆括号表示，“（”和“）”
BBOX (Prerequisite: TYPENAME) (Mutually exclusive with FEATUREID and FILTER.)	0		在要素 ID 或过滤器的环境下，客户端可以指定边界框，如 13.3.3 中描述的那样。

11.7.4.2 示例

例1

下面示例锁定所有 INWATERA_1M. 要素类型的实例。

```
http://www.someserver.com/wfs.cgi?
  SERVICE=WFS&
  VERSION=1.0.0&
  REQUEST=LockFeature&
  TYPENAME=INWATERA_1M
```

例2:

下列例子锁定标识为“ROADL_1M1013”的要素。

```
http://www.someserver.com/wfs.cgi?
  SERVICE=WFS&
  VERSION=1.0.0&
  REQUEST=LockFeature&
  FEATUREID=ROADL_1M.1013
```

例3:

下列例子锁定要素类型为“INWATERA_1M”和“BUILTUPA_1M”的所有要素。

```
http://www.someserver.com/wfs.cgi?
  SERVICE=WFS&
```

```

VERSION=1.0.0&
REQUEST=LockFeature&
TYPENAME=INWATER_1M,BUILTUPA_1M

```

例4:

下列例子锁定要素类型为“INWATER_1M”和“BUILTUPA_1M”的在用户感兴趣的范围内的所有要素。

```
http://www.someserver.com/wfs.cgi?
```

```

SERVICE=WFS&
VERSION=1.0.0&
REQUEST=LockFeature&
LOCKACTION=ALL&
TYPENAME=INWATER_1M,BUILTUPA_1M&
FILTER=(<Filter><Within><PropertyName>WKB_GEOM</PropertyName><gml:Box>
<gml:coordinates>10,10 20,20</gml:coordinates></gml:Box></Within>

```

```

</Filter>)(<Filter><Within><PropertyName>WKB_GEOM</PropertyName><gml:Box>
<gml:coordinates>10,10 20,20</gml:coordinates></gml:Box></Within>
</Filter>)

```

11.7.5 Transaction 操作

事务处理接口唯一支持的操作是删除操作。插入或更新请求的表达，非常冗长，不方便用一对“关键字—值”对编码。

11.7.5.1 请求**表11 事务处理编码**

URL 组成	O/M	缺省值	描述
REQUEST=Transaction	M		WFS 请求的名字
OPERATION=Delete	M		执行事务处理。当前只定义删除。
TYPENAME (Optional if FEATUREID is specified.)	M		用于操作的要素类型列表。
RELEASEACTION=[<u>ALL</u> SOME]	0		ALL 表示当事务操作结束后释放所有锁定要素。SOME 表示仅仅释放那些做了改动的要素。剩余锁继续维护。
FEATUREID (Mutually exclusive with FILTER and BBOX)	0		用于操作的要素枚举列表。没有缺省值。
FILTER (Prerequisite: TYPENAME) (Mutually exclusive with FEATUREID and BBOX)	0		过滤器规范描述一系列要素操作。过滤器遵照过滤器编码规范定义[3]。如果采用过滤器参数，对于每一个在类型名称参数列表中的要素类型必须规定过滤器。单个以过滤器参数编码的过滤器以圆括号表示，“（”和“）”

BBOX (Prerequisite: TYPENAME) (Mutually exclusive with FILTER and FEATUREID)	0		在要素 ID 或过滤器的环境下，客户端可以指定边界框，如 13.3.3 中描述的那样。
--	---	--	---

11.7.5.2 示例 Examples

例1:

删除标识符为“ROADL_1M.1013”的要素。

http://www.someserver.com/wfs.cgi?

SERVICE=WFS&
VERSION=1.0.0&
REQUEST=Transaction&
OPERATION=Delete&
FEATUREID=ROADL_1M.1013

例2:

删除要素类型为INWATER_1M和BUILTUPA_1M并且位于指定框中的所有要素。

http://www.someserver.com/wfs.cgi?

SERVICE=WFS&
VERSION=1.0.0&
REQUEST=Transaction&
OPERATION=Delete&
TYPENAME=INWATER_1M,BUILTUPA_1M&

FILTER=(
<Filter>
<Within>
<PropertyName>INWATER_1M/WKB_GEOM
<PropertyName>
<gml:Box>
<gml:coordinates>10,10 20,20</gml:coordinates>
</gml:Box>
</Within>
</Filter>
<Filter>
<Within>
<PropertyName>BUILTUPA_1M/WKB_GEOM
<PropertyName>
<gml:Box>
<gml:coordinates>10,10 20,20</gml:coordinates>
</gml:Box>
</Within>
</Filter>
)

例3:

与例2相同，除了BBOX参数是采用删除操作中的空间约束条件。

http://www.someserver.com/wfs.cgi?

