

北京大学

硕士研究生学位论文

Semantic Web 中空间信息
互操作框架研究

An Open Spatial Information Interoperation
Framework in the Semantic Web

姓 名：丁 力

系 别：计算机科学技术系

专 业：计算机软件与理论

研究方向：空间信息系统与智能辅助决策

导 师：许卓群 教授

二零零一年五月

声 明

任何收存和保管本论文各种版本的单位和个人, 未经本论文作者授权, 不得将本论文转借他人并复印、抄录、拍照、或以任何方式传播。否则, 引起有碍作者著作权益之问题, 将可能承担法律责任。

我有一个关于 WWW 的梦想..... 它由两个部分组成。

在第一部分中，WWW 变成了一个有利于人们相互合作的强大工具。我总是把信息空间想象为某种每个人都可以直接凭直觉获得的东西，人们不仅可以浏览，而且可以创造它.....

在梦想的第二部分，合作延伸到了计算机。机器变得可以分析 WWW 上的所有数据——包括内容、链接、以及人与计算机之间的交易。能够实现这一切的一种“Semantic Web”尚未出现，但一旦它出现，每天的交易、行政活动，以及我们的日常生活就将通过机器之间的对话来完成，而人类只要提供灵感和直觉就行了.....

—— Tim Berners-Lee, WWW 的发明者

摘 要

Internet 革命性地把用户与分布在世界各地、以不同形式存在的信息连接起来。可是, Internet 上的空间信息资源仍然面向特定行业, 依赖于特定的支撑环境, 形成了分布且异构的空间信息孤岛: 各自独立、相对封闭、无法互相沟通和协作。

如何让这些空间信息孤岛既相对独立又能互操作; 如何让 Internet 用户快速准确地定位、获取并使用他们所关心的空间信息……这是目前国际上空间信息技术研究的焦点问题, 也是“数字地球”和“空间信息基础设施”项目的关键技术之一。

为此, 论文提出了开放空间信息互操作框架 (Open Spatial Information Interoperation Framework, OSIIF)。论文采用 Semantic Web 倡导的 XML 和 RDF 技术, 以开放动态为首要原则, 重点研究了 OSIIF 的体系结构、信息模型和服务模型。

OSIIF 体系结构的关键就是把信息和服务统一看作 Internet 上的资源。通过划分资源和识别资源之间的关系来实现空间信息互操作。同时使用元数据描述资源特性。

OSIIF 信息模型通过提取空间信息领域的共识, 采用 XML 作为空间信息的表示方法和交换格式, 以 RDF 表达空间信息的语义, 着重解决应用程序之间理解和交换空间信息的问题。

OSIIF 的服务模型提供一种 Internet 上面面向大粒度服务的分布计算平台, 以此提高应用系统的构造效率。同时 OSIIF 也提供了“服务元数据”以管理并使用分布在 Internet 上的众多服务资源, 它可以唯一地标识服务、描述服务的能力以及服务的接口。另外, OSIIF 提供了导航服务, 可以帮助用户查找资源, 支持动态加入资源、动态使用服务等; OSIIF 也提供了协调服务, 以任务脚本的形式为接口向用户提供服务。提出了一种具有较强表达能力的抽象任务脚本——信息流动图。

为了说明 OSIIF 的可实现性, 论文也列举了一些具体例子来说明如何建设和使用 OSIIF。

最后对论文工作进行了总结和展望, 探讨了 OSIIF 设计中尚需深入研究的地方, 如网络安全问题, 也指明了 OSIIF 的广阔的应用前途——不但为网上空间信息的管理和导航提供有效的方法, 对网上其他信息的管理和导航也有重要的借鉴意义。

关键词: 开放空间信息互操作框架 (OSIIF), Semantic Web,
RDF, XML, 空间信息, 互操作, 空间元数据,
服务元数据, 信息流动图

目 录

第 1 章 简介与动机	1
1.1. 动机.....	1
1.2. Internet 用户离空间信息有多远	1
1.3. 空间信息互操作中的重要问题.....	2
1.3.1 可达性问题 (Accessibility)	3
1.3.2 可用性问题 (Interoperability)	3
1.3.3 易用性问题 (User-friendly)	4
1.4. 相关工作.....	4
1.4.1 信息互操作框架	5
1.4.1.1 Semantic Web	5
1.4.1.2 Digital Object 服务框架	5
1.4.1.3 OGC 的工作.....	5
1.4.1.4 TC211 的工作	5
1.4.2 GIS 实现技术简述	6
1.5. 文章结构与主要贡献.....	7
1.5.1 文章结构.....	7
1.5.2 主要贡献.....	8
第 2 章 开放空间信息互操作框架	9
2.1. 目标.....	9
2.2. OSIIF 的设计原则.....	9
2.2.1 开放性	9
2.2.2 动态性	9
2.2.3 自动性	9
2.2.4 简单性	9
2.2.5 灵活性	10
2.3. OSIIF 的体系结构.....	10
2.3.1 OSIIF 体系结构图.....	10
2.3.2 OSIIF 的信息.....	11
2.3.3 服务	12
2.3.4 Internet 用户	12
2.3.5 OSIIF 中的基本关系.....	13
2.4. 人与 OSIIF 的关系.....	14
2.5. OSIIF 与 Semantic Web 的关系.....	14
2.6. 本章小结.....	15
第 3 章 基于共识的分级空间信息模型	16
3.1. 目标.....	16
3.2. 知识表示.....	16
3.2.1 计算机的知识表示方法	16
3.2.2 XML 与 RDF	16
3.2.2.1 XML 协议集	16

3.2.2.2	XMLDTD 与 XML Schema.....	17
3.2.2.3	RDF 与 RDF Schema.....	18
3.2.2.4	XML 与 RDF	19
3.2.3	计算机的知识表示层次.....	19
3.2.3.1	计算机内部的知识表示.....	20
3.2.3.2	计算机内部的知识表示间的映射.....	21
3.3.	共识与 Ontology	21
3.3.1	概念模型级共识.....	21
3.3.2	领域级共识.....	21
3.3.3	应用级共识.....	21
3.3.4	Ontology	22
3.4.	基于共识的分级空间信息模型.....	22
3.4.1	OSIIF 的空间信息共识.....	22
3.4.2	OSIIF 的空间信息交换机制.....	23
3.4.2.1	两种信息翻译模式.....	23
3.4.2.2	OSIIF 的信息分级交换模型.....	23
3.4.3	空间信息交换的分级共识矩阵.....	24
3.5.	创建空间信息产品的规程.....	26
3.6.	本章小结.....	26
第 4 章	面向 Internet 计算的服务模型	27
4.1.	OSIIF 服务模型.....	27
4.1.1	服务的层次.....	28
4.1.2	OSIIF 中服务的分类.....	29
4.1.2.1	协调服务 (Coordination Service)	29
4.1.2.2	服务导航服务 (Service Navigation Service)	29
4.1.2.3	信息导航服务 (Information Navigation Service)	30
4.1.2.4	空间信息获取服务 (Spatial Information Retrieval Service)	30
4.1.2.5	空间信息处理服务 (Spatial Information Process Service)	31
4.1.3	OSIIF 服务模型的特点.....	31
4.1.3.1	支持加入空间信息产品.....	31
4.1.3.2	支持加入服务资源.....	31
4.1.3.3	支持动态访问服务资源.....	31
4.1.3.4	支持动态获取空间信息产品.....	32
4.1.3.5	支持任务求解的多级别的服务.....	33
4.1.3.6	支持任务求解的多路径选择.....	33
4.2.	OSIIF 的服务元数据.....	33
4.2.1	设计思想.....	33
4.2.2	服务标识.....	34
4.2.3	服务能力描述.....	35
4.2.3.1	服务分类.....	35
4.2.3.2	输入/输出参数.....	35
4.2.4	服务接口描述.....	36
4.2.4.1	服务的通信接口.....	36

4.2.4.2 生成 HTTP 消息	37
4.2.5 服务元数据的应用举例	38
4.2.5.1 查询图层获取服务	38
4.2.5.2 支持服务之间的通信参数匹配	39
4.3. OSIIF 的服务过程	39
4.3.1 协调服务的脚本分类	39
4.3.1.1 任务说明书	39
4.3.1.2 信息流动图	40
4.3.2 任务说明书的辅助生成	41
4.3.2.1 生成任务说明书	41
4.3.2.2 生成信息流动图	41
4.4. 本章小结	42
第 5 章 用 OSIIF 构建空间信息领域服务体系	43
5.1. 增加空间信息产品实例——注册图层	43
5.2. 增加空间信息产品类型——地图	45
5.2.1 明确问题域——定义地图	45
5.2.2 提取共识——将地图纳入空间信息模型	45
5.2.3 设计地图的对象系统	47
5.2.4 设计地图的元数据	48
5.3. 增加服务资源	48
5.3.1 直接获取空间信息产品	48
5.3.2 动态获取空间信息产品	48
5.3.2.1 设计 GeoUnionDB 图层获取服务的元数据	48
5.3.2.2 设计图层获取服务	50
5.4. 一个框架服务的应用——获取图层	51
5.4.1.1 初始状态	52
5.4.1.2 查找信息	52
5.4.1.3 生成信息流动图	53
5.4.1.4 执行服务过程	53
5.5. 本章小结	54
第 6 章 结论与展望	55
6.1. 结论	55
6.2. 进一步的工作	55
参考文献	57
附录一: GML2.0 中对象类的类图	61
附录二: Map.xsd	63
硕士期间发表的论文	67
致谢	68

图目录

图 1-1 Internet 用户与空间信息资源之间的鸿沟	2
图 1-2 文章结构	7
图 2-1 OSIIF 的体系结构	10
图 2-2 OSIIF 中的基本关系	13
图 2-3 人与 OSIIF 的关系	14
图 3-1 XML 文本及三种文档定义方法	17
图 3-2 计算机中的知识表示的映射	20
图 3-4 OSIIF 的空间信息共识	22
图 3-3 OSIIF 的分级信息交换机制	24
图 3-5 Simple Feature Model 的 RDF 表示	25
图 4-1 OSIIF 的服务模型	27
图 4-2 服务的层次划分及 OSIIF 服务所处的层次	28
图 4-3 OSIIF 的服务元数据的派生关系 (RDF 图)	34
图 4-4 OSIIF 服务元数据的组成要素	34
图 4-5 服务标识作为基本服务元数据的属性 (RDF 图)	35
图 4-6 服务分类相关的服务元数据 (RDF 图)	35
图 4-7 输入/输出参数作为基本服务元数据的属性 (RDF 图)	36
图 4-8 服务接口描述作为基本服务元数据的属性 (RDF 图)	36
图 4-9 服务的通信接口	36
图 4-10 HTTP 服务接口描述的属性 (RDF 图)	37
图 4-11 根据不同的操作特性映射服务	38
图 4-12 描述中空间信息规格的空间元数据	38
图 4-13 OSIIF 的协调服务支持两种服务脚本	39
图 4-14 信息流动图	41
图 5-1 支持图层获取自动化的图层元数据 (RDF 图)	43
图 5-2 OSIIF 的空间信息共识及 mapping 应用中的共识	45
图 5-3 地图的 Xlink 定义	46
图 5-4 地图的抽象对象体系 (概念级表示)	46
图 5-5 地图的对象级表示	47
图 5-6 地图定义文件与 GML2.0 定义文件的依赖关系	47
图 5-7 图层获取中相关的服务元数据 (RDF 图)	49
图 5-8 服务脚本规格定义举例 (RDF 图)	50
图 5-9 Geo-Union5.0 的客户端空间信息访问构件结构	51
图 5-10 Internet 用户掌握的资源	52
图 5-11 协调服务掌握的资源	52
图 5-12 协调服务查找信息	52
图 5-13 本例的信息流动图	53
图 5-14 信息流动图的具体执行	53

表目录

表 1-1 通信角度的 GIS 分类.....	6
表 1-2 数据存取角度的 GIS 分类 ^[1.36]	6
表 1-3 系统构造技术角度的 GIS 分类.....	6
表 3-1 XML DTD 和 XML Schema 的关系.....	18
表 3-2 RDF Schema 对 class 的特殊定义.....	18
表 3-3 RDF 与 XML 的关系 ^[3.13]	19
表 3-4 两种常见翻译模式的对比.....	23
表 3-5 空间信息交换的分级共识矩阵.....	24
表 4-1 OSIIF 中服务分类.....	29
表 4-2 HTTP 消息的格式.....	37
表 4-3 一个任务说明书的例子.....	40
表 4-4 信息流动图中的节点.....	40
表 4-5 信息流动图中的弧.....	40
表 4-6 信息流动图的 Backus-Naur 范式定义.....	41
表 5-1 图层访问路径特有属性的 BNF 描述.....	43
表 5-2 一个以 GML2 (XML Schema) 形式存储的图层的部分元数据.....	44
表 5-3 一个存储在 Geo-Union 5.0 data server 中的图层的部分元数据.....	44
表 5-4 一个图层获取服务的元数据举例.....	49
表 5-5 发送给信息获取服务的服务层消息.....	54

第1章 简介与动机

1.1. 动机

从 80 年代第一个地理信息系统（Geographical Information System, GIS）商业化产品诞生开始, GIS 逐渐形成了一个重要的计算机应用产业。它的发展与计算机技术的进展紧密结合。尤其是随着以 Internet 为代表的网络技术的迅猛进步, 使 GIS 应用飞跃到了网络时代。

虽然网络上的空间信息资源在不断增长, 但由于行业管理和数据安全的原因, 这些空间信息资源大多是面向行业的、依赖于特定的支撑环境和运行环境。他们各自独立、相对封闭、无法互相沟通和协作——形成了空间信息孤岛, 难以满足 Internet 上空间信息相关的综合决策的需要。于是, Internet 用户至今还不得不面对这样的尴尬局面: 难以在 Internet 上找到需要的地图; 即使找到一个地图, 还因为没有安装相应的浏览工具而无法浏览它; 程序员也难以在 Internet 上构造一个可以使用异构空间信息资源的应用系统。

因此, 在 Internet 上共享空间信息资源的呼声与日俱增: 1998 年 1 月美国副总统提出了“数字地球”的概念^[1.1]; 1999 年 12 月第一届数字地球国际论坛的“数字地球北京宣言”提出了携手共建数字家园的倡议^[1.2]。而“网上异构空间信息的互操作与导航”问题也成为了目前国际上空间信息技术研究的焦点和热点问题, 还是“数字地球”和“空间信息基础设施”项目的关键技术之一。

2001 年三月, W3C 提出了 Semantic Web 技术, 倡导在 Internet 上共享并使用包含语义的信息, 支持机器对信息的理解, 为软件的互操作提供服务^[1.3]。OGC (Open GIS Consortium) 也提出了关于空间信息 Web 服务的技术倡议^[1.4]。可以看出, Semantic Web 技术拥有强大的信息表达和信息交换能力, 加之 OGC 倡导的面向服务的系统构造手段, 这对解决“网上异构空间信息的互操作与导航”问题有很大的帮助。

论文依据上述两项技术, 提出由空间信息和相关服务组成空间信息互操作框架, 希望通过服务把 Internet 用户和分布在网上的异构空间信息有机地连接起来, 形成 Internet 上空间信息互操作服务体系, 并以此向 Semantic Web 提供空间信息服务。

1.2. Internet 用户离空间信息有多远

使用信息的活动包含三类角色: 被使用的信息——信息资源; 处理信息的服务软件——服务资源; 信息的使用者——用户。空间信息互操作主要研究的是用户如何共享空间信息资源, 以及服务资源之间如何互操作。

OGC 认为, 空间信息 (Spatial Information) 就是描述地球上的客观事物的地理位置与地理特性的信息。它是道路、大楼、车辆、湖泊、森林等现实世界的事物的抽象表示^[1.5]。

OGC 认为, GIS 是一种专用的数据库, 其中的数据大多使用了空间位置描述信息。GIS 中包括若干子系统: 数据录入子系统; 数据存储、检索与表现子系统; 数据管理、转换与分析子系统; 数据报表与汇总子系统。通常人们把 GIS