SERVICE=WFS&
VERSION=1.0.0&
REQUEST=Transaction&
OPERATION=Delete&
TYPENAME=INWATER_1M,BUILTUPA_1M&
BBOX=10,10,20,20

11.7.6 GetCapabilities 操作

11.7.6.1 请求

表12 获得能力编码

URL 组成	O/M	缺省值	描述
--------	-----	-----	----

REQUEST=GetCapabilities	M		请求的名称
-------------------------	---	--	-------

11.7.6.2 示例 Examples

例1:

请求WFS的能力文档:

<http://www.someserver.com/wfs.cgi?>

VERSION=1.0.0&

SERVICE=WFS&

REQUEST=GetCapabilities

附 录 A

XML 模式定义（规范性附录）

A.1 介绍

这个附录包括WFS操作的标准XML模式定义，能力文档和异常处理。

附录A2包括由WEB要素服务（网络要素服务）产生的异常信息的XML模式定义。WFS产生的所有异常信息必须对这个模式有效。

附录A3包括WEB要素服务（网络要素服务）必须提供的基本要求的XML模式定义。一系列基本操作包括获得能力，描述要素类型和获得要素。一个WEB要素服务（网络要素服务）必须实现这些操作。

附录A4包括与事务处理相关操作的XML模式定义。这一系列操作包括获得锁定要素，要素锁定和事务处理。一个WEB要素服务（网络要素服务）可以实现部分或所有这些操作。

附录A5包括WEB要素服务（网络要素服务）响应获得能力请求时，产生的能力文档的XML模式定义。web服务器的获得能力响应必须对此文档是有效的。更进一步，如果WEB要素服务（网络要素服务）完成客户的特殊扩展，附录A5中的文档必须在需要处进行扩展，使得客户特殊扩展可以被客户应用程序发现。

A.2 OGC-exception.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema
  targetNamespace="http://www.opengis.net/ogc"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">

  <xsd:element name="ServiceExceptionReport">
    <xsd:annotation>
      <xsd:documentation>
        本服务异常报告元素，包含有WFS服务的一或者多个服务异常元素。
      </xsd:documentation>
    </xsd:annotation>
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="ServiceException"
          type="ogc:ServiceExceptionType"
          minOccurs="0" maxOccurs="unbounded">
          <xsd:annotation>
            <xsd:documentation>
              本服务异常元素，包含单个的异常信息用于描述某WFS服务的处理异常。
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
      <xsd:attribute name="version" type="xsd:string" fixed="1.2.0"/>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

```

    </xsd:complexType>
</xsd:element>

```

```

<xsd:complexType name="ServiceExceptionType">

```

```

    <xsd:annotation>

```

```

        <xsd:documentation>

```

本服务异常类型定义了某种服务异常元素。指出该元素的内容是某种异常消息，用于本WFS服务希望传给客户端的请求。

```

        </xsd:documentation>

```

```

    </xsd:annotation>

```

```

    <xsd:simpleContent>

```

```

        <xsd:extension base="xsd:string">

```

```

            <xsd:attribute name="code" type="xsd:string">

```

```

                <xsd:annotation>

```

```

                    <xsd:documentation>

```

一个服务可能关联到的某种异常的编码，通过使用该编码的属性值。

```

                    </xsd:documentation>

```

```

                </xsd:annotation>

```

```

            </xsd:attribute>

```

```

            <xsd:attribute name="locator" type="xsd:string">

```

```

                <xsd:annotation>

```

```

                    <xsd:documentation>

```

本地（locator）这个属性被用来指出：在某个客户端本身的请求里异常的位置。一旦该请求中包含了“handle”属性，是被用来标识出作为该locator属性报告请求中的、特定的异常组成部分报告出来。否则，是说handle属性没有被指定，web服务规范将尝试用其他的方法来定位错误，如用行号或者以请求开始时的字节偏移等方法。

```

                    </xsd:documentation>

```

```

                </xsd:annotation>

```

```

            </xsd:attribute>

```

```

        </xsd:extension>

```

```

    </xsd:simpleContent>

```

```

</xsd:complexType>

```

```

</xsd:schema>

```

A.3 WFS-basic.xsd

```

<?xml version="1.0"?>

```

```

<xsd:schema

```

```

    targetNamespace="http://www.opengis.net/wfs"

```

```

    xmlns:wfs="http://www.opengis.net/wfs"

```

```

    xmlns:ogc="http://www.opengis.net/ogc"

```

```

    xmlns:gml="http://www.opengis.net/gml"

```

```

    xmlns:xlink="http://www.w3.org/1999/xlink"

```

```

    xmlns:xsd="http://www.w3.org/2001/XMLSchema"