看作信息处理手段而不是特定的信息^[1.5]。本文也采用这个观点。

与一般的信息资源不同，空间信息资源通常与特定的 GIS 产品密切相关。换言之，为了访问某个空间信息资源，用户必须使用特定的空间信息服务——GIS 产品。由于 GIS 产品在服务接口上的各异性和局限性，它很难满足 Internet 上空间信息互操作的需求，这样 Internet 用户与空间信息资源之间就出现了两大鸿沟：信息获取鸿沟和信息理解鸿沟（如图 1-1）。

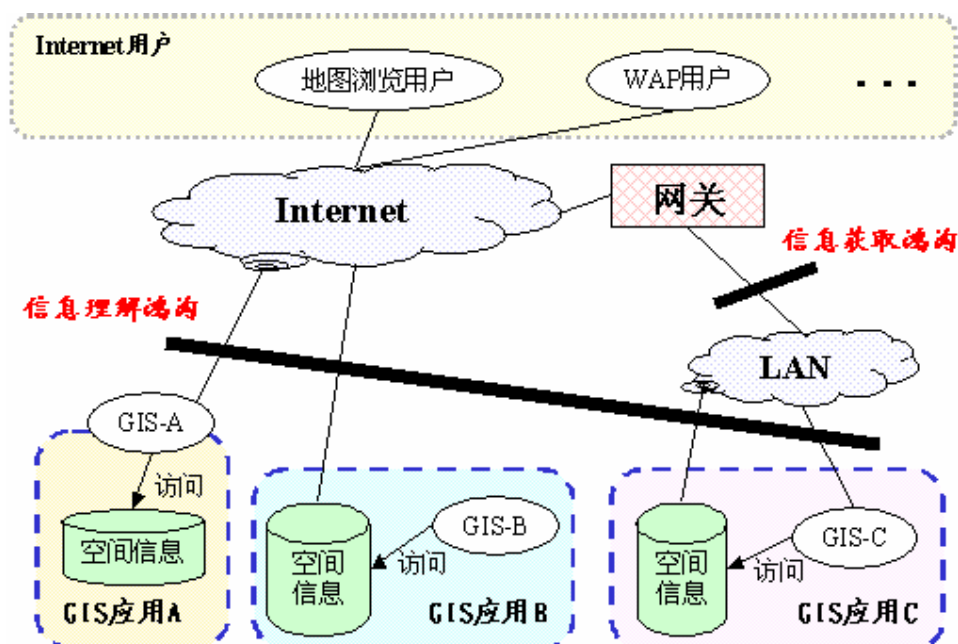


图 1-1 Internet 用户与空间信息资源之间的鸿沟

信息获取鸿沟表现为空间信息资源没有直接连接到 Internet 上。由于历史原因或出于信息管理或信息安全考虑，空间信息提供者不能或不想把空间信息直接连接到 Internet 上。例如，GIS 应用 C 中的空间信息就面临这样的问题——Internet 用户必须越过网关才能访问空间信息。

信息理解鸿沟表现为用户无法理解空间信息。这与 GIS 厂商的纵向产品线密切相关，即一个 GIS 产品会从数据管理、数据传输到数据表现全方位地向用户提供服务。而不同的 GIS 产品有不同的接口，它所管理的空间信息也是各不相同的。于是，Internet 用户只能依赖特定的 GIS 产品去访问特定的空间信息资源。例如，上图中 GIS 应用 A 中，虽然 GIS-A 能提供空间信息服务，但 Internet 用户无法理解它传来的信息的含义。又如，上图中 GIS 应用 B 中，虽然 Internet 用户可以直接访问空间信息，但它也同样面临理解的困难。

1.3. 空间信息互操作中的重要问题

Internet 用户与空间信息资源之间的鸿沟可以归纳成空间信息互操作中的三个重要问题：可达性问题，即用户难以获取空间信息；可用性问题，即用户难以理解空间信息；易用性问题，即用户难以方便地使用并处理空间信息。

1.3.1 可达性问题 (Accessibility)

空间信息通常具有比较复杂的结构,不同 GIS 产品采用不同的数据模型来描述空间信息,因此用户只能用特定的 GIS 产品访问特定的空间信息资源。不幸的是,以往大量 GIS 应用都是针对单机或局域网开发的,而目前不少 GIS 应用也会出于安全考虑而只建设在局域网或企业网中,这样就出现了大量不可在 Internet 上被直接访问的空间信息孤岛 (Isolated Spatial Information Resource, ISIR)。为了帮助 Internet 用户有效地访问 ISIR, 需要考虑以下问题:

- 设计怎样的中介软件 (Mediator) 来建立 ISIR 和 Internet 应用的连接
- Internet 用户如何了解到 ISIR 和中介软件的分布
- Internet 用户如何通过中介软件访问 ISIR
- Internet 用户以怎样的格式得到 ISIR

目前,有很多组织机构都在致力于这方面的工作。主要有两种策略,标准化组织侧重于设计 Web 上空间信息服务的统一接口规范,如 OGC^[1.6], ISO/TC211^[1.7]; 专业 GIS 厂商侧重于提供 Web 服务的实现——WebGIS, 如 ESRI 的 Internet Map Server(IMS)^[1.8], MapInfo 公司的 MapXtreme^[1.9], Intergraph 公司的 GeoMedia Web Map^[1.10]等。

1.3.2 可用性问题 (Interoperability)

不同的 GIS 厂商的纵向产品线造成了空间信息提供者与空间信息处理服务提供者的一体化的局面,而不同的 GIS 产品的边界导致了用户难以理解和使用不同空间信息资源的局面。随着空间信息在分布式应用系统 (尤其是 Internet 应用) 中的广泛使用,用户不希望受到过多的局限,迫切需要打破这种割据局面,实现空间信息互操作。

Max J.Egenhofer 在 1999 年提出“GIS 互操作的目标就是实现信息的自动处理,从而使用户有效地越过信息收集者,软件服务的设计者自有的边界,与他们共享信息与服务”^[1.11]。

ISO/TC211 给出了互操作的定义,认为若两个实体 X 和 Y 能相互操作,则 X 和 Y 对处理请求 Ri 具有共同理解,并且如果 X 向 Y 提出处理请求 Ri, Y 能对 Ri 作出正确反应,并将结果 Si 返回给 X^[1.12]。

有很多学者和机构也对 GIS 互操作进行了探讨研究:Y.A.Bishr(NCGIA)^[1.13], Buehler (OGC)^[1.5], M.F.Goodchild^[1.14], 黄裕霞 (中科院地理所)^[1.15]讨论了 GIS 互操作的层次划分;M.F.Goodchild^[1.14]探讨了 GIS 互操作的特征;黄裕霞^[1.16]综述了 GIS 互操作的概念,“Interoperating Geographic Information Systems”一书^[1.17]从理论与概念、分布环境建模、及系统开发等三个角度全面地探讨了 GIS 互操作问题。他们观点强调了 GIS 互操作中的两个重要问题:如何互操作异构的空间信息,如何互操作空间信息服务软件。

Internet 环境中空间信息异构问题可以从语法和语义两个不同的层次研究:

- 语法上的差异:不同的空间信息资源可以采用不同存储格式,而同一类存储格式也可能有版本的差异。
- 语义上的差异:不同的空间信息资源可以采用不同的概念体系表示,而

同一个概念体系中的概念也可能有同型异义或同义异型的现象。

人与人之间说话就是一个很好的一个信息互操作的例子。比如说，张三问李四今天的气温，李四听懂了，然后告诉张三今天 23 度，而张三也听懂了，这就是互操作。若是张三问 Tom 今天的气温，Tom 不懂汉语，听不懂张三问什么，自然无法回答问题，这就不算互操作。因为，虽然发生了信息的交换，但是参与者之间并不能相互理解对方传递的信息。

上面的例子主要体现了语法异构造成的理解障碍，目前在这个方面得已经有不少解决办法。主流的方法是语法翻译，即编制一个转换程序，把采用某种存储格式的空间信息转存为另一种存储格式，再交给可以理解该格式的 GIS 产品使用。XML 的出现，一方面可以缩小语法的差异；另一方面也在信息中增加了语义内容，使得消除语义差异成为可能。

Internet 环境中空间信息服务软件互操作的问题则可以参考网络分层模型中服务定义^[1.18]来考虑：

- 软件提供怎样的服务能力
- 软件提供怎样的服务接口
- 软件之间采用怎样的协议通信并协作。

空间信息处理软件的互操作需要使用机器可理解信息，Semantic Web 中提出的利用 XML+RDF 在信息中增加语义部分是解决这个问题的好一个尝试。

1.3.3 易用性问题 (User-friendly)

用户在使用空间信息时有各式各样的需求：有的希望使用众所周知的服务，为此 GIS 厂商提供了集成环境；有的则有特殊的要求，为此 GIS 厂商提供二次开发平台以支持用户构造自己的应用。在 Internet 环境中，应用系统的构造方法也从单机单任务模式发展到了多任务分布计算模式。潜在的用户市场不可能再被一两家大厂商垄断，事实上他们也很难针对特定应用提供服务。这样就需要一种开放的信息互操作框架，框架的建设者可以提供信息原子与服务原子，然后框架的使用者可以通过原子之间的组合完成任务。其中需要考虑如下问题：

- 框架中有哪些服务原子和信息原子
- 用户如何了解并使用服务原子和信息原子
- 服务原子可以采用哪些方法实现
- 服务原子与信息原子之间如何通信
- 服务原子与信息原子之间如何协同

组件技术是比较流行的空间信息应用构造手段^[1.19]，而基于 Agent 的软件技术则是一种更为新颖的解决方案^[1.20]。

1.4. 相关工作

相关工作主要体现在信息互操作框架的设计与实现，以及目前常见的 GIS 实现技术。

1.4.1 信息互操作框架

1.4.1.1 Semantic Web

WWW 的发明者 Tim Berners-Lee 早在 1999 年就提出了 Semantic Web 的概念^[1.21]。他所领导的非赢利性组织 W3C 于 2000 年创建了 www.SemanticWeb.org 网站^[1.22]。在 Semantic Web 看来, web 上的定义和关联的数据都可以被机器使用,不但可以显示,还可以在各种各样的应用中使用^[1.23]。它认为数据和软件都可以是网络上的资源,软件就是网络中一些组件的组合。它的建设分成三个步骤:通过标记语言(如 XML)定义一套术语,形成某个领域的 Ontology,这样不断地在网络中建设很多 Ontology;然后编制软件,能够根据 Ontology 之间的关系进行推理;最后基于推理软件建设应用。Tim Berners-Lee 认为^[1.24]: Semantic Web 将会给网页有意义的内容带来结构,从而为网络中的代理了用户复杂需求的 Agent 提供一个具有足够信息丰度的环境。

Semantic Web 是一种概念框架,具体实现需要由不同的领域中的专家和实体努力才能完成。它提出的是一种通用的思路,并不针对空间信息。参考这方面的因素,可以帮助框架设计者理解领域之间协作的问题。

1.4.1.2 Digital Object 服务框架

TCP/IP 协议的创始人之一 Kahn 提出了 Digital Object 服务框架。这是一个开放的可扩展的分布式数字信息服务体系结构^[1.25]。数字图书馆是这种服务框架的一个应用实例。它把每条信息看作一个 Digital Object,所有服务都是围绕 Digital Object 设计的。框架提供了一套包括命名、标识、激活 Digital Object 的服务以及一套存储 Digital Object 的机制。

这个服务框架经过多年的研究开发,其中的部分内容已经有商业化实现了,如 HANDLE system^[1.26]。不过这个框架强调的是信息的管理,与我们强调的空间信息获取还有一定差别。

1.4.1.3 OGC 的工作

OGC 在抽象定义中提出的 Simple Feature Model^[1.27]总结了人们对空间信息的共识。OGC 于 1998 年提出 Web Mapping Testbed(WMT),历经 WMT1, WMT2,并于 2000 年提出了 Interoperability Program(IP)。IP 是用来帮助参与者互动地达成一致的间接接口的工程化方法^[1.28]。2001 年 3 月,OGC 提出了关于空间信息 Web 服务的技术倡议(Request for Technology, RFT)^[1.4],其中描述了服务模型,标识了模型中的角色与操作,提出从概念层、技术层和系统层,三个层次可以定义并逐步实现空间信息的 Web 服务。关于 web 服务的接口,OGC 也有专门论述^[1.29],提出了 WFS(Web Feature Server), WMS(Web Mapping Server)等 Web 服务的接口规范。

OGC 和 W3C 类似,负责协调大家的意见,并且寻求推出一些规范或推荐标准。它的成员主要是空间信息领域的研究机构和主要 GIS 厂商,它的工作目标针对空间信息应用领域。OGC 的工作对本文有很大的启发和指导作用。

1.4.1.4 TC211 的工作

TC211 是 ISO 中一个面向空间领域的一个组织,它由 5 个小组构成: WG1

负责框架与参照模型；WG2 负责空间信息模型和操作；WG3 负责空间信息管理；WG4 负责空间信息服务；WG5 负责文档和功能规范。TC211 的工作与 OGC 类似，它也对服务模型^[1.30]和元数据^[1.31]提出了规范。但是与 OGC 相比，OGC 侧重于民间的合作，而它则注重政府之间的合作以便达成国际规范。

1.4.2 GIS 实现技术简述

GIS 是一个面向具体应用的信息产业领域，因此它更强调如何集成最新的技术来支持具体应用的开发。下面的三个表（表 1-1，表 1-2，表 1-3）分别描述了几种 GIS 的分类，这几种分类说明了 GIS 发展的现状和发展趋势。

表 1-1 通信角度的 GIS 分类

名称	说明
单机版	通信在进程内部完成。如 MapInfo 的 MapInfo Professional。
网络版	通信在进程之间进行，采用 TCP/IP 协议，或者更高一级的通信手段，如 RMI, RPC, 或者分布式计算平台，微软的 DCOM ^{[1.32][1.33]} 、OMG 的 CORBA ^[1.34] 或 SUN 的 Java 和 Javabeans ^[1.35] 通信。如 Geo-Union 的 Geo-Union 5.0。
Web 版	进程利用 HTTP 协议在 Internet 上通信。如 OGC 的 Web Feature Server, ESRI 的 IMS。

表 1-2 数据存取角度的 GIS 分类^[1.36]

名称	说明
文件型	空间信息以本地文件的格式存储。如 ESRI 采用 Shape 格式存储空间信息。
数据库型	空间信息由数据库管理系统（DBMS）管理，用户通过 DBMS 来访问空间信息。如 ESRI 采用数据库存储空间信息，并用 SDE 访问数据库获得空间信息。
Internet 型	空间信息分布在 Internet 上，其存储管理由当地的 GIS 应用系统负责，Internet 用户主要通过中介服务以交换格式只读地获取空间信息。如本文的开放空间信息互操作框架和 Semantic Web。

表 1-3 系统构造技术角度的 GIS 分类

名称	说明
结构化设计	GIS 的功能以代码模块实现，通过代码连编形成应用程序。如早期的地理信息系统。
面向构件	GIS 的功能以对象构件实现，通过组合构件形成应用程序。如 MapInfo 公司的 MapX 和 ESRI 公司的 Map Objects。
面向 Agent	GIS 的功能以独立的应用程序实现，通过程序之间的通信协作达到应用目标。如本文的开放空间信息互操作框架。

1.5. 文章结构与主要贡献

1.5.1 文章结构

论文共分六章，结构如图 1-2。



图 1-2 文章结构

第一章“动机与简介”探讨了空间信息互操作中的困难，由此总结了其中的三个重要问题：可达性、可用性和易用性问题，随后介绍了信息互操作框架和一些 GIS 实现技术。

第二章“开放空间信息互操作框架”介绍了开放空间信息互操作框架(OSIIF)的基本结构——它是由服务和信息组成的支持空间信息互操作的服务模型。基于开放、动态、自动、简单和灵活的原则的 OSIIF 是 Internet 用户和网上异构空间信息之间的桥梁。其体系结构设计主要包括三部分：资源分类，资源之间的关系，参与人员的角色。最后也探讨了 OSIIF 与 Semantic Web 的结合点。

第三章“基于共识的分级空间信息模型”描述了 OSIIF 的信息模型。OSIIF 的信息模型用于解决信息可用性问题。其中探讨了 Internet 上的知识表示方法，并对比了几种 XML 知识表示方法的特点；划分了知识表示的级别，将信息异构分解到各个级别，简化信息转换复杂度；为了实现语义层次的信息互操作，划分了空间信息领域中共识的层次；可以用 XML DTD、XML Schema、RDF Schema 在结构、对象和语义三个不同级别向用户提供空间信息；设计了空间信息交换的分级共识矩阵，具体说明 OSIIF 的空间信息的交换格式。最后说明了构造空间信息产品的规程。

第四章“面向 Internet 计算的服务模型”描述了 OSIIF 的服务模型。该服务模型通过“角色—能力”方法将服务分类，具有开放性和动态性两大特点：支持

资源的动态加入,也支持动态的资源访问能力,还可以提供多层次多路径的服务。

该服务模型通过提供信息导航服务和信息获取服务来解决可达性问题。信息导航服务帮助用户了解存在哪些信息资源,而信息获取服务则可以作为桥梁,支持在 Internet 上获取空间信息资源,

另外,该服务模型通过提供服务导航服务来解决易用性问题。服务导航服务帮助用户了解存在哪些服务资源,服务元数据则不仅描述了服务的能力,还说明了服务的接口。这样不但可以提高人机交互的友好性,甚至可以支持信息处理自动化。

最后,还其中提出了服务过程描述方法——任务说明书,如信息流动图,以及任务说明书的生成技术。

第五章“用 OSIIF 构建空间信息领域服务体系”通过具体的例子说明了怎样建设和使用 OSIIF。其中包括:怎样向 OSIIF 中增加新的空间信息类型;怎样向 OSIIF 中增加空间信息资源;怎样向 OSIIF 中增加服务。最后以一个协调服务如何完成用户以服务脚本提交的任务,来说明 OSIIF 中问题求解的基本机制。

第六章“结论与展望”总结了全文,并且说明了进一步的工作方向。

1.5.2 主要贡献

论文围绕“网上异构空间信息的互操作与导航”提出了开放空间信息互操作框架(OSIIF),主要贡献如下:

- 设计了空间信息互操作框架体系结构,将服务和空间信息分离后统一为 Internet 上的资源,从资源之间关系的角度讨论空间信息互操作问题。
- 基于 Semantic Web 的关键技术——XML 和资源定义框架(Resource Definition Framework, RDF),设计了空间信息的分级共识矩阵,解决网上的异构空间信息的表示和交换问题。从而使处于异构环境、分布存放的各种专业和行业空间信息库可以协同工作,互为补充,既互相独立又能互操作。
- 设计了空间信息领域的信息服务体系:将空间信息服务划分成较大粒度的领域原子服务,方便用户构造 Internet 上的空间信息应用
- 分离出空间信息产品和空间信息获取服务,由后者动态地访问前者。一方面,增加了空间信息获取路径,另一方面也保证了所得信息与原始信息的一致性。
- 提供基于元数据库的导航服务,让普通的用户快速而准确地定位、获取并使用他们所关心的空间信息或相关服务
- 提出了服务元数据的概念,支持开放的服务体系,支持动态使用服务。

第2章 开放空间信息互操作框架

2.1. 目标

为了解决 Internet 上空间信息互操作中可达性、可用性、易用性问题,为了跨越 Internet 用户与空间信息之间的鸿沟,本文提出了开放空间信息互操作框架(Open Spatial Information Interoperation Framework, OSIIF),划分了空间信息互操作的角色以及它们之间的关系,简化空间信息互操作的复杂度。

2.2. OSIIF 的设计原则

2.2.1 开放性

开放性包括三个方面。第一,开放空间信息,OSIIF 要让信息提供者能够有效地在 Internet 上发布新的空间信息。第二,开放服务,OSIIF 也要让服务提供者能够有效地在 Internet 上发布新的服务。第三,OSIIF 应当寻求信息和服务中的共识,通过共识来保证空间信息互操作,这体现为实体之间可以采用相同的协议和一定的交换格式传递信息并理解信息。开放性是 OSIIF 与传统 GIS 或空间数据仓库的最大区别,它可以动态地包容网上的异构信息。

2.2.2 动态性

元数据是关于数据的数据。它可以帮助用户动态地获取信息、使用服务。OSIIF 提供了空间信息的元数据,从而支持信息使用者动态地定位、获取并使用空间信息。OSIIF 还提供了服务元数据,从而支持服务使用者在运行时刻,通过查询和理解其他服务的能力与接口,动态地定位并使用所需要的服务。

2.2.3 自动性

在元数据的支持下,OSIIF 中的服务可以帮助用户动态完成一些重复性的工作。例如导航服务,它可以帮助一个不熟悉框架的用户方便地找到所需的空间信息,也可以帮助用户寻找到相应的服务来获取该信息。又如,过滤服务可以帮助用户以所需要的信息中提取自己关心的信息。通过归纳常用的重复性工作,设计出 OSIIF 服务来自动完成这些工作,帮助用户把精力投注于空间信息应用的设计之中。

2.2.4 简单性

OSIIF 中的角色的接口应当简单,以便扩大其应用范围。例如,使用 OSIIF 服务时应当具有简单明了的接口,从而减少用户培训时间、减少系统中的限制、增大框架角色实现的多样性,从而促进框架的推广。又如,OSIIF 中通过抽象其中的角色与角色间的活动,把具体实现隐蔽起来。OSIIF 只要求说明各个角色的

接口，而不管他们具体采用什么方法实现：空间信息产品可以存储在数据库里，也可以存储在文件中；访问空间信息可以利用 FTP 协议，也可以利用 HTTP 协议；空间信息处理服务可以用 Java 编制，也可以用 C++编制；空间信息应用系统中可以同时包含 Unix 平台、Windows 平台。

2.2.5 灵活性

以往的 GIS 厂商都是自顶向下提供一整套系统。其中各个环节互相依赖，耦合度很强。而它们往往需要面对很多类型的用户，可能难以在特定应用中达到局部最优，有时甚至无法达到应用系统要求。参考软件构件的思想，OSIIF 把系统的各个环节相对独立起来，形成服务，通过服务之间的协作获取并处理信息，最终解决应用问题。这样，不但可以提高系统的整体稳定性，还有利于系统的扩展与升级。这样 GIS 就不再是由某些大公司所控制的垄断产品，而广大用户和自由软件开发者都可以为空间信息共享贡献力量，形成类似于 Linux 的 Internet 开发模式。

2.3. OSIIF 的体系结构

2.3.1 OSIIF 体系结构图

OSIIF 是连接 Internet 用户和现有空间信息资源的桥梁。其体系结构如图 2-1。

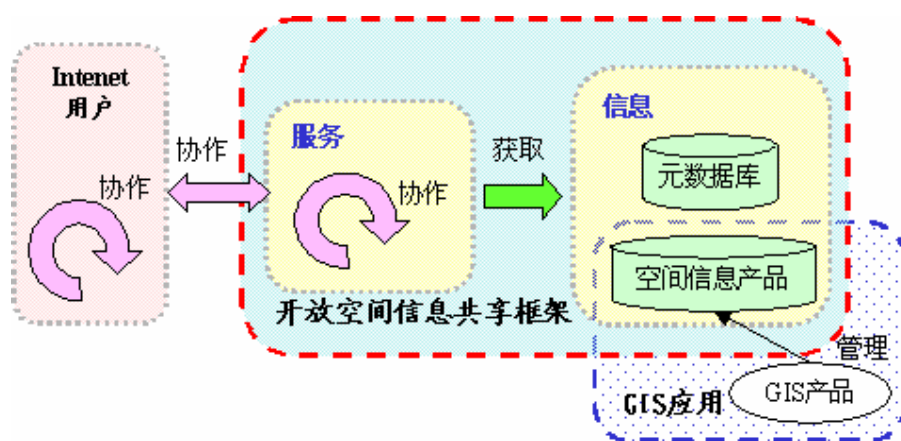


图 2-1 OSIIF 的体系结构

从图 2-1 可以看出，OSIIF 由两种不同的资源组成：信息和服务。通过分离信息和服务，可以有效地说明互操作的范畴：信息之间并没有直接的互操作；服务和 Internet 等应用程序的“协作”是互操作，服务“获取”信息的也是一种互操作。同时，这也有利于增加新的信息和服务，有利于增加信息获取途径等。

因为 OSIIF 中的空间信息资源大多来自现有的 GIS 应用系统，其中的信息已经且只能被特定的 GIS 产品管理维护，所以 OSIIF 重点考虑空间信息资源的获取与使用，而不关心他们的具体管理维护。这样可以降低问题的复杂度。

此外，图 2-1 还说明了 OSIIF 的是如何解决空间信息互操作中的三个重要问

题:

- 为了完成一定的任务, Internet 用户和 OSIIF 的服务之间可以进行多种形式、多种级别的协作。OSIIF 中的每个服务都是独立的进程, 具有独立完成一定任务的能力, 这样就形成了以服务为粒度的 Internet 计算平台, 为 Internet 用户构造系统提供便利, 这样有助于解决易用性问题。
- OSIIF 的服务还可以获取并理解 OSIIF 中的信息。OSIIF 以信息获取服务作为中介连接 Internet 用户和空间信息, 这样有助于解决可达性问题。
- OSIIF 的信息可以是空间信息产品, 也可以是元数据库。元数据可以描述信息, 也可以描述服务。通过元数据, 不但能方便用户理解并使用 OSIIF 的信息和服务, 还可以方便机器理解并使用 OSIIF 的信息和服务, 这样有助于解决可用性问题。

2.3.2 OSIIF 的信息

OSIIF 中有两类信息: 空间信息产品和元数据库。前者是空间信息领域应用所关心的信息, 而后者是对 OSIIF 中各种资源的描述性信息。

□ 空间信息产品

空间信息是用来描述客观世界中具有空间特性的事物的信息, 例如地图 (map)、图层 (layer) 和空间特征 (feature)。空间信息可以被创建、存储、处理、删除、浏览, OSIIF 假定空间信息已经存在, 侧重研究如何在 Internet 上获取并理解这些信息。

人们不但希望使用自己创建的信息, 还希望能使用别人创建的信息。可是, 大多 Internet 用户并不象信息创建者那样对 Internet 上的空间信息有充分的理解, 他们可能因为不理解信息的语法与语义而无法使用这些信息。所以当信息提供者把自己的信息共享出来时, 不但要提供信息还要提供信息的元数据。空间信息的元数据简称为“空间元数据”, 用户可以在它的帮助下了解并使用被它描述的空间信息。空间元数据用于帮助用户理解并使用空间信息, 它可以涉及空间信息的各个方面, 如信息标识、内容摘要与分类、信息来源、信息质量、更新状态、存储格式, 获取途径等。而不同的具体应用往往只会关注空间元数据的特定部分。

在空间信息交换过程中, 空间信息与空间元数据缺一不可, 这一点空间信息与现实世界中的商品十分类似。因此 OSIIF 把他们合称为“空间信息产品”。事实上关于信息交易的概念 OGC 早在 1998 年的简介中就论述了这个观念^[2.1]。空间信息产品的概念的引入, 有利于对空间信息的检索和使用, 也有利于将空间信息发展成一种可交易的信息商品, 进而促进电子商务在信息技术领域的业务拓展。

□ 元数据库

元数据是描述数据的数据。OSIIF 的元数据分两类: 空间元数据和服务元数据。空间元数据如前面论述; 而服务元数据则用于描述 OSIIF 服务的处理能力、接口协议、访问路径等信息。

OSIIF 将分布在各地的空间元数据和服务元数据收集起来, 通过一定的方式存储在一起, 形成元数据库。为了方便用户查找, 元数据库里存储的元数据不一

定要和资源提供者提供的原始元数据完全一致，其组织结构要受具体应用的功能和性能需求影响。另外，由于其中的元数据与具体的资源密切相关，一旦资源发生的变化，应当将变化体现到元数据库中相应的元数据项中。参照元数据的分类，元数据库可以分为空间元数据库和服务元数据库。

2.3.3 OSIIF 的服务

OSIIF 的服务功能多种多样，大致可以分成三类：空间信息领域原子服务、导航服务和协调服务。前两类服务具有原子性，他们构成了 OSIIF 的服务资源。

□ 空间信息领域原子服务

空间信息领域原子服务（下面简称“领域原子服务”）针对空间信息领域应用而设计的。他们负责处理空间信息。根据能否访问空间信息，他们又可以划分为两类：空间信息获取服务，负责获取空间信息产品，并以指定的中介格式返回结果；空间信息处理服务，负责再加工空间信息产品，例如过滤信息，融合信息，形成信息增殖服务。由于具体功能的差异，这两类服务还可以进一步细分。

□ 导航服务

导航服务是一种特殊的服务。它负责访问元数据库，利用元数据帮助用户查找所需的资源，或者从元数据库中提取信息帮助用户使用资源。由于资源分为服务和信息两类，也可以将导航服务分成两类：信息导航服务负责查找并获取空间信息产品的元数据；服务导航服务负责查找并获取领域原子服务的元数据。

□ 协调服务

协调服务是 OSIIF 中的一种特殊服务，它可以代表用户组织 OSIIF 中的领域原子服务协作，以服务过程的形式完成一定的任务。

2.3.4 Internet 用户

Internet 用户是 OSIIF 设计的重要参考因素，它是 OSIIF 以外的系统或人的代表。常见的用户有：

- 最基本的 Internet 上的地图浏览应用；
- Internet 上的空间信息资源与空间信息服务资源的搜索引擎；
- 使用了空间信息的决策支持系统；
- Semantic Web 上的 Ontology 推理机等。

从参与程度来看，有的用户只需要一个完整的解决方案，而有的用户则希望更多地参与服务过程的控制。

从信息使用角度来看，有的用户可以专门处理空间信息；有的用户则更注重通用的信息推理。

虽然用户的具体需求存在差异，但绝大多数 Internet 用户都有一致的要求，就是可以找到、访问并使用网上的异构空间信息和相关服务。

2.3.5 OSIIF 中的基本关系

如图 2-2，OSIIF 中的基本关系包括：两种活动——服务获取信息，服务之间协作；一种关系——元数据描述资源。两种活动事实上就是空间信息互操作的具体表现。

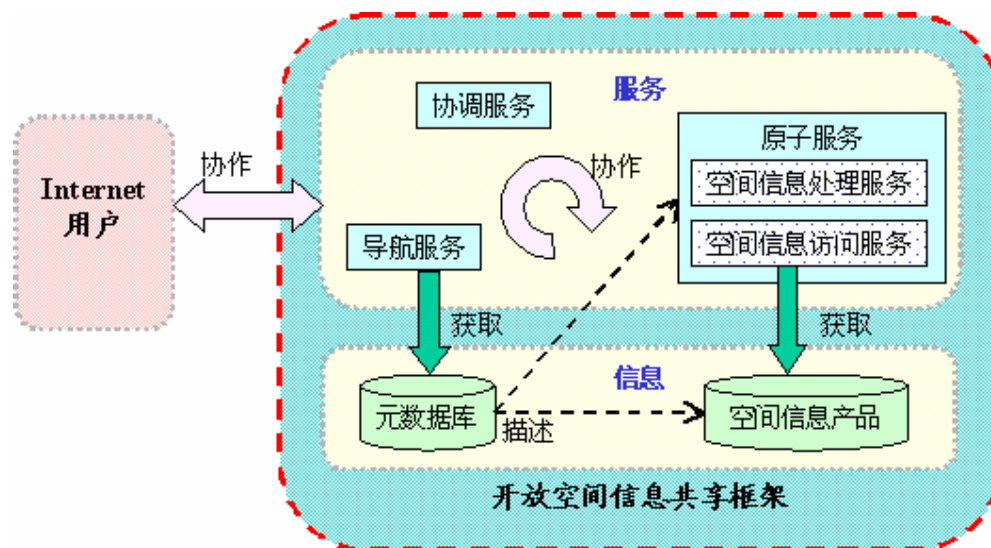


图 2-2 OSIIF 中的基本关系

□ 服务获取信息的活动

首先，空间信息获取服务可以获取空间信息产品的信息。因为服务和信息是相对独立的，所以它是一个动态的过程：空间信息产品由于存储规格不同，需要有不同的空间信息获取服务来访问，而获取服务则需要服务请求者说明空间信息产品的访问路径，以及信息获取要求。其次，导航服务可以获取元数据库的信息，这是一个相对静态的过程，用户不必了解元数据库的物理组织形式，只要依据由导航服务提供的元数据库的逻辑结构（Schema）就可以形成查询了。