```

```
elementFormDefault="qualified">
```

```
<!-- =====
Includes and Imports
===== -->
```

```
<xsd:import namespace="http://www.opengis.net/gml"
            schemaLocation="../../gml/2.1/feature.xsd"/>
<xsd:import namespace="http://www.opengis.net/ogc"
            schemaLocation="../../filter/1.0.0/filter.xsd"/>
```

```
<!-- =====
REQUEST MESSAGES
===== -->
```

```
<xsd:element name="GetCapabilities" type="wfs:GetCapabilitiesType">
```

```
<xsd:annotation>
```

```
<xsd:documentation>
```

获取能力（GetCapabilities）元素用于请求一个WEB要素服务（网络要素服务）生成一篇XML文档（描述该服务的提供机构、该服务支持的各种WFS操作、该服务可操作的要素类型列表和该服务提供的过滤能力）。这种XML文档称为能力文档。

```
</xsd:documentation>
```

```
</xsd:annotation>
```

```
</xsd:element>
```

```
<xsd:element name="DescribeFeatureType" type="wfs:DescribeFeatureTypeType">
```

```
<xsd:annotation>
```

```
<xsd:documentation>
```

描述要素类型（DescribeFeatureType）元素是用于请求一个WEB要素服务（网络要素服务）生成一篇文档描述出一或多种要素类型。

```
</xsd:documentation>
```

```
</xsd:annotation>
```

```
</xsd:element>
```

```
<xsd:element name="GetFeature" type="wfs:GetFeatureType">
```

```
<xsd:annotation>
```

```
<xsd:documentation>
```

获取要素（GetFeature）元素请求是用于请求一个WEB要素服务（网络要素服务）返回一或多种要素类型的要素实例。

```
</xsd:documentation>
```

```
</xsd:annotation>
```

```
</xsd:element>
```

```
<!-- =====
RESPONSE MESSAGES
===== -->
```

```
<xsd:element name="FeatureCollection"
            type="wfs:FeatureCollectionType"
```

```

        substitutionGroup="gml:_FeatureCollection">
<xsd:annotation>
    <xsd:documentation>
        该元素是作为获取要素（GetFeature或者GetFeatureWithLock）响应的一个容器。
    </xsd:documentation>
</xsd:annotation>
</xsd:element>

```

```

<!-- =====
COMMON ATTRIBUTES DOCUMENTATION
=====
-->

```

```

<!-- =====
TYPES
===== -->

```

```

<!-- GETCAPABILITIES -->
<xsd:complexType name="GetCapabilitiesType">

```

```

    <xsd:annotation>
        <xsd:documentation>
            本类型定义了获取能力的操作。在对于获取能力的请求的响应中，WEB要素服务（网络要素服务）需要生成一篇XML能力文档，用来验证在WFS-capabilities.xsd 中定义的文档模型schemas。
        </xsd:documentation>
    </xsd:annotation>

```

```

    <xsd:attribute name="version"
        type="xsd:string" use="optional" fixed="1.0.0"/>
    <xsd:attribute name="service"
        type="xsd:string" use="required" fixed="WFS"/>
</xsd:complexType>

```

```

<!-- DESCRIBEFEATURETYPE -->
<xsd:complexType name="DescribeFeatureTypeType">

```

```

    <xsd:annotation>
        <xsd:documentation>
            DescribeFeatureType操作的功能是生成一个模式描述WFS实现所能提供服务的要素类型。模式定义了如何编码WFS实现所期望的在输入要素实例以及输出时如何生成一个要素实例。一个WEB要素服务（网络要素服务）必然能够生成符合GML 2 的应用模型。
        </xsd:documentation>
    </xsd:annotation>

```

如果DescribeFeatureType元素是空的，它就被解释为请求所以WFS可以提高服务的要素类型。对DescribeFeatureType请求的XML编码用以下XML模式片段定义。

```

    </xsd:documentation>
</xsd:annotation>
<xsd:sequence>
    <xsd:element name="TypeName" type="xsd:QName"
        minOccurs="0" maxOccurs="unbounded">

```

```

        <xsd:annotation>
            <xsd:documentation>
                TypeName元素用来对将要描述的要素类型名进行编码。
            </xsd:documentation>
        </xsd:annotation>
    </xsd:element>
</xsd:sequence>
<xsd:attribute name="version"
                type="xsd:string" use="required" fixed="1.0.0"/>
<xsd:attribute name="service"
                type="xsd:string" use="required" fixed="WFS"/>
<xsd:attribute name="outputFormat"
                type="xsd:string" use="optional" default="XMLSCHEMA">
    <xsd:annotation>
        <xsd:documentation>

```

outputFormat属性被用来指明用来描述要素类型模式描述语言。响应DescribeFeatureType操作的唯一的强制性输出格式是XML模式，用outputFormat属性的XML模式值表示。其他提供商指定格式也可行，但必须在能力文档中作出声明。当outputFormat属性值已经被设为XML模式，WFS实现必须能提供一个合法GML[2]应用Schema的XML[6]模式，并且定义列在请求列表中的要素类型。

```

        </xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
</xsd:complexType>
<!-- GETFEATURE -->
<xsd:complexType name="GetFeatureType">
    <xsd:annotation>
        <xsd:documentation>

```

<GetFeature>元素包含一个或多个<Query>元素，每个<Query>元素包含一个对查询的描述。GetFeature请求中的所有查询结果连结在一起组成结果集。GetFeature请求的响应格式由outputFormat属性控制。outputFormat属性的默认值应该是GML2。这表明WFS必须生成结果集的GML文档，该文档应与OpenGIS地理置标语言实现规范一致版本2.1.1 [2]。另外，由DescribeFeatureType操作生成的输出必须是经过GML应用模式验证有效的。一个WEB要素服务（网络要素服务）应该可以支持其它

```

        </xsd:documentation>
    </xsd:annotation>
<xsd:sequence>
    <xsd:element ref="wfs:Query" maxOccurs="unbounded"/>
</xsd:sequence>
<xsd:attribute name="version"
                type="xsd:string" use="required" fixed="1.0.0"/>
<xsd:attribute name="service"
                type="xsd:string" use="required" fixed="WFS"/>

```

```

<xsd:attribute name="handle"
                type="xsd:string" use="optional"/>
<xsd:attribute name="outputFormat"
                type="xsd:string" use="optional" default="GML2">
  <xsd:annotation>
    <xsd:documentation>

        </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="maxFeatures" type="xsd:positiveInteger"
                use="optional">
    <xsd:annotation>
      <xsd:documentation>
        maxFeatures属性用来限制GetFeature请求取得的要素数，一但MaxFeature限制
        达到了，获取要素操作必须在这点返回相应的结果（而不管该查询实际的命中数）。
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>
<xsd:element name="Query" type="wfs:QueryType">
  <xsd:annotation>
    <xsd:documentation>

        </xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:complexType name="QueryType">
    <xsd:annotation>
      <xsd:documentation>
        本查询元素是查询类型中的一种。
      </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element ref="ogc:PropertyName" minOccurs="0" maxOccurs="unbounded">
        <xsd:annotation>
          <xsd:documentation>

              </xsd:documentation>
            </xsd:annotation>
          </xsd:element>
          <xsd:element ref="ogc:Filter" minOccurs="0" maxOccurs="1">
            <xsd:annotation>
              <xsd:documentation>

```

```

        </xsd:documentation>
    </xsd:annotation>
</xsd:element>
</xsd:sequence>
<xsd:attribute name="handle"
                type="xsd:string" use="optional"/>
<xsd:attribute name="typeName"
                type="xsd:QName" use="required"/>
<xsd:attribute name="featureVersion"
                type="xsd:string" use="optional">
    <xsd:annotation>
        <xsd:documentation>

            </xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
</xsd:complexType>
<!-- RESPONSE TYPE -->
<xsd:complexType name="FeatureCollectionType">
    <xsd:annotation>
        <xsd:documentation>

            </xsd:documentation>
        </xsd:annotation>
    <xsd:complexContent>
        <xsd:extension base="gml:AbstractFeatureCollectionType">
            <xsd:attribute name="lockId" type="xsd:string" use="optional">
                <xsd:annotation>
                    <xsd:documentation>

                        </xsd:documentation>
                    </xsd:annotation>
                </xsd:attribute>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
</xsd:schema>

```

A.4 WFS-transaction.xsd

```

<?xml version="1.0"?>
<xsd:schema
    targetNamespace="http://www.opengis.net/wfs"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"

```

```

xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc"
xmlns:wfs="http://www.opengis.net/wfs"
elementFormDefault="qualified">

```

```

<!-- =====
      Includes and Imports
      ===== -->
<xsd:include schemaLocation="WFS-basic.xsd"/>
<xsd:import namespace="http://www.opengis.net/gml"
            schemaLocation="../../../gml/2.1/feature.xsd"/>
<xsd:import namespace="http://www.opengis.net/ogc"
            schemaLocation="../../../filter/1.0.0/filter.xsd"/>

<!-- =====
      REQUEST MESSAGES
      ===== -->
<xsd:element name="GetFeatureWithLock" type="wfs:GetFeatureWithLockType">
  <xsd:annotation>
    <xsd:documentation>
      这是带锁获取要素（GetFeatureWithLock）请求的根元素。
      带锁获取要素操作与获取要素（GetFeature）请求相同，除了带锁获取要素
      （GetFeatureWithLock）请求锁定所有的、在结果数据集中的要素实例和在响应中返回一个锁定的客户
      端的应用标识符。
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="LockFeature" type="wfs:LockFeatureType">
  <xsd:annotation>
    <xsd:documentation>
      这是锁定要素（LockFeature）请求的根元素。
      该锁定要素（LockFeature）请求用于锁定一或者多个要素实例。
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="Transaction" type="wfs:TransactionType">
  <xsd:annotation>
    <xsd:documentation>
      这是处理（Transaction）请求的根元素。一个处理（Transaction）请求通过插入、
      更新和删除操作完成创建、修改或移除要素实例。
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>