□ 服务间协作的活动

Internet 用户与服务之间，服务与服务之间都可以有协作。协作采用主从方式：Internet 用户可以组织并控制一组 OSIIF 服务的协作；OSIIF 的协调服务也可以组织并控制 OSIIF 中其他服务的协作；导航服务和领域原子服务都是原子级服务。协作中需要考虑通信机制和协作控制。

□ 元数据库描述资源关系

元数据库用于描述 OSIIF 中的资源。用户可以在元数据的帮助下形成对 OSIIF 的全面认识，可以了解其中资源的分布、资源的分类、资源间的关系等信息。

2.4. 人与 OSIIIF 的关系

OSIIF 是由人来建设和管理的,因此需要明确人与 OSIIF 的关系。人在 OSIIF 中可以充当四种角色:框架设计者、信息提供者、服务提供者和用户(如图 2-3)。

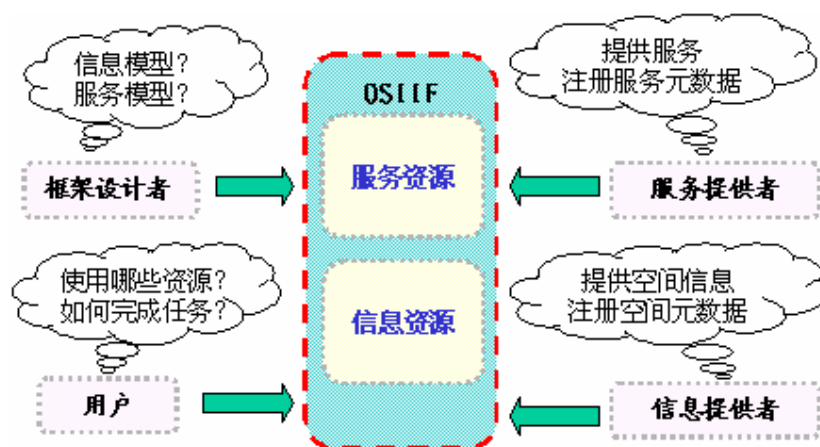


图 2-3 人与 OSIIF 的关系

第一,框架设计者负责提出大家应当遵循的共识,如采用怎样的信息模型和服务模型。

第二,信息提供者负责提供空间信息产品,不但要提供空间信息,还要设计并将相应的空间元数据注册到空间元数据库中。如果空间信息产品没有直接连接在 Internet 上,信息提供者还要充当服务提供者,提供中介的空间信息获取服务。

第三,服务提供者负责提供能在 Internet 上被访问到的服务,同时也要将设计并将相应的服务元数据注册到服务元数据库中。

最后,用户需要负责如何提出任务,组织服务过程。他要说明任务中要使用哪些资源,如何组织服务和信息共同完成任务。

2.5. OSIIF 与 Semantic Web 的关系

OSIIF 支持在 Internet 上空间信息互操作,它与 Web 在结构上是相容的。OSIIF 采用 RDF 定义元数据。此外,OSIIF 的信息获取服务输出的信息要遵循 OSIIF 规定的基于 XML 的中间格式,OSIIF 信息获取服务可以提供采用 RDF Schema 定义的中间格式。这样就把信息的语义与内容有机地结合起来,从而达到 Semantic Web 对信息交换格式的基本要求:用 Ontology 描述某个领域的一组概念,并且依据 Ontology 组织信息的交换格式,使交换格式中兼具语法和语义信息。

可见 OSIIF 在 Semantic Web 中扮演两个角色:空间信息领域的 Ontology 的提供者,这可以通过导航服务和元数据库来实现;有语义的空间信息的提供者,这通过空间信息获取服务和空间信息产品来实现。以上两个角色是 Semantic Web 基础设施建设的重要组成。

2.6. 本章小结

本章介绍了“开放空间信息互操作框架”的基本结构——是由服务和信息组成的支持空间信息互操作的服务模型。基于开放、动态、自动、简单和灵活的原则的 OSIIF 是 Internet 用户和网上异构空间信息之间的桥梁。OSIIF 的设计主要考虑三个问题：。

- 资源分类。OSIIF 由信息和服务两种 Internet 上的资源组成。信息可依据功能的差异分成空间信息产品和元数据库；服务可以分成协调服务，导航服务和领域原子服务。
- 资源之间的关系。OSIIF 中的基本关系包括服务获取信息活动、服务间协作活动和元数据库描述资源关系。前两种活动就是空间信息互操作在 OSIIF 中的具体表现。
- 参与人员的角色。我们把人在参与 OSIIF 的建设中的角色划分为用户、信息提供者、服务提供者和框架设计者，以明确参与人员的责任和分工，并说明如何具体建设 OSIIF。

此外，OSIIF 可以作为有语义的空间信息的提供者和空间信息领域的 Ontology 的提供者在 Semantic Web 中出现。

第3章 基于共识的分级空间信息模型

3.1. 目标

OSIIF 的空间信息模型是用来解决空间信息共享的可用性问题的。其中考虑两点：建造层次化知识表示模型，降低信息交换的复杂性；归纳空间信息领域的共识，建立相应的 Ontology，提供了信息理解、信息互操作的基础。

3.2. 知识表示

人们认识现实世界的过程，就是把客观事物映射到自己的知识体系中的过程。人与人之间交流知识、人记录自己的知识都需要通过某种媒体承载，例如文字，图形，声音等。知识表示就是人把自己的知识映射到某种具体的媒体的过程。鲁迅写杂文、贝多芬写交响曲、毕加索绘画等都是知识表示的行为，而知识表示的结果就是出现了文章、乐曲、图画等作品。事实上在人们的日常生活中，说话、写字等知识表示方法也都很常见。不仅是人类，计算机应用中也有知识表示的问题。计算机中知识可以二进制数据流的形式或程序的形式存在。本文讨论的是如何用二进制数据流来表示知识。

3.2.1 计算机的知识表示方法

Tim Berners-Lee^[3.1]认为计算机知识表示有两种流派。一种是传统人工智能的知识表示系统，这种系统中所有人都使用相同的概念定义。另一种流派就是以 Semantic Web 为代表的描述性知识表示系统。前者虽然能够正确地回答一些问题，但它们使用的推理规则通常只能局限在一个很窄的范围中。这是因为人们难以在很多概念上同时达成共识的，没有共识这样的系统就无法运行。而后者以允许冲突存在为代价，支持最大的灵活性。它的思路就是提供一种语言把所有的数据和各种推理规则都先开放到网络上，再开发一定的应用来完成推理工作。显然，后者更适于作为 Internet 上的信息表示方法。从实现角度看，XML 和 RDF 的结合就能构成一个很好的描述性知识表示系统。

3.2.2 XML 与 RDF

3.2.2.1 XML 协议集

XML1.0 通过标记文本的方法，不但可以表达信息内容，还可以表达信息的组织结构与语义。将计算机软件之间交换的知识提升到了有结构且有意义的符号流的水平。作为数据存储的重要组成的数据库领域，也出现了支持 XML 的数据库（Oracle9i），Internet 上的 XML 数据库（Wisconsin-Madison 大学的 Niagara 项目^[3.2]）的发展趋势。因此在网络上以 XML 交换数据是发展趋势。

用 XML 表示信息有很多分支。其中与 OSIIF 关系比较紧密的规范枚举如下：

- XML1.0 规范^[3.3]，其中包括了 XML 文本的定义，XML DTD 的定义；
- XML Namespace^[3.4]，描述了如何在 XML 中声明名字空间，如何使用有

限制的名字以区分确定同名标记；

- XML Schema^{[3.5][3.6][3.7]}，是一种面向机器、可定义数据结构的语言；
- RDF^[3.8]，以 XML 语法为基础，提供了描述了网络资源及资源之间关系的模型与语法格式；
- RDF Schema^[3.9]，提出了抽象世界中主要关系——由此建立类型系统，从而支持从客观世界到抽象世界的映射，也为知识共享打下基础；
- XML Information Set^[3.10]，提出了 XML 的数据模型，用于描述一个 Well-formed XML document 中的信息结构。
- XLink^[3.11]，支持在 XML 文档中加入链接元素，从而描述资源之间的关联。

其中 XML 文本 (XML document) 用于描述 XML 的通用语法，凡满足 XML 语法定义的文件就是定义良好的 (Well-formed)；而 XML DTD、XML Schema、RDF Schema 三种文档定义方法，则用于从不同的角度描述 XML 文本中用户定义的内容。若 XML 文本满足其中任意一种定义方法，则称它是有效的 (Valid)。这三种文档定义方法的描述能力各不相同 (如图 3-1)。

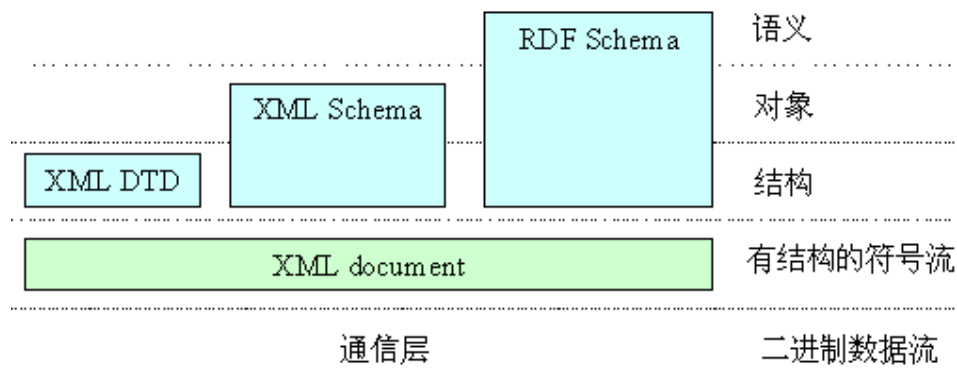


图 3-1 XML 文本及三种文档定义方法

XML 文本 (XML document) 的主要功能就是以字符流的形式存放的数据结构定义以及真实数据。它采用标记 (markup) 来区分信息流中指令与数据。其中还规定了字符集、基本符号、逻辑结构、物理结构的组织方式、以及 XML 文本正确性规则^[3.12]。

3.2.2.2 XML DTD 与 XML Schema

XML DTD 非常适合于描述结构化信息。它支持用户自定义结构，并提供了描述信息的逻辑结构以及信息的物理存储结构的方法：在逻辑结构中说明了如何用元素 (ELEMENT)、属性列表 (ATTLIST) 来存储信息的结构定义；在物理结构中说明了如何通过实体 (ENTITY) 组织具体的信息内容。

XML Schema 非常适合于对象化信息。它提供了用户自定义对象的能力；提供了用继承、组成等关系描述信息结构的能力；提供了基本类型；支持 XML 的名字空间，使得对象有了全局标识。XML Schema 还有一大好处就是面向机器处理，它的定义很支持语法解析器对信息流的处理。

表 3-1 中说明了 XML DTD 和 XML Schema 的关系。

表 3-1 XML DTD 和 XML Schema 的关系

XML DTD 和 XML Schema 着眼于描述以 XML 形式组织的符号流的语法结构。XML DTD 偏重信息的结构不考虑信息的含义；XML Schema 则把符号流的结构提高到对象的层次在一定程度上考虑了信息的含义，方便将以 XML 格式存在的信息映射到计算机软件中的对象体系，简化了分布环境中对象迁移中的存储格式转换过程。

3.2.2.3 RDF 与 RDF Schema

RDF 是 W3C 提出的一种知识表示模型。它希望支持网络上的信息共享（Information Sharing）与信息交换（Information Exchange）。它具有通用性，并不限于某个领域的网络资源定义。而它所描述的网络资源也可以是任意格式的。用 RDF 格式定义的信息，是机器可理解的（Machine-Understandable），这样可以支持对网络资源的自动处理，也能够由此实现 Internet 上的互操作^[3.13]。

RDF 提出了一个抽象世界——由资源（Resource）、属性（Property）、声明（Statement）构成的数据模型。通过命名属性和值，这个数据模型可以描述资源之间的内在关系。同时提出描述这个数据模型的基本语法，主要包括三部分内容：

- 以 Backus-Naur 范式规定了语法的以及一些缩写格式；
- 提出了三种容器 bag, queue, alternative 以表示资源的集合；
- 采用具体化（reification）的方法，实现声明（statement）的自我定义。这样，声明包括四个属性（subject, predicate, object, type）。

另外，RDF 主要有三种表现方式：RDF 图、四元组（声明，主体，属性，目标）或 RDF Schema。

RDF Schema 提供了一种把 RDF 映射到 XML 文本的方法。RDF Schema 中描述了 XML 文本的语法，同时也提出了一种对象类的定义方法，并且基于这种定义方法描述 XML 文本中的语义信息。其中包括几个重要概念：提出了类（class）的概念；不但描述存在哪些主要属性，也可以描述属性可以从属于哪些类；提出了限制（constrain）的概念^[3.13]。表 3-2 说明了 RDF Schema 与其他类型系统的区别。RDF Schema 的优势在于它的可扩展性与灵活性，但这也导致了它的缺点，因需要解释而处理效率比较低。一般来说只有必须用到语义的应用才考虑使用 RDF 格式规定的信息。

表 3-2 RDF Schema 对 class 的特殊定义

一般对象系统通过 class 拥有的属性来定义 class，而 RDF Schema 中 class 是一种资源（resource）的分类方法，类定义里并不描述它有什么属性，而用属性描述 class 与资源之间的关系。

3.2.2.4 XML 与 RDF

RDF 与 XML 有千丝万缕的关系，一般人很容易混淆其中的一些概念。表 3-3 说明了 RDF 与 XML 的关系。

表 3-3 RDF 与 XML 的关系^[3.13]

RDF 是一种元数据模型，而 XML 是一种语法格式（syntax）。RDF 数据模型可以用 XML 表示，同时 RDF 数据模型也可以用其他语法格式描述。RDF Schema 则是基于 XML 对 RDF 的一种实现。

数据模型分成通用（general）模型和专用（application-specific）模型。目前有很多通用模型，包括 XML Information Set（定义了 XML 数据模型），RDF data model and syntax（定义了 RDF 数据模型），这两种数据模型是平级的。同样，XML Schema 和 RDF Schema 分别基于不同的模型，它们不相同也没有必要合并。这里要注意两个概念：

- 数据模型，是一种抽象的概念，例如 RDF data model 是一种数据模型；
- 语言，是对抽象概念的形象描述，例如 XML 是一种语言，RDF Schema 是一种对扩展 XML 的符号和语法后的语言，它可以表示 RDF data model。

总之，XML 与 RDF 结合可以比较好地解决计算机知识表示问题。XML 可以帮助解决语法异构的问题，机器可以从用 XML 格式的信息表示中提取信息内容和信息的组织结构。同时，RDF 可以帮助解决语义异构的问题，它提供了表达知识和规则的概念模型。很多知识表示模型，如 John F. Sowa 提出的概念图（Concept Graph）^[3.14]、ISO/IEC JTC 1 提出的话题图（Topic Map）^[3.15]、Rational 公司提出的统一建模语言（UML, Unified Modeling Language）^[3.16] 和 Stanford 等大学提出的 OIL（Ontology Inference Layer）^[3.17]等都可以用 RDF 表示。

3.2.3 计算机的知识表示层次

人们通常用符号作为具体知识表现形式到抽象知识的中介。符号具有一定结构，有利于映射到特定的知识表现形式；而且它可以具有一定的含义，符号之间也可以存在有意义的关系，有利于映射到抽象知识。在计算机中的知识表示的映射，就是从人的知识体系映射到计算机的符号表示体系，再最终映射到机器的物理信号表示体系。