```

```

<!-- =====
      RESPONSE MESSAGES
      ===== -->
<xsd:element name="WFS_LockFeatureResponse"
             type="wfs:WFS_LockFeatureResponseType">
  <xsd:annotation>
    <xsd:documentation>

      </xsd:documentation>
    </xsd:annotation>
  </xsd:element>
<xsd:element name="WFS_TransactionResponse"
             type="wfs:WFS_TransactionResponseType">
  <xsd:annotation>
    <xsd:documentation>

      </xsd:documentation>
    </xsd:annotation>
  </xsd:element>

<!-- =====
      COMMON ATTRIBUTE DOCUMENTATION
      =====
-->

<!-- =====
      TYPES
      ===== -->
<!-- GETFEATUREWITHLOCK -->
<xsd:complexType name="GetFeatureWithLockType">
  <xsd:annotation>
    <xsd:documentation>

      </xsd:documentation>
    </xsd:annotation>
  <xsd:sequence>
    <xsd:element ref="wfs:Query" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="version"
                 type="xsd:string" use="required" fixed="1.0.0"/>
  <xsd:attribute name="service"
                 type="xsd:string" use="required" fixed="WFS"/>
  <xsd:attribute name="handle"
                 type="xsd:string" use="optional"/>

```

```

<xsd:attribute name="expiry"
                type="xsd:positiveInteger" use="optional"/>
<xsd:attribute name="outputFormat"
                type="xsd:string" use="optional" default="GML2"/>
<xsd:attribute name="maxFeatures"
                type="xsd:positiveInteger" use="optional"/>
</xsd:complexType>
<!-- LOCKFEATURE -->
<xsd:complexType name="LockFeatureType">
  <xsd:annotation>
    <xsd:documentation>

    </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="Lock" type="wfs:LockType" maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation>

        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
  <xsd:attribute name="version"
                  type="xsd:string" use="required" fixed="1.0.0"/>
  <xsd:attribute name="service"
                  type="xsd:string" use="required" fixed="WFS"/>
  <xsd:attribute name="expiry"
                  type="xsd:positiveInteger" use="optional"/>
  <xsd:attribute name="lockAction"
                  type="wfs:AllSomeType" use="optional">
    <xsd:annotation>
      <xsd:documentation>

      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>
<xsd:complexType name="LockType">
  <xsd:annotation>
    <xsd:documentation>

    </xsd:documentation>
  </xsd:annotation>

```

```

    <xsd:sequence>
      <xsd:element ref="ogc:Filter" minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attribute name="handle"
      type="xsd:string" use="optional"/>
    <xsd:attribute name="typeName"
      type="xsd:QName" use="required"/>
  </xsd:complexType>
  <!-- TRANSACTION -->
  <xsd:complexType name="TransactionType">
    <xsd:annotation>
      <xsd:documentation>

        </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element ref="wfs:LockId" minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>

            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
          <xsd:element ref="wfs:Insert"/>
          <xsd:element ref="wfs:Update"/>
          <xsd:element ref="wfs:Delete"/>
          <xsd:element ref="wfs:Native"/>
        </xsd:choice>
      </xsd:sequence>
      <xsd:attribute name="version"
        type="xsd:string" use="required" fixed="1.0.0"/>
      <xsd:attribute name="service"
        type="xsd:string" use="required" fixed="WFS"/>
      <xsd:attribute name="handle"
        type="xsd:string" use="optional"/>
      <xsd:attribute name="releaseAction"
        type="wfs:AllSomeType" use="optional">
        <xsd:annotation>
          <xsd:documentation>

            </xsd:documentation>
          </xsd:annotation>
        </xsd:attribute>

```

```

</xsd:complexType>
<xsd:element name="LockId" type="xsd:string">
  <xsd:annotation>
    <xsd:documentation>

    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="Insert" type="wfs:InsertElementType">
  <xsd:annotation>
    <xsd:documentation>

    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:complexType name="InsertElementType">
  <xsd:sequence>
    <xsd:element ref="gml:_Feature" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:attribute name="handle" type="xsd:string" use="optional" />
</xsd:complexType>
<xsd:element name="Update" type="wfs:UpdateElementType">
  <xsd:annotation>
    <xsd:documentation>

    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:complexType name="UpdateElementType">
  <xsd:sequence>
    <xsd:element ref="wfs:Property" maxOccurs="unbounded" />
    <xsd:element ref="ogc:Filter" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>

        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
  <xsd:attribute name="handle" type="xsd:string" use="optional" />
  <xsd:attribute name="typeName" type="xsd:QName" use="required" />
</xsd:complexType>
<xsd:element name="Delete" type="wfs:DeleteElementType">
  <xsd:annotation>

```

```

        <xsd:documentation>

        </xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:complexType name="DeleteElementType">
    <xsd:sequence>
        <xsd:element ref="ogc:Filter" minOccurs="1" maxOccurs="1">
            <xsd:annotation>
                <xsd:documentation>

                </xsd:documentation>
            </xsd:annotation>
        </xsd:element>
    </xsd:sequence>
    <xsd:attribute name="handle" type="xsd:string" use="optional"/>
    <xsd:attribute name="typeName" type="xsd:QName" use="required"/>
</xsd:complexType>
<xsd:element name="Native" type="wfs:NativeType">
    <xsd:annotation>
        <xsd:documentation>

        </xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:complexType name="NativeType">
    <xsd:attribute name="vendorId" type="xsd:string" use="required">
        <xsd:annotation>
            <xsd:documentation>

            </xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="safeToIgnore" type="xsd:boolean" use="required">
        <xsd:annotation>
            <xsd:documentation>

            </xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
</xsd:complexType>

<!-- define structure to specify a property value -->
<xsd:element name="Property" type="wfs:PropertyType">

```

```

<xsd:annotation>
  <xsd:documentation>

    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:complexType name="PropertyType">
  <xsd:sequence>
    <xsd:element name="Name" type="xsd:string">
      <xsd:annotation>
        <xsd:documentation>

          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    <xsd:element name="Value" minOccurs="0">
      <xsd:annotation>
        <xsd:documentation>

          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
<!-- RESPONSE TYPES -->
<xsd:complexType name="WFS_LockFeatureResponseType">
  <xsd:annotation>
    <xsd:documentation>

      </xsd:documentation>
    </xsd:annotation>
  <xsd:sequence>
    <xsd:element ref="wfs:LockId">
      <xsd:annotation>
        <xsd:documentation>

          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    <xsd:element name="FeaturesLocked"
      type="wfs:FeaturesLockedType" minOccurs="0">
      <xsd:annotation>
        <xsd:documentation>

```

```

        </xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="FeaturesNotLocked"
    type="wfs:FeaturesNotLockedType" minOccurs="0">
    <xsd:annotation>
        <xsd:documentation>

            </xsd:documentation>
        </xsd:annotation>
    </xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="FeaturesLockedType">
    <xsd:sequence maxOccurs="unbounded">
        <xsd:element ref="ogc:FeatureId"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="FeaturesNotLockedType">
    <xsd:sequence maxOccurs="unbounded">
        <xsd:element ref="ogc:FeatureId"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="WFS_TransactionResponseType">
    <xsd:annotation>
        <xsd:documentation>

            </xsd:documentation>
        </xsd:annotation>
    <xsd:sequence>
        <xsd:element name="InsertResult"
            type="wfs:InsertResultType"
            minOccurs="0" maxOccurs="unbounded">
            <xsd:annotation>
                <xsd:documentation>

                    </xsd:documentation>
                </xsd:annotation>
            </xsd:element>
        <xsd:element name="TransactionResult"
            type="wfs:TransactionResultType">
            <xsd:annotation>
                <xsd:documentation>

```

```

        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
  <xsd:attribute name="version"
    type="xsd:string" use="required" fixed="1.0.0"/>
</xsd:complexType>
<xsd:complexType name="TransactionResultType">
  <xsd:sequence>
    <xsd:element name="Status" type="wfs:StatusType">
      <xsd:annotation>
        <xsd:documentation>

          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="Locator" type="xsd:string" minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>

            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="Message" type="xsd:string" minOccurs="0">
          <xsd:annotation>
            <xsd:documentation>

              </xsd:documentation>
            </xsd:annotation>
          </xsd:element>
          <xsd:sequence>
            <xsd:attribute name="handle" type="xsd:string" use="optional"/>
          </xsd:complexType>
        <xsd:complexType name="InsertResultType">
          <xsd:sequence>
            <xsd:element ref="ogc:FeatureId" maxOccurs="unbounded"/>
          </xsd:sequence>
          <xsd:attribute name="handle" type="xsd:string" use="optional"/>
        </xsd:complexType>
      <xsd:complexType name="StatusType">
        <xsd:choice>
          <xsd:element ref="wfs:SUCCESS"/>
          <xsd:element ref="wfs:FAILED"/>
          <xsd:element ref="wfs:PARTIAL"/>
        </xsd:choice>
      </xsd:complexType>
    </xsd:sequence>
  </xsd:complexType>