另外，直接来自概念世界的符号系统通常比较复杂，为了降低符号系统之间转换的复杂度，S.Melnik^[3.18]主张将信息分成三个层次：

- 语义层（或知识表示层）包括概念建模和知识工程。
- 对象层（或框架层）用于给出领域应用的面向对象描述。
- 语法层，负责将面向对象的信息具体表达成文本或字节流。

参考 Melnik 的观点，OSIIF 中将计算机的符号表示体系划分成四个层次，每两个层次之间存在映射，如图 3-2。

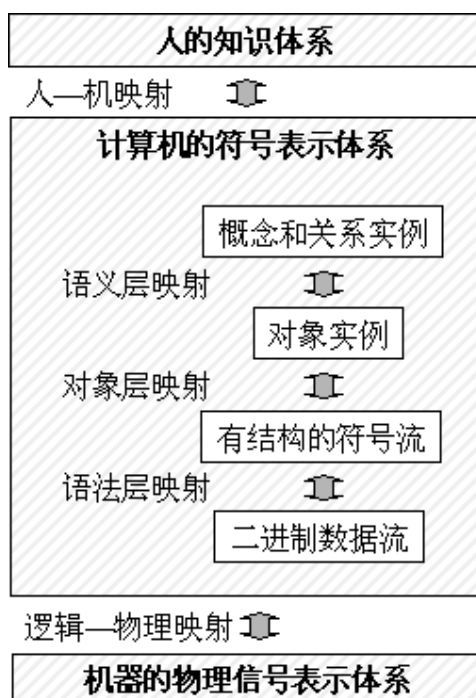


图 3-2 计算机中的知识表示的映射

3.2.3.1 计算机内部的知识表示

二进制数据流是计算机最原始的信息表示层次。其中只有 0, 1 两个符号。人很难从中识别出任何信息结构或理解这样形式的表示（不过，早期的计算机工作者由于只能面对这样的信息格式，他们的确能够从一堆 0, 1 中发现信息，并且直接用 0, 1 来表示他们的知识）。

有结构的符号流比二进制数据流前进了一步。其中规定了更为丰富的符号集，而且符号之间也出现了一定的结构。例如 XML 就是一种有结构的符号流，XML1.0^[3,3]中规定，XML 文本中的合法字符只有“tab”，“carriage return”，“line feed”，和所有 Unicode 和 ISO/IEC 10646 的合法符号。另外 XML1.0 也规定了符号之间的基本分类，以及符号组合的一些规则。

对象是抽象概念与具体符号结合点，也是人和计算机相互理解的中介。对象由对象类和对象实例成。一方面对象实例可以有效地用 XML 文本或其他表达形式（如内存表示）表示，另一方面对象类也可以很容易地映射到用户的概念体系。这样，既方便了不同表达形式之间的转换，也有利于提高用户表达能力。目前的面向对象应用程序就是一个很好的例子。

概念和关系描述了对象实例的含义和对象实例之间的关系，它是对信息内容的附加描述。有的应用并非针对某个特定的对象或结构，而是关心信息的含义。例如，在 Internet 上搜寻“名字”中含有“北大”的“公司”，就需要了解两部分信息：Internet 上有哪些对象类用于描述“公司”，而这些对象中又有哪些含“名字”这个属性；这样类的实例中“名字”属性值中是否含有“北大”。

3.2.3.2 计算机内部的知识表示间的映射

语法层映射负责从二进制数据流到有结构的符号流的映射。例如把二进制流映射到 XML 文本, 需要符号到二进制流的映射规则, 基本符号与符号组织规则, 扩展符号与符号组织规则等。

对象层映射负责从有结构的符号流到一组对象实例的映射。例如从 XML 文档中提取对象时, 需要设计一些对象类、描述对象类的含义、描述对象类之间的关系等。

语义层映射负责从一组对象实例到一组概念和关系的映射。例如从对象系统中提取概念与关系, 对象类可以与某个概念形成映射, 而对象类之间的关系可以直接或通过一个新生成的对象类映射到某个关系。

3.3. 共识与 Ontology

不同的人对客观世界的认识是有差异的。这种可能是由于客观事物不同而导致的, 也可能是由于不同人对同一事物具有不同的概念而导致的。后者是本文要讨论的语义异构问题。语义异构就是人与人之间缺乏共识而导致的。共识是指人们对同一个客观事物就具有相同的概念。形成共识的过程是这样的, 在社会活动中, 人们在已有共识的基础上交流有差异的信息, 最终形成新的共识逐渐就形成了共识。共识可以划分为三个层次: 概念模型级共识, 领域级共识和应用级共识。

3.3.1 概念模型级共识

每一种概念体系都拥有一套基本元素: 概念和规则。采用同一种概念体系的应用所采用的概念和规则也是一致的。例如, RDF 中由资源、主体、属性、目标、声明等; 概念图中有概念 (concept)、概念关系 (conceptual relation)、参与者 (actor) 等; 话题图中有主题 (theme)、话题 (topic)、关联 (association) 等; OIL 中有类 (class)、槽 (slot)、个体 (individual) 等。概念模型层是表达具体的领域知识的基础。

3.3.2 领域级共识

每个领域都有一套自己的基本概念体系。领域划分一般来自现实世界被大家所普遍接受的划分方法, 提出一套该被领域公认的概念体系, 例如医疗 (Healthcare), 交通 (Transportation), 电子商务 (Electronic Commerce), 金融服务 (Financial Services), 制造业 (Manufacturing Telecommunications), 空间信息 (Geospatial) 等。领域也可以划分成子领域, 也可以相互交叉。例如, 空间信息子领域可以有地图浏览 (view), 制图 (mapping), 出版 (publishing) 等。又如, 空间信息领域可以与电子商务领域结合形成房地产评估领域。

3.3.3 应用级共识

每个具体的应用系统中的软件需要互操作, 他们之间交换的信息不仅限于领域的共识, 而且需要在具体应用的概念体系上达成共识。应用系统内部的软件个体之间存在共识, 不同的应用系统之间也存在共识。例如地图浏览应用系统中,

空间信息提供者和地图浏览器之间的共识就不仅限于领域层共识，他们之间还需要对地图，样式表等概念形成共识。

3.3.4 Ontology

T. R. Gruber 给出了 Ontology 的定义：Ontology 是对概念系统的描述 [3.19]。在知识共享中，Ontology 是对一个或一组 Agent 所使用的概念和关系的描述，也就是概念模型。它与哲学里的 Ontology 是不同的。Ontology 就是一个具有共同兴趣的团体在术语上的共识。这些团体可以是人，也可以是自主的 Agent。Ontology 需要一种语言来表示，XML DTD、XML Schema 以及 RDF 都可以用于描述 Ontology。而基于 RDF 和 RDF Schema 的一种新语言 OIL (Ontology Inference Layer) 是目前 Semantic Web 推荐的 Ontology 表示语言。

3.4. 基于共识的分级空间信息模型

依据前面提到的两个理论，OSIIF 中提出了的基于共识的分级空间信息模型。其中包括 OSIIF 的空间信息共识，OSIIF 的空间信息交换机制两部分。最后提出了空间信息模型的核心——空间信息交换的共识矩阵

3.4.1 OSIIF 的空间信息共识

OSIIF 的空间信息共识包括三个层次，概念模型层，空间信息领域层，空间信息应用层（如图 3-4）。

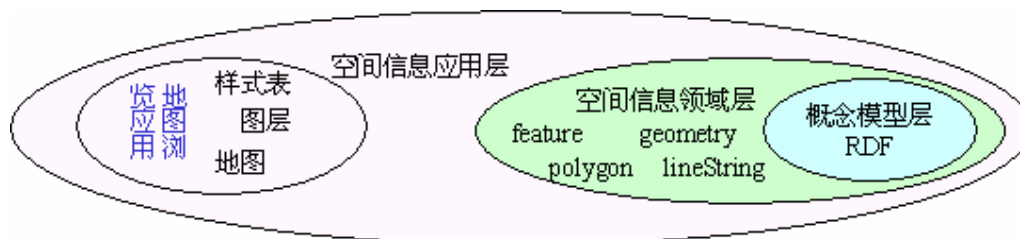


图 3-4 OSIIF 的空间信息共识

- 概念模型层选择 RDF 为基础，一方面考虑到它具有很强的概念描述能力，另一方面也考虑到它可以比较容易地映射到对象层表示。
- 空间信息领域层考虑如何统一对信息的空间特性描述。主要包括空间信息的基本特征概念，如点，线，面，坐标系，抽象的地理特征（Feature）等。框架中选择 OGC 的 Simple Feature Model 来界定领域层的概念。
- 空间信息应用层需要根据具体的应用来界定应用中的共识。例如，在地图浏览应用中存在地图、图层、样式表等概念。

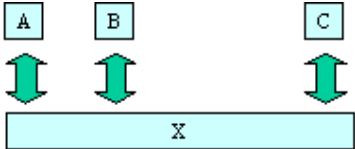
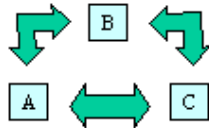
3.4.2 OSIIF 的空间信息交换机制

3.4.2.1 两种信息翻译模式

在空间信息交换中,不同的应用程序可以使用相同的信息表示,也可以使用不同的信息表示。采用不同信息表示的应用程序之间要交换信息,就需要翻译,即通过转换器转换信息表示。通常翻译有两种模式(如表 3-4)。

一方面,在信息表示比较多的情况下,中介模式需要的转换器要比对等模式要少得多;但是另一方面,中介模式中语法转换需要两次翻译,效率不及对等型,如果中介形式设计不好,还会导致转换中信息冗余或信息丢失。

表 3-4 两种常见翻译模式的对比

	中介模式	对等模式
原理	先从把源信息表示转换到某一中介形式,再由中介形式转换到目标信息表示。	任意两个不同的信息表示之间直接转换
转换 n 种信息表示的工作量	设计一种中介格式和 n 个双向转换器	需要 $n * (n-1) / 2$ 个双向转换器
图示		

因此需要根据具体情况决定采用哪种翻译模型。在 OSIIF 中,服务程序的数量是不可预测的,他们所接受的信息表示也难以控制,很难为每个新加入的服务的信息表示提供与已有信息表示之间转换器。因此 OSIIF 采用中介模型进行信息翻译,为框架中的信息交换设计中介信息表示。

3.4.2.2 OSIIF 的信息分级交换模型

OSIIF 的信息交换需要考虑两个问题:应用程序如何解析交换的信息;应用程序之间在哪个级别上交换信息。前一个问题可以参考前面论述的知识表示分层思想在应用程序开发中具体实现。后一个问题则需要在这里讨论。

关于 XML 文本(XML document), XML DTD, XML Schema, RDF Schema 的比较参阅 3.2.2 的论述。XML 的特性就是允许在交换的信息中包含用户对信息的说明,或者说应用程序交换信息时,不但需要一个包含信息的 XML 文本,还要对该 XML 文本的说明信息。用户使用信息时有不同的侧重点:有的只关心信息的结构、有的关心信息的对象化、有的则要求使用信息结构中的语义。一般来说,交换信息的级别越低,对信息表示方法的要求就越低。而信息表示能力越强,往往复杂度也就越大,也就越难以被机器处理。因此,OSIIF 应当在不同信息交换级别上选择合适的 XML 表示方法(如图 3-3)。

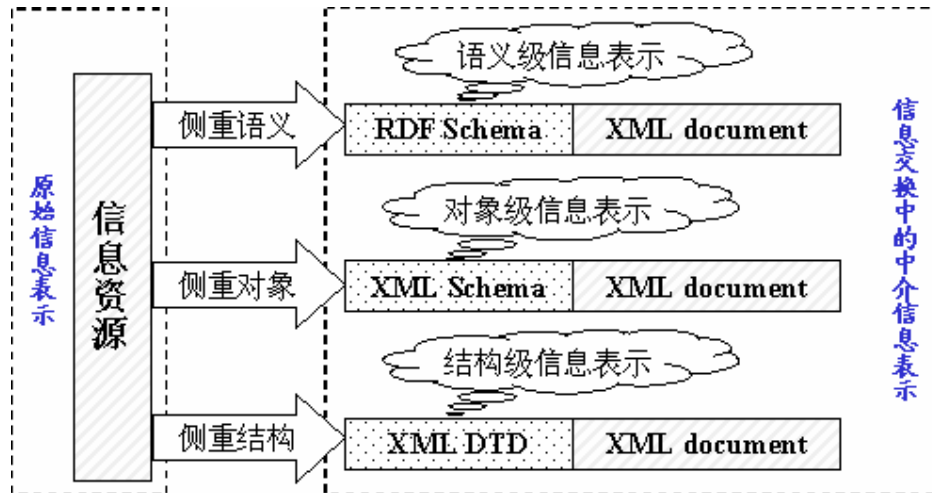


图 3-3 OSIIF 的分级信息交换机制

图 3-3 说明了在 OSIIF 中，空间信息要分成两部分，一部分是以 XML 文本表示的具体空间信息，它是 OSIIF 中信息交换的基础；另一部分是说明 XML 文本的 Ontology，说明了 XML 中用户定义信息的组织结构和具体含义。由于侧重点不同，所以具体采用的 Ontology 表示语言也是不同的：侧重结构只要使用基于 XML DTD 的空间信息定义；侧重对象只要使用基于 XML Schema 的空间信息定义；侧重语义只要使用基于 RDF Schema 的空间信息定义。空间信息领域的专用系统一般采用前两种表示；而一般的搜索或推理引擎则可以使用后两种表示。

3.4.3 空间信息交换的分级共识矩阵

OSIIF 中各个级别上都存在共识，可以用矩阵的形式表示出来（如表 3-5）。

表 3-5 空间信息交换的分级共识矩阵

共识 级别	概念模型层 (RDF)	领域层 空间信息领域 Simple Feature Model	以道路图层为例	
			应用层	具体信息
语义级	RDF Schema	gml.rdf [GML1.0 profile 3]	road.rdf	road.rdfs
对象级	XML schema	xlink.xsd geometry.xsd feature.xsd [GML2.0]	road.xsd	road.xml
结构级	XML DTD	gmlgeometry.dtd gmlfeature.dtd	road.dtd	road.xml

该矩阵说明了 OSIIF 中不同层次的共识在不同级别的推荐表达方式。概念模型层共识采用 W3C 提出的规范，在前面已经有所论述。领域层的共识采用 OGC 提出的 GML 规范。应用层的共识应当由具体应用决定。

GML (Geography Markup Language) 是由 OGC 提出的。此前 OGC 已经提出 Simple Feature Model, 用于描述空间信息领域中的共识。而 GML 关注如何组织 XML 文本来表示空间信息领域的共识。GML 有两个版本, GML 1.0 版[3.20]分别从 XML DTD 和 RDF Schema 中方法描述共识, 而 2.0 版[3.21]改用 XML schema 来描述共识。从 1.0 到 2.0 的变化说明了 OGC 为了推广 GML, 更倾向于把共识描述成容易被机器处理的信息。

GML1.0 的 profile 1 和 profile 2 描述了空间信息领域在结构级的共识。其中 profile 1 使用 GML 定义的两个 DTD 文件描述空间信息的 XML 文本格式, 而 profile 2 中允许用户基于 gmlgeometry.dtd 定义自己的数据结构。具体定义文件可以参见 gmlgeometry.dtd, gmlfeature.dtd。

GML2.0 描述了空间信息领域在对象级的共识。它允许用户利用抽象类派生来定义自己的对象。它采用 UML 描述了 OGC 的 Simple Feature Model, 并且给出了相应的 XML Schema 定义文件: xlink.xsd, Geometry.xsd, Feature.xsd。

GML1.0 的 profile 3 描述了空间信息领域在语义级的共识。它允许用户通过 RDF 定义方法描述自己的知识。图 3-5 说明了 GML 如何用 RDF 数据模型描述 Simple Feature Model。其定义文件为 gml.rdf。