```

```

    </xsd:choice>
</xsd:complexType>
<xsd:element name="SUCCESS" type="wfs:EmptyType"/>
<xsd:element name="FAILED" type="wfs:EmptyType"/>
<xsd:element name="PARTIAL" type="wfs:EmptyType"/>
<!-- MISC TYPES -->
<xsd:complexType name="EmptyType"/>
<xsd:simpleType name="AllSomeType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="ALL"/>
        <xsd:enumeration value="SOME"/>
    </xsd:restriction>
</xsd:simpleType>
</xsd:schema>

```

A.5 WFS-capabilities.xsd

```

<?xml version="1.0" ?>
<xsd:schema
    targetNamespace="http://www.opengis.net/wfs"
    xmlns:wfs="http://www.opengis.net/wfs"
    xmlns:gml="http://www.opengis.net/gml"
    xmlns:ogc="http://www.opengis.net/ogc"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified">

    <!-- =====
        Imports ...
        ===== -->
    <xsd:import namespace="http://www.opengis.net/ogc"
        schemaLocation="../../filter/1.0.0/filterCapabilities.xsd" />

    <!-- =====
        Global elements and attributes
        ===== -->

    <xsd:element name="Abstract" type="xsd:string"/>

    <xsd:element name="AccessConstraints" type="xsd:string"/>
    <xsd:element name="Fees" type="xsd:string"/>

    <xsd:element name="Keywords" type="xsd:string"/>

    <xsd:element name="OnlineResource"/>
    <xsd:element name="SRS" type="xsd:string"/>

```

```

<xsd:element name="Title" type="xsd:string"/>

<xsd:element name="Query" type="wfs:EmptyType"/>
<xsd:element name="Insert" type="wfs:EmptyType"/>
<xsd:element name="Update" type="wfs:EmptyType"/>
<xsd:element name="Delete" type="wfs:EmptyType"/>
<xsd:element name="Lock" type="wfs:EmptyType"/>

<!-- REDEFINE THIS ELEMENT AS NEEDED IN YOUR XML -->
<xsd:element name="VendorSpecificCapabilities" type="xsd:string"/>

<!-- =====
      Root element
      ===== -->

<xsd:element name="WFS_Capabilities" type="wfs:WFS_CapabilitiesType"/>

<!-- =====
      Types
      ===== -->
<xsd:complexType name="WFS_CapabilitiesType">
  <xsd:sequence>
    <!-- The Service element provides metadata for
          the service as a whole. -->
    <xsd:element name="Service" type="wfs:ServiceType"/>

    <xsd:element name="Capability" type="wfs:CapabilityType"/>
    <xsd:element name="FeatureTypeList" type="wfs:FeatureTypeListType"/>
    <xsd:element ref="ogc:Filter_Capabilities" />
  </xsd:sequence>

  <xsd:attribute name="version"
    type="xsd:string" fixed="1.0.0"/>

  <xsd:attribute name="updateSequence"
    type="xsd:nonNegativeInteger" default="0"/>
</xsd:complexType>

<xsd:complexType name="ServiceType">
  <xsd:sequence>

```

```

    <xsd:element name="Name" type="xsd:string"/>
    <xsd:element ref="wfs:Title"/>
    <xsd:element ref="wfs:Abstract" minOccurs="0"/>
    <xsd:element ref="wfs:Keywords" minOccurs="0"/>
    <xsd:element ref="wfs:OnlineResource"/>
    <xsd:element ref="wfs:Fees" minOccurs="0"/>
    <xsd:element ref="wfs:AccessConstraints" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CapabilityType">
  <xsd:sequence>
    <xsd:element name="Request" type="wfs:RequestType"/>

    <xsd:element ref="wfs:VendorSpecificCapabilities" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="FeatureTypeListType">
  <xsd:sequence>
    <xsd:element name="Operations"
      type="wfs:OperationsType" minOccurs="0"/>
    <xsd:element name="FeatureType"
      type="wfs:FeatureTypeType" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="RequestType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="GetCapabilities"
      type="wfs:GetCapabilitiesType"/>
    <xsd:element name="DescribeFeatureType"
      type="wfs:DescribeFeatureTypeType"/>
    <xsd:element name="Transaction"
      type="wfs:TransactionType"/>
    <xsd:element name="GetFeature"
      type="wfs:GetFeatureTypeType"/>
    <xsd:element name="GetFeatureWithLock"
      type="wfs:GetFeatureTypeType"/>
    <xsd:element name="LockFeature"
      type="wfs:LockFeatureTypeType"/>
  </xsd:choice>
</xsd:complexType>
<xsd:complexType name="GetCapabilitiesType">

```

```

<xsd:sequence>
  <xsd:element name="DCPType"
    type="wfs:DCPTypeType" maxOccurs="unbounded" />
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="DescribeFeatureTypeType">
  <xsd:sequence>
    <xsd:element name="SchemaDescriptionLanguage"
      type="wfs:SchemaDescriptionLanguageType" />
    <xsd:element name="DCPType"
      type="wfs:DCPTypeType" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="TransactionType">
  <xsd:sequence>
    <xsd:element name="DCPType"
      type="wfs:DCPTypeType" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="GetFeatureTypeType">
  <xsd:sequence>
    <xsd:element name="ResultFormat"
      type="wfs:ResultFormatType" />
    <xsd:element name="DCPType"
      type="wfs:DCPTypeType" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="LockFeatureTypeType">
  <xsd:sequence>
    <xsd:element name="DCPType"
      type="wfs:DCPTypeType" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="DCPTypeType">
  <xsd:sequence>
    <xsd:element name="HTTP" type="wfs:HTTPType" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="FeatureTypeType">
  <xsd:sequence>
    <xsd:element name="Name" type="xsd:QName" />
    <xsd:element ref="wfs:Title" minOccurs="0" />

```

```

    <xsd:element ref="wfs:Abstract" minOccurs="0"/>
    <xsd:element ref="wfs:Keywords" minOccurs="0"/>
    <xsd:element ref="wfs:SRS"/>
    <xsd:element name="Operations"
        type="wfs:OperationsType" minOccurs="0"/>
    <xsd:element name="LatLongBoundingBox"
        type="wfs:LatLongBoundingBoxType"
        minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="MetadataURL"
        type="wfs:MetadataURLType"
        minOccurs="0" maxOccurs="unbounded"/>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="GetType">
    <xsd:attribute name="onlineResource" type="xsd:string" use="required"/>
</xsd:complexType>

<!-- Available HTTP request methods. -->
<xsd:complexType name="HTTPType">
    <xsd:choice maxOccurs="unbounded">

        <xsd:element name="Get" type="wfs:GetType"/>
        <xsd:element name="Post" type="wfs:PostType"/>
    </xsd:choice>
</xsd:complexType>

<xsd:complexType name="LatLongBoundingBoxType">
    <xsd:attribute name="minx" type="xsd:string" use="required"/>
    <xsd:attribute name="miny" type="xsd:string" use="required"/>
    <xsd:attribute name="maxx" type="xsd:string" use="required"/>
    <xsd:attribute name="maxy" type="xsd:string" use="required"/>
</xsd:complexType>

<xsd:complexType name="MetadataURLType">
    <xsd:simpleContent>
        <xsd:extension base="xsd:string">
            <xsd:attribute name="type" use="required">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:NMTOKEN">
                        <xsd:enumeration value="TC211"/>
                        <xsd:enumeration value="FGDC"/>
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:attribute>
        </xsd:extension>
    </xsd:simpleContent>

```

```

        </xsd:simpleType>
      </xsd:attribute>
      <xsd:attribute name="format" use="required">
        <xsd:simpleType>
          <xsd:restriction base="xsd:NMTOKEN">
            <xsd:enumeration value="XML" />
            <xsd:enumeration value="SGML" />
            <xsd:enumeration value="TXT" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:attribute>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="OperationsType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element ref="wfs:Insert" />
    <xsd:element ref="wfs:Update" />
    <xsd:element ref="wfs:Delete" />
    <xsd:element ref="wfs:Query" />
    <xsd:element ref="wfs:Lock" />
  </xsd:choice>
</xsd:complexType>
<xsd:complexType name="PostType">
  <xsd:attribute name="onlineResource" type="xsd:string" use="required" />
</xsd:complexType>

<xsd:element name="GML2" type="wfs:EmptyType" />
<xsd:complexType name="ResultFormatType">
  <xsd:sequence maxOccurs="unbounded">
    <xsd:element ref="wfs:GML2" />
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="XMLSCHEMA" type="wfs:EmptyType" />
<xsd:complexType name="SchemaDescriptionLanguageType">
  <xsd:sequence maxOccurs="unbounded">
    <xsd:element ref="wfs:XMLSCHEMA" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="EmptyType" />
</xsd:schema>

```

附 录 B

一致性测试（规范性附录）

WEB要素服务（网络要素服务）的明确的一致性检测没有完全确定，将根据需要加入这个文档的未来版本。

同时，WFS执行必须满足下列系统特征以符合这个文档规范。

必须支持获得能力，描述要素类型和获得要素操作。

一个获得能力请求返回的扩展标记语言（XML）文档，必须与附录A4中的XML模式定义一致。这些可以采用普通XML验证工具进行确认。

获得要素操作响应，WFS必须能够生成GML[2]作为输出格式中的一个。

获得要素请求返回的扩展标记语言必须对描述要素类型请求响应生成的模式有效。这些可以采用常见的XML验证工具进行确认。

在这篇文档标准化部分的所有条款，满足关键字“必须”，“不必”，“必需的”，“将会”和“将不会”。