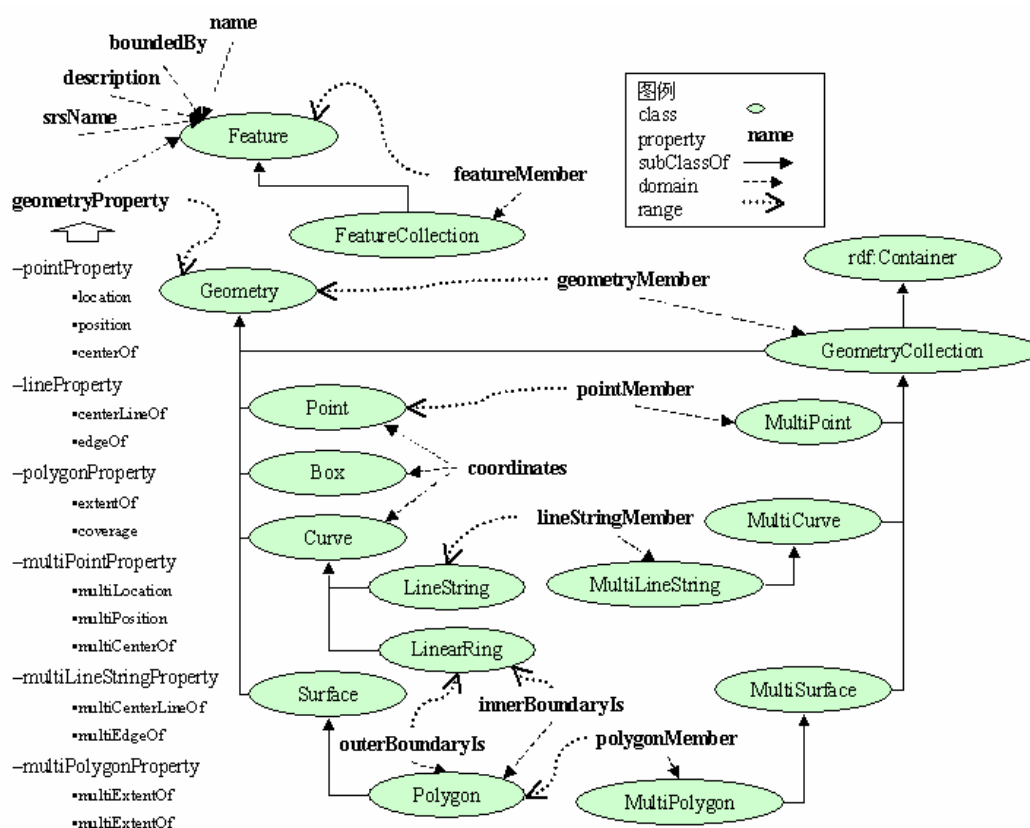


图 3-5 Simple Feature Model 的 RDF 表示

3.5. 创建空间信息产品的规程

创建空间信息产品分三个方面：设计空间信息产品定义，设计空间元数据，增加信息资源。

设计空间信息产品定义的过程是一个根据特定要求识别并描述空间信息的复杂过程，其中包括如下步骤：

1. 明确问题域，界定领域应用所涉及的信息；
2. 界定新概念的内容，含义；
3. 界定新概念与现有概念的关系，以及新概念之间的关系；
4. 设计对象体系；
5. 将概念映射为对象类，将概念之间的关系映射为对象的关系，若无法直接映射，可以先把关系映射到一个新对象类，再将该对象类和相关概念对应的对象类关联起来；
6. 描述为新加对象设计语法格式。说明现有的规格，以及中介格式。

空间元数据要根据应用要求并参照其他空间元数据来设计。其中应重点考虑内容说明、分类信息、存储规格和访问路径。

增加信息资源可以直接使用现有的空间信息产品定义，也可以先完成前两步后再增加。前两步针对的是增加新的空间信息应用时需要开发新的空间信息产品而设计的，而第三步主要是为已有的空间信息应用提供信息资源而设计的。

3.6. 本章小结

Internet 的信息应当采用描述性知识表示系统来表达。XML 和 RDF 技术结合就可以很好地做到这一点。XML DTD 和 XML Schema 侧重表示知识的结构，RDF 则具有同时表示知识语法结构和语义的能力。

计算机内部的知识表示的层次划分可以降低信息转换的复杂度。为此 OSIIF 把知识表示划分为概念和关系层、对象层、有结构的符号流层和二进制数据流层四个层次，层与层之间存在映射规则。

为了实现语义层次的信息互操作，可以划分三个层次的共识：概念模型级共识，领域级共识和应用级共识。基于共识可以定义空间信息领域的概念体系，即 Ontology。

根据上述方法，我们提出了 OSIIF 的基于共识的分级空间信息模型：

- 概念模型级采用 RDF，领域级共识采用 OGC 的 Simple Feature Model，应用级共识由用户群体自定义；
- 采用 XML 作为 Internet 上的空间信息交换格式，所有异构的空间信息资源在使用时都会以 XML 格式存在；
- 可以用 XML DTD，XML Schema，RDF Schema 在结构、对象和语义三个不同级别向用户提供空间信息；
- 设计了空间信息交换的分级共识矩阵，具体说明 OSIIF 的空间信息的交换格式。

创建空间元数据信息产品需要考虑一些的关键问题，如定义空间信息产品，设计相应的元数据，增加空间信息资源等。

第4章 面向 Internet 计算的服务模型

OSIIF 是一种 Internet 上的大粒度的分布计算平台，其重要目标就是帮助用户有效地在 Internet 上获取空间信息，而不论信息分布在哪里或者信息的具体存储格式如何。它通过服务模型和信息模型来达到这个目标。上一章讨论的空间信息模型提出了信息的理解问题的解决方法，而本章讨论的服务模型主要着重解决信息获取问题，即说明 OSIIF 如何向用户提供服务并完成信息获取任务。

4.1. OSIIF 服务模型

图 4-1 描述了 OSIIF 的服务模型，其中重点说明三个问题。第一，服务模型中有五种主要服务——服务导航服务、信息导航服务、空间信息访问服务、空间信息处理服务，以及协调服务。第二，服务模型是开放的，可以加入新的信息资源和服务资源；同时它也是动态的，协调服务可以在服务导航服务的帮助下动态集成服务路径，动态指派信息资源实例的访问者。第三，元数据描述了框架中资源的特性和使用方法，他们可以支持服务模型的开放性和动态性，也可以与导航服务配合为搜寻服务资源和信息资源提供支持。

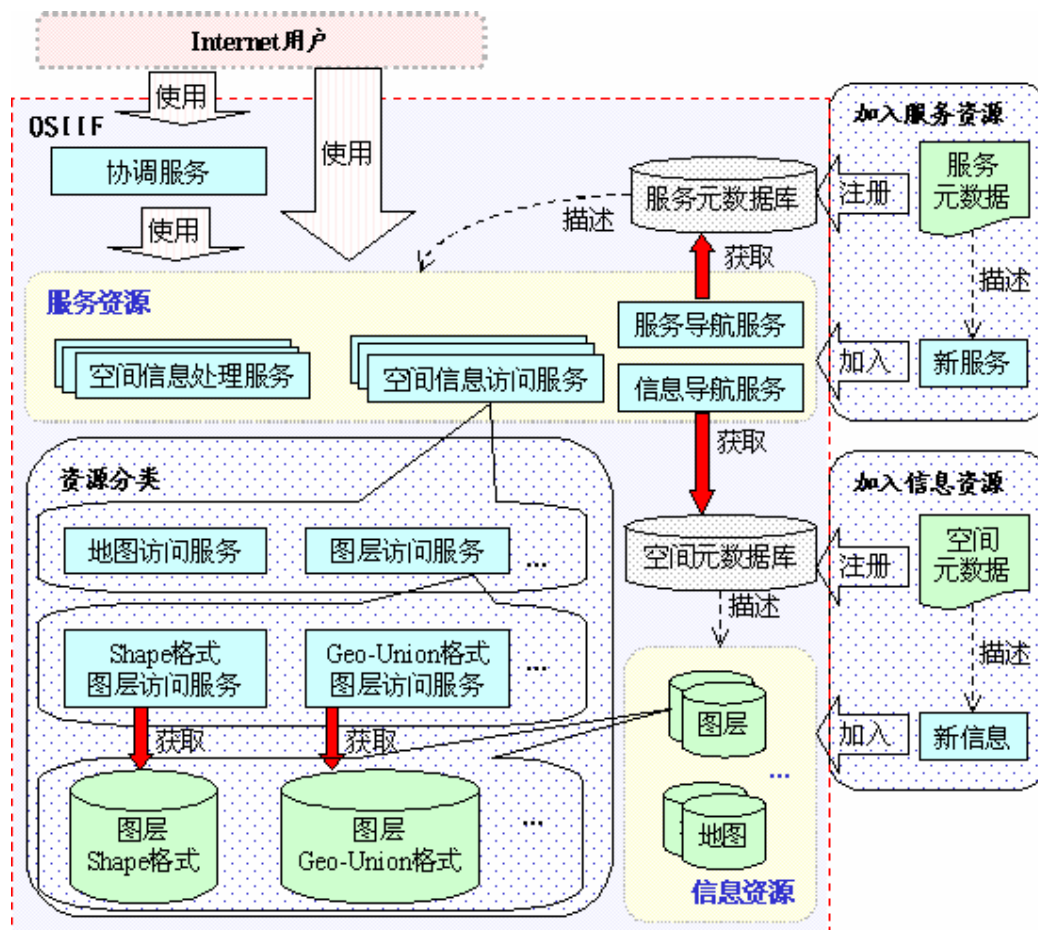


图 4-1 OSIIF 的服务模型

4.1.1 服务的层次

Tanenbaum 认为, 服务是一个实体向另一个实体提供的操作的集合^[4.1]。根据提供服务的实体在粒度上的差异, 我们把服务分成五层 (如图 4-2)。OSIIF 是面向 Internet 计算的, 它的服务模型主要考虑在两个层次上提供服务: 系统集成层 (提供协调服务) 和领域原子服务层 (提供领域原子服务)。

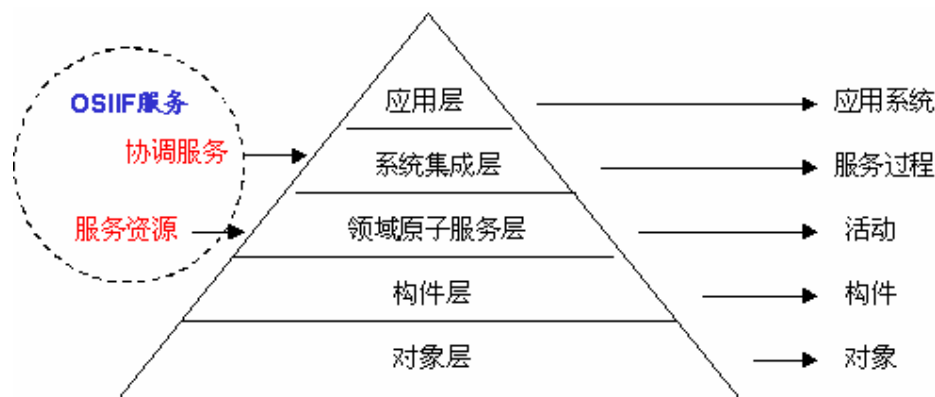


图 4-2 服务的层次划分及 OSIIF 服务所处的层次

对象层的服务实体就是对象。一个对象提供若干方法。对象封装了具体实现技术, 它是抽象服务和具体实现的基本结合点。

构件层的服务实体就是构件。一个构件由若干对象组成, 能够完成一定的任务。构件的交互需要特定的基础设施支持, 如微软的 COM/DCOM、OMG 的 CORBA 以及 SUN 的 JavaBean。构件层可以提供对服务分布性的支持。有很多 GIS 产品以支持二次开发的构件形式存在, 如 Map Objects (ESRI), MapX (MapInfo) 等。

领域原子服务层是针对领域中的原子服务而划分的。划分该层有两个原因: 首先 Internet 上的计算处于一个比较复杂环境, 而构件层的通信基础设施通常受到操作系统、网络结构等因素的限制, 因此需要考虑为应用提供一个通用一些的计算平台; 其次 GIS 产品中的构件受到厂商实现技术的限制, 不同厂商提供的 GIS 服务很难在构件层次上互操作, 而 Internet 上空间信息领域服务的应用往往又需要对服务的集成, 因此需要考虑一种支持互操作的服务体系。这样就形成了以应用程序形式存在的, 完成领域内一个不可再分的活动的原子服务。例如图层获取服务, 它具有明确的能力 (可以获取按一定存储规格存储的图层, 并且按中介格式输出), 它的一次执行需要确定的输入 (图层的访问路径, 要求的输出格式), 并且有确定的输出 (图层的交换格式)。

系统集成层服务是针对特定的用户需求而设计的, 它的一个服务实体可以完成一个服务过程。服务过程是来自各个领域的原子服务的有机组合。服务过程可以是众所周知的, 也可以是特定的。众所周知的服务过程可以定制, 例如获取图层的的服务过程, 用户只要指定了图层名和输出格式就可以了, 服务过程的协调者会根据对服务过程的认识, 自动去寻找适当的信息资源和服务资源来完成任务。特定的服务过程则需要通用的描述方法, 再由理解这种描述方法的服务来完成。另外, 服务过程可以依据集成要求来选择支撑服务, 它可以是一组构件的组合,

也可以是一组领域原子服务的组合。一般来说,支撑服务的粒度越小,集成者要完成的工作就越多。

应用层服务是面向具体应用的。它的一个服务实体就是一个包含人机交互界面的应用系统。它可以由若干服务过程的有机组成。空间信息领域的应用就属于这个层次。这个层次直接向最终用户提供了可操作的系统界面。

4.1.2 OSIIF 中服务的分类

OSIIF 服务模型中包括很多服务。为了方便管理这些服务,需要对它们分类。服务分类的方法有很多种:OGC 依据空间信息处理能力划分了十五种服务^[4.2],汪小林按应用需求划分了五种服务^[4.3],我们依据服务的“角色—能力”来分层次地划分服务。

“角色—能力”分类法的思想是:角色指服务在服务模型中担任的角色,能力就是服务的功能。首先划分角色,然后再根据服务的功能嵌套地划分服务。服务的能力是按照嵌套分类法定义的,例如 GeoUnionDB 图层获取服务的角色是“空间信息获取服务”,其能力可以表示成一个广义表:(“图层”, (“DATABASE”, “GeoUnionDB”))。

服务模型中有五种角色如表 4-1。其中协调服务由具体需求决定。两个导航服务的功能是比较确定的,用户只是不了解他们的访问路径。空间信息获取服务和空间信息处理服务是空间信息获取领域的原子服务,用户不仅需要了解他们的访问路径,还希望确切地知道他们具备什么具体的服务能力。这时就需要考虑“角色—能力”分类法中服务的能力了。例如空间信息获取服务可以访问哪种类型的空间信息产品,甚至具体到它可以访问什么样存储规格的空间信息产品。

表 4-1 OSIIF 中服务分类

服务分类	服务层次	描述
协调服务	系统集成层	负责组织服务资源完成一定的服务过程
服务导航服务	领域原子服务层	属于服务资源,帮助用户查找服务资源
信息导航服务	领域原子服务层	属于服务资源,帮助用户查找信息资源
空间信息获取服务	领域原子服务层	属于服务资源,帮助用户获得空间信息
空间信息处理服务	领域原子服务层	属于服务资源,帮助用户处理空间信息

4.1.2.1 协调服务 (Coordination Service)

协调服务是系统集成层的服务。它负责组织 OSIIF 的服务资源来实现一个服务过程。它具有两种服务模式,定制型和灵活型。定制型协调服务主要针对某个众所周知的服务过程,其中已经比较清楚地知道服务过程中的用到哪些类型的原子服务,以及原子服务之间的关系,所不清楚的只是具体包括哪些原子服务。灵活型协调服务主要面向特定的用户需求,这时用户应当通过一种描述向协调服务说明服务过程如何实现。

4.1.2.2 服务导航服务 (Service Navigation Service)

服务导航服务负责访问服务元数据库,帮助客户有效且迅速地查找服务资源。它负责帮助用户查找与服务相关的信息。针对不同的服务资源,它可以提供不同的服务。

服务导航服务的重要功能就是提供它所能获取的服务元数据库的组织结构。OSIIF 的服务元数据库以 RDF 模式组织, 采用 RDF Schema 表达元数据的格式, 因此, 服务导航服务可以向用户提供服务元数据库的 RDF Schema。

一旦拥有 RDF Schema, 用户使用 Xquery 查询^[4.4]向导航服务提出请求, 并得到相应结果。这样服务导航服务的另一个功能就是支持 XQuery 查询。

对两种导航服务来说, 由于用户已经了解导航服务的访问接口, 它只需要能够返回其他导航服务们的服务访问路径列表即可。这样, 就可以帮助用户找到其他导航服务了。

对一般的服务资源(非导航服务)来说, 它也可以提供一些定制的查询支持, 例如按“角色—能力”查找服务资源、按服务标识获得服务的整个元数据等。

4.1.2.3 信息导航服务 (Information Navigation Service)

信息导航服务负责访问空间元数据库, 帮助客户有效且迅速地查找信息资源。它可以帮助用户查找与空间信息产品有关的信息。它提供的服务与服务导航服务有类似之处:

- 提供描述空间元数据库的 RDF Schema;
- 提供 XQuery 查询支持;
- 对空间信息产品提供一些定制查询, 如按一定条件(如空间范围, 地名等)查找信息资源、获得指定信息资源的空间元数据。

从实现角度来看, 导航服务有两个重要评价指标, 命中率和响应时间。针对特定查询的专用的检索机制往往可以大大提高检索的正确性和缩短响应时间, 导航服务可以考虑支持多种检索机制, 并且针对查询组织其中的元数据存储, 提供一定的索引机制。导航服务的用户可以是人或机器, 它应当提供机器可理解 (Machine-understandable) 的接口。由于元数据库采用 RDF Schema 描述, 因此导航服务可以考虑支持基于语义的查询。

4.1.2.4 空间信息获取服务 (Spatial Information Retrieval Service)

空间信息获取服务侧重信息获取。它的主要任务就是访问一个空间信息产品实例, 并且把该产品以指定的交换格式反馈给用户。该服务一般具有如下特性:

- 可以按框架规定的交换格式输出空间信息, 这样在信息获取的初期就可以消除空间信息的语法差异;
- 可以支持动态信息服务。它可以直接从原始信息源获得信息, 而不必从其复本中获得信息, 这样就可以保证信息的实时性;
- 可以扩充能力。它可以仅只是完整地获取指定空间信息的交换格式, 也可以在获取信息时增加过滤条件。例如, 我们可以独立开发出一个获取并转换 ESRI 的 SHAPE 文件格式的信息获取服务; 也可以利用 MapObjects 和 SDE 来开发一个获取并转换 ArcInfo 数据的信息获取服务, 在服务中提供一定的查询接口, 将用户的查询条件映射到 MapObejcts 和 SDE 的相应接口, 以提高查询效率。

空间信息产品有很多种, 每种空间信息产品可以有不同存储规格, 而即使存储规格相同, 其访问路径也有所不同。一般一种空间信息获取服务只能访问一种特定类型的空间信息产品, 例如, 对于 ERSI 格式的空间信息产品, OSIIF 需要提供可以访问 ESRI 格式空间信息产品的空间信息获取服务。因此空间信息获取

服务的分类可以参照空间信息产品分类方法。例如它可以分成图层获取服务, 地图获取服务等。而图层获取服务又可以分成 MIF 图层获取服务、GeoUnionDB 图层获取服务。

4.1.2.5 空间信息处理服务 (Spatial Information Process Service)

空间信息处理服务不直接访问信息资源, 侧重于向用户提供计算资源。它遵循一定通信机制, 提供一定处理能力, 一次服务可以有若干输入、输出。一般来说, 这种服务的原子性体现在它只完成一个服务活动, 例如过滤一个图层, 综合几个图层生成地图。由于处理能力各不相同, 空间信息处理服务目前并没有细分下去。

4.1.3 OSIIF 服务模型的特点

根据前面对 OSIIF 服务模型的描述, 它具有三大特点: 开放性、动态性和多样性。开放性体现在框架允许加入信息资源和服务资源, 不仅可以增加资源的实例, 还可以增加资源类型。动态性体现在用户在使用框架服务时, 可以通过描述性的任务说明得到具体的任务实现路径, 服务请求者可以执行时再获得所需的服务资源和信息资源的具体访问路径。多样性体现在框架提供多层次的服务支持, 以及允许同一任务有多条完成路径。下面分别讨论 OSIIF 服务模型的特点。

4.1.3.1 支持加入空间信息产品

OSIIF 服务模型支持信息提供者发布空间信息产品。为此需要空间元数据库的支持。空间元数据库存储了所有注册的空间信息产品的空间元数据。加入空间信息产品涉及两个问题, 新增空间信息产品类型和新增空间信息产品实例。前者在上一章已经讨论过, 这里主要说明如何增加已有空间信息产品的实例, 具体步骤如下:

1. 使空间信息产品可以被 OSIIF 中相应的信息获取服务资源访问到
2. 在空间元数据库中注册相应的空间元数据。这里要注意几个问题, 不同的空间信息产品的空间元数据格式是有差异的, 注册信息应当按照相应格式填写; 注册信息中应当包括该空间信息产品的访问路径。

4.1.3.2 支持加入服务资源

OSIIF 服务模型支持服务提供者发布服务资源。为此需要服务元数据库的支持。这里也涉及到增加服务类型和增加服务实例的问题。实现方法与加入信息资源类似: 前者考虑设计新的服务元数据格式; 后者也分两步, 先把服务资源放到 Internet 上, 再在服务元数据库中注册服务元数据, 也要包括服务的访问路径。关于服务元数据的设计将在本章第二节中详细讨论。对于两种导航服务, 只需要将他们的访问路径注册到服务元数据库中就完成了他们的发布。

4.1.3.3 支持动态访问服务资源

服务模型支持服务使用者在框架中动态访问服务资源。为此需要服务导航服务和服务元数据库的支持。服务资源分布在 Internet 上, 用户很难完全掌握服务的分布情况。因此在服务模型中, 服务请求者可以在服务导航服务的帮助下, 在服务元数据库查找满足要求的服务, 获得指定服务的能力和访问路径, 并且据此

实现对该服务的访问。其中涉及两个重要问题：服务元数据库如何描述服务的能力，服务请求者如何使用服务。

□ 服务元数据库如何描述服务的能力

为了区分服务并查找所需服务，需要描述服务能力。服务模型提供了按“角色-能力”的分类方法，一方面通过划分服务减少搜索范围，另一方面也表达了服务在能力上的共性与差异。同时，服务模型通过服务元数据还描述了服务的输入、输出的规格，这样有利于判断服务之间是否能够交换信息。

□ 服务请求者如何使用服务

为了帮助服务请求者使用服务，需要服务提供者和服务请求者之间达成共识。一方面，服务提供者需要通过服务元数据描述服务的访问路径，如主机地址，通信协议等；另一方面，服务请求者需要能够理解访问路径，并且利用访问路径信息访问服务资源。

相关服务元数据的设计将在本章第二节中详细讨论。

4.1.3.4 支持动态获取空间信息产品

服务模型支持信息使用者动态获取空间信息产品，这样有助于解决两个问题：

□ 信息异构问题

服务模型提供自动寻找可以获取空间信息产品并以指定格式输出的空间信息获取服务。这样就在信息获取的初期，解决了信息异构问题。事实上，采用数据仓库也可以解决信息异构，就是先把异构信息转换成中介格式，然后存储到数据仓库中，用户直接到数据仓库中查询所需的空間信息。但这种方法静态的，中介格式难以保持与原始信息的一致。因此 OSIIF 在这一点上是优于数据仓库的。

□ 信息不完整问题

Internet 上的信息资源分布情况各式各样，信息使用者难以全面了解 Internet 上空间信息资源分布的情况。用户只能提供要什么信息，而不知如何获得信息。因此需要 OSIIF 的服务导航服务、服务元数据库、信息导航服务、空间元数据和空间信息获取服务的支持。一个不了解 Internet 上资源分布情况的信息使用者获取信息的过程如下：

1. 通过信息导航服务的帮助（也就是在空间元数据库中查找），找到满足要求的空间信息产品；
2. 通过信息导航服务的帮助，得到指定空间信息产品的类型、存储规格和访问路径等；
3. 通过服务导航服务的帮助（也就是在服务元数据库中查找），以空间信息产品类型、存储规格和所需的输出格式为条件，查找满足条件的空间信息获取服务；
4. 通过服务导航服务，得到指定的空间信息获取服务的访问路径；

5. 利用空间信息产品的访问路径和空间信息获取服务的访问路径, 向特定得空间信息获取服务发出请求, 并且获得具体的空间信息产品。

4.1.3.5 支持任务求解的多级别的服务

OSIIF 服务模型可以对任务求解提供多级别的服务支持。它可以通过向 Internet 用户开放所有服务, 在两个服务级别上提供服务:

- OSIIF 的协调服务可以向用户提供系统集成级的服务。用户可以直接把一个比较复杂的任务交给协调服务完成。
- OSIIF 的四种服务资源可以向用户提供领域原子服务级的服务。用户可以直接与服务资源交互, 或者组织它们的协作以完成任务。

4.1.3.6 支持任务求解的多路径选择

服务模型可以对同一任务提供多条实现路径。信息资源中可能有内容一样, 但存储规格不一样的空间信息产品; 服务模型中可以存在具有相同功能, 但访问路径存在差异 (如所处的主机不同、接口不同) 的服务。这些因素都可能导致任务求解中出现有多种可选的实现路径。这样用户可以根据自己的偏好或应用特性来决定如何选择服务路径。

4.2. OSIIF 的服务元数据

为了管理并使用分布在 Internet 上的众多服务资源, 需要用服务元数据描述服务的能力和访问接口。服务元数据是描述服务的信息。服务元数据库是一个逻辑概念, 它可以存放注册的服务元数据。

OSIIF 的服务元数据主要包括三个部分服务的标识、能力和访问接口。OSIIF 的服务元数据库存放了所有服务资源的元数据, 但不包括协调服务的元数据。服务资源中的四种服务都处于领域原子服务层, 每个服务只具有一种能力, 也只对应一条服务元数据。

4.2.1 设计思想

首先, OSIIF 的服务元数据应当是机器可理解的, 因此它应当用 W3C 的 RDF 来描述服务元数据定义, 用 XML 作为信息表示载体。这样应用程序就可能自动理解服务元数据并且利用它执行操作了。

其次, OSIIF 的服务模型采用继承方法定义服务元数据。OSIIF 服务之间存在差异, 而即便是同一种服务也可以有不同的实现。服务的分类关系体现它们的共性与差异, 即如果一类服务可以分成若干服务子类或服务个体, 那么这些服务子类或服务个体所拥有的元数据格式应当从该类服务的元数据中派生出来。图 4-3 说明了如何用 RDF 定义空间信息服务元数据。

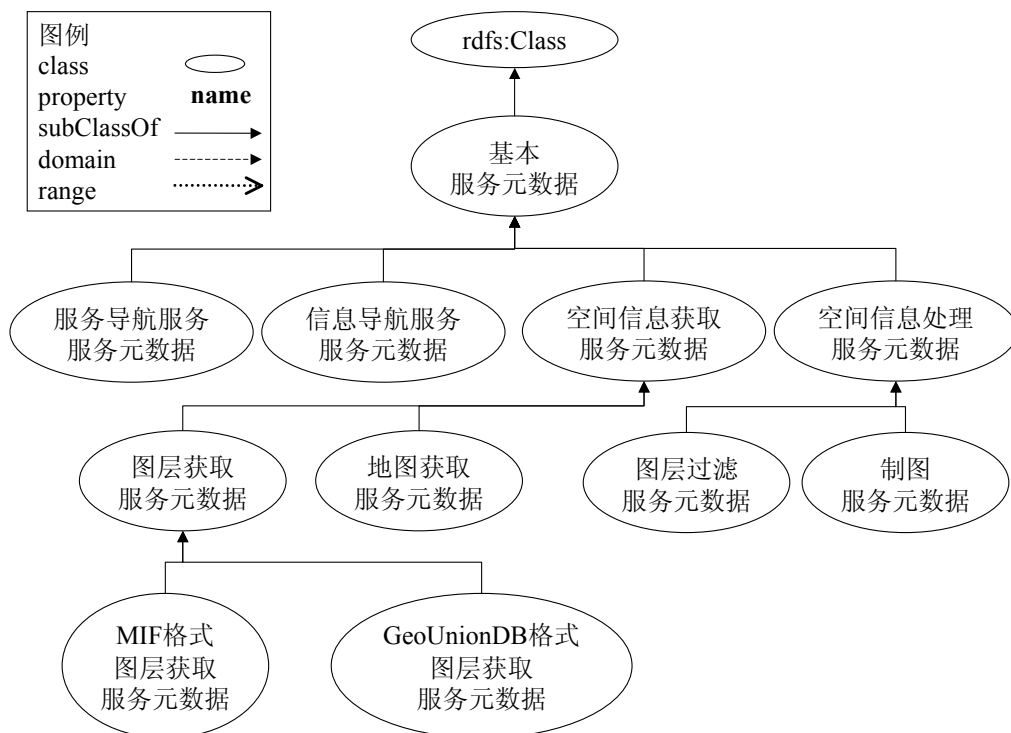


图 4-3 OSIIF 的服务元数据的派生关系 (RDF 图)

（注，此后所使用的图中若有“RDF 图”标记，都使用本图提供的图例。图例中的概念请查阅 RDF Schema 规范）

第三，OSIIF 的服务元数据的组成要素应当包括服务标识、服务能力描述和服务接口描述（如图 4-4）。服务的标识区分具体的服务，可以帮助获取服务元数据；服务能力描述用于说明服务的抽象定义，可以帮助用户查找服务；服务接口描述用于说明服务的通信接口，可以帮助用户使用服务。

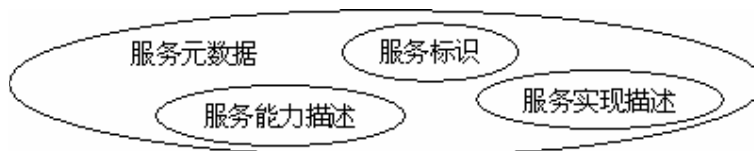


图 4-4 OSIIF 服务元数据的组成要素

4.2.2 服务标识

OSIIF 的建设者在加入资源的时候，要为每个资源分配一个全球唯一的标识，这就是服务标识（RID，Resource Identifier）。URI^[4.5]、XML 名字空间^[4.6]、域名系统^{[4.7][4.8]}和 Digital Object 系统的 Handle 子系统^[4.9]研究了 RID 的实现。OSIIF 的实现中提出了一种 RID 的分配方法：

RID ::= 元数据库标识 : 资源类型 : 库内资源标识

- * 元数据库标识: 是全球唯一的标识, 一般采用元数据库所在的主机的 URL
- * 资源类型: “S”代表服务资源, “I”代表信息资源。
- * 库内资源标识: 由资源注册的元数据库负责分配并保证其唯一性。

服务标识可以作为基本服务元数据的一个属性加入 (如图 4-5)。

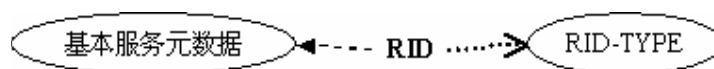


图 4-5 服务标识作为基本服务元数据的属性 (RDF 图)

4.2.3 服务能力描述

服务提供者可以从多种角度来描述服务的能力, 如服务的“服务分类”和“输入/输出参数”。

4.2.3.1 服务分类

OSIIF 采用“角色—能力”作为分类规则, 显然这种分类是可以描述服务能力的。首先根据角色划分出四种服务资源。其次, 根据处理空间信息产品的能力作为划分出空间信息获取服务的标准。再次, 根据能访问的空间信息产品的存储规格作为划分图层获取服务的标准 (如图 4-6)。这样, 对服务能力的描述被不断细化, 精确化, 为准确定位服务提供了依据。

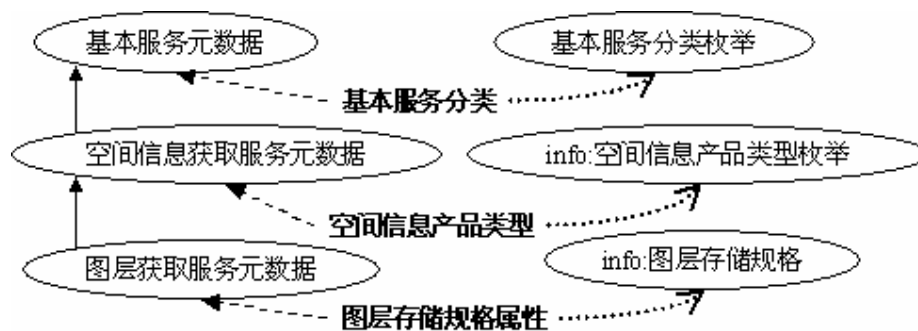


图 4-6 服务分类相关的服务元数据 (RDF 图)

4.2.3.2 输入/输出参数

由于 OSIIF 的服务只有单一的功能, 所以服务的一次执行所用到的输入和输出可以作为服务能力的一种表现。输入参数描述了一个服务执行前需要哪些信息, 输出参数则说明一个服务执行后可以提供哪些信息。每个参数有两个属性: 参数 ID 是用来标识参数的, 参数类型描述了该参数所对应的空间信息类型定义的 schema 的 URI。因为所有服务都要包括服务的输入输出参数, 所以它可以作为基本服务元数据的属性加入 (如图 4-7)。

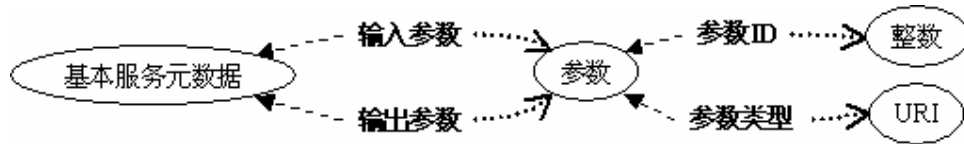


图 4-7 输入/输出参数作为基本服务元数据的属性 (RDF 图)

4.2.4 服务接口描述

服务接口描述用于说明服务的对外接口，这需要考虑两点：服务能理解什么样的信息，服务之间怎样交换信息。目前 OSIIF 的服务采用 HTTP 通信（如图 4-8），下面讨论 OSIIF 基于 HTTP 通信的机制，以及相关的服务元数据。

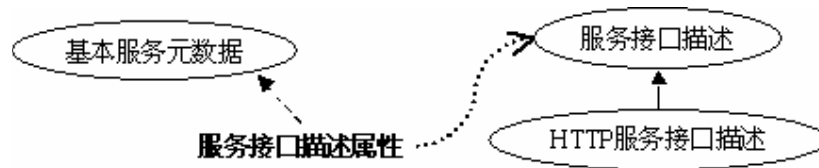


图 4-8 服务接口描述作为基本服务元数据的属性 (RDF 图)

4.2.4.1 服务的通信接口

基于 HTTP 的 OSIIF 服务的通信接口可以分三个层次：抽象层、服务层和通信层（如图 4-9）。

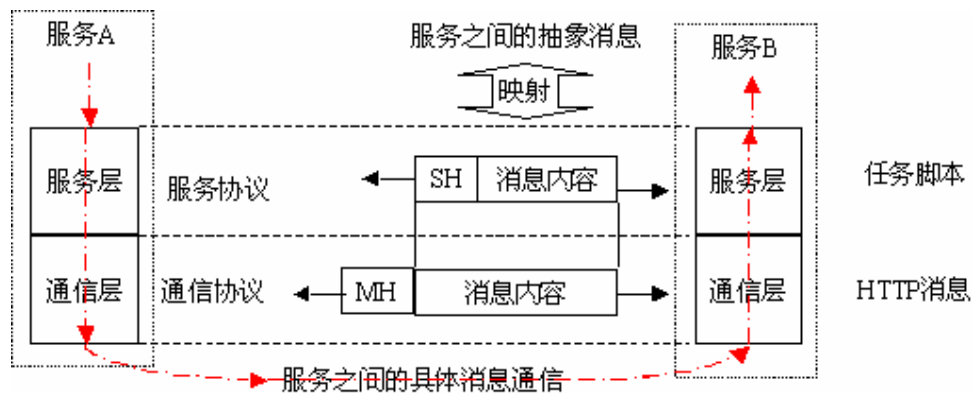


图 4-9 服务的通信接口

抽象层描述服务之间可以传递哪些消息，这个层次可以通过输入/输出参数来描述。

服务层关心服务之间能够理解哪些信息，这个层次可以用服务脚本来描述。不同的服务可以支持不同的服务脚本，因此需要考虑如何将输入/输出参数映射到服务层的消息中。关于服务脚本的讨论将在 4.3 节中讨论。

通信层关心服务之间如何具体传递消息，这个层次负责将服务层的消息直接封装，然后发送给其他服务。OSIIF 中目前规定所有服务在这个层次都采用 HTTP

协议，即采用 HTTP 消息封装服务脚本，采用 HTTP 协议进行通信。该层已经明确了消息封装的策略，只需要说明服务的访问路径即可。

4.2.4.2 生成 HTTP 消息

通信层封装的策略由通信协议决定，OSIIF 采用 HTTP 协议。HTTP1.1 协议 [4.5] 中规定了 HTTP 消息的格式，现摘录一部分如表 4-2。

表 4-2 HTTP 消息的格式

HTTP-message	= Request Response	; HTTP/1.1 messages
Request	= Request-Line	; Section 5.1
	* (general-header	; Section 4.5
	request-header	; Section 5.3
	entity-header)	; Section 7.1
	CRLF	
	[message-body]	; Section 7.2
Request-Line	= Method SP Request-URI SP HTTP-Version CRLF	
Response	= Status-Line	; Section 6.1
	* (general-header	; Section 4.5
	response-header	; Section 6.2
	entity-header)	; Section 7.1
	CRLF	
	[message-body]	; Section 7.2

生成一个 HTTP 消息包括几个子任务，相应地也需要一定的服务元数据（如图 4-10）。

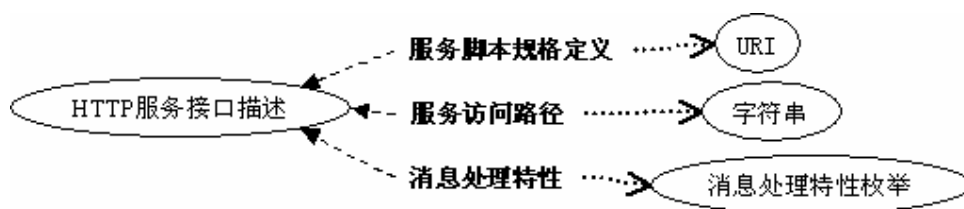


图 4-10 HTTP 服务接口描述的属性（RDF 图）

首先，要考虑消息发送给谁，这需要通过 Request-URI 部分指定目标服务的访问路径。因此服务元数据中应当包括“服务访问路径”。

其次，要考虑消息的内容，这需要用 message-body 来传递服务层的信息，这需要说明目标服务接收怎样格式的服务脚本。因此服务元数据中应当包括“服务脚本规格定义”。

第三，要考虑消息的类型。服务的“消息处理特性”对生成 HTTP 消息是有影响的。服务有两种消息处理特性：单输入型——每接收到一个请求消息，就执行一次，如 CGI 程序；多输入型——接收多个请求消息后再执行一次，如独立服务器。处理这样，对于有多个输入的服务来说，就可能有两种不同的输入/输出参数到 HTTP 消息的映射机制（如图 4-11）。因此，服务元数据中应当包括“消

息处理类型”。

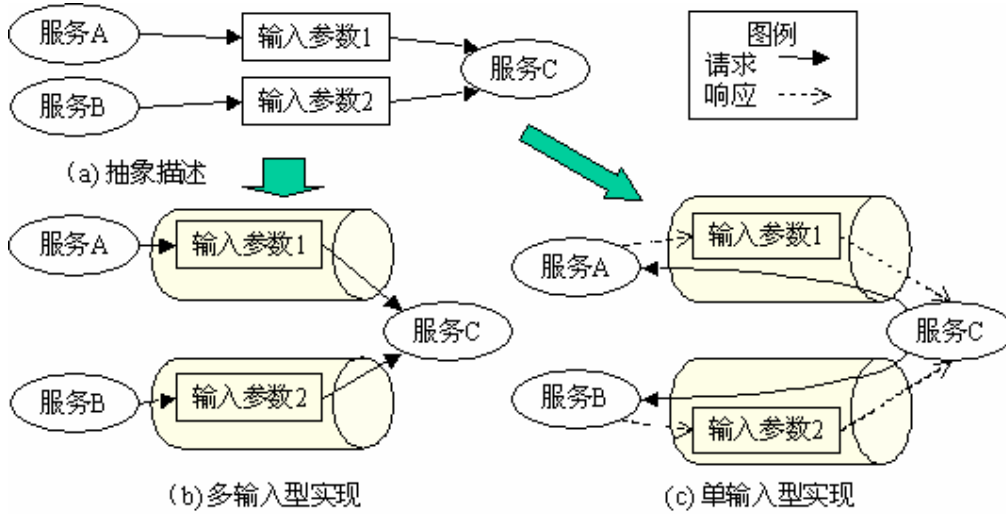


图 4-11 根据不同的操作特性映射服务

4.2.5 服务元数据的应用举例

4.2.5.1 查询图层获取服务

图层的空间元数据格式（如图 4-12）与上面描述的图层获取服务的元数据格式具有相同的部分。而这两部分信息的匹配恰恰表明了空间信息与空间信息获取服务的匹配。

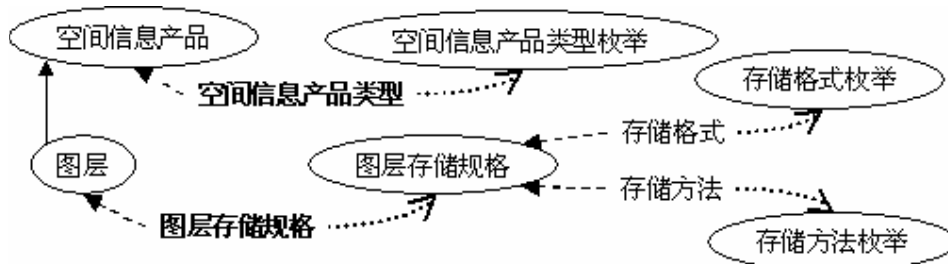


图 4-12 描述中空间信息规格的空间元数据

图中空间信息产品枚举可以包括“图层信息”、“地图”和“符号表”等。同类的空间信息产品可以有不同的“存储规格”，因此需要有不同的信息获取服务来访问。例如，图层的“存储规格”包括“存储方法”和“存储格式”两部分。

“存储方法”描述获取图层时所必需的访问协议。不同的物理存储如 FTP 服务器、WEB 服务器、本地文件系统或数据库服务器存储，有不同的访问接口。

“存储格式”描述图层在物理存储中的具体组织。图层一般以文件或数据库存储的形式存在，图层文件有很多格式，如 SHAPE (ESRI), MIF (MapInfo), GML DTD (OGC), GML Schema (OGC), GML RDF (OGC), GUM (Geo-Union) 等，数据库存储也有不同的访问接口，如 MO 驱动的 SDE(ESRI)，或 GeoUnion5.0 的客户端构件等。

4.2.5.2 支持服务之间的通信参数匹配

服务之间的通信应当满足参数匹配的特性。参数匹配就是看一个服务的输出能否作为另一个服务的输入，只有两种情况可以接受：采用相同空间信息类型定义；输出参数的空间信息类型定义是从输入参数的空间信息类型中派生的。这一点也是可以通过在服务元数据库中的查询完成的。

4.3. OSIIF 的服务过程

服务过程说明了 OSIIF 服务模型如何向用户提供服务。一个服务过程描述了用户的一个具体任务，其中包括两个主要问题：用户如何描述服务过程；服务如何具体执行以完成服务过程。OSIIF 服务模型基于服务脚本，分别对两个问题提供了一定的支持：

- 提供一定的服务脚本，通过协调服务和导航服务，并辅助用户创建描述服务过程的服务脚本；
- 根据 4.2.4.1 中描述的层次化通信机制，通过服务导航服务和脚本，辅助服务程序自动创建并传递 HTTP 消息（参见 4.2.4.2， 4.2.5.2）。

本小节主要讨论如何 OSIIF 提供哪些服务脚本，以及如何辅助用户创建服务脚本。

4.3.1 协调服务的脚本分类

用户通过服务脚本向协调服务提交任务，其中要说明用户想得到什么信息，要用到哪些信息，哪些功能服务，以及信息与服务的关系。考虑到 Internet 用户具有不同的需求、对 OSIIF 不同的了解程度、以及在任务求解过程中不同的参与程度等因素，OSIIF 的协调服务支持多种具有不同描述能力的服务脚本。图 4-13 表明，它支持两种服务脚本：任务说明书和信息流动图。而任务说明书是可以转换成信息流动图的。

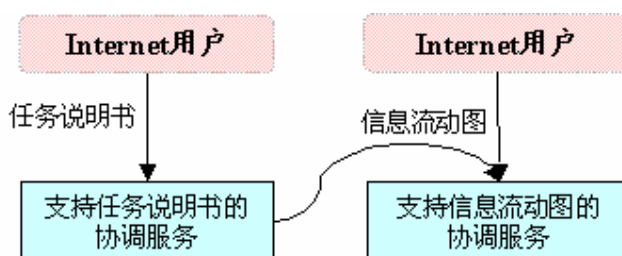


图 4-13 OSIIF 的协调服务支持两种服务脚本

4.3.1.1 任务说明书

空间信息获取活动中，有一些很常用的信息获取服务过程，例如获取图层和获取地图等。使用这些服务的用户往往又不关心框架的具体实现，他们只希望使用简单的服务接口和确定的服务过程。因此可以定制一个协调服务：通过编程直接将服务过程固定在该协调服务中，形成简单而固定的服务接口。

例如，表 4-3 是一个获取空间信息产品的协调服务的任务说明书，它可以找到相应的空间信息产品，以及相应的获取服务，最终返回用户希望的结果。

表 4-3 一个任务说明书的例子

<任务说明书>
<类型>图层</类型>
<名称>北京交通</名称>
<返回格式>GML2</返回格式>
</任务说明书>

又如，OGC 的 WMS (Web Map Server) ^[4.10]，WFS (Web Feature Server) ^[4.11] 的接口定义也都是是一种特殊的任务说明书。

这种协调服务的实现可以采用两种策略：直接实现；或者将用户的任务描述映射到信息流动图，再由可以处理信息流动图的服务完成（参见 4.3.1.2）。

4.3.1.2 信息流动图

空间信息获取活动中，有一些服务过程是不太确定的。用户不关心任务的具体实现，只想说明信息如何流动，这样就可以比较抽象地描述服务的执行过程。这就需要一种灵活的对任务的抽象描述方法，以及支持这种描述的协调服务。因此就有了信息流动图。

信息流动图是一个有向图，其中包括节点和有向弧。它用于描述信息在 OSIIF 中传递的过程。节点对应 OSIIF 涉及的角色，而有向弧对应 OSIIF 中的空间信息流动。

信息流动图中的节点分三类（见表 4-4）。其中注意两点：首先，用户节点和服务节点都可以有多个输入和输出；其次服务资源可能在一个任务中的不同场合中重复出现。在一个任务中，需要仔细考虑重复使用的服务资源，如果服务资源重复出现于相同环境的场合，就需要考虑简并子任务；如果场合不同，那就需要把这两次使用区分开。

表 4-4 信息流动图中的节点

分类	说明	标记
用户节点	代表用户，可以发起请求，并接收空间信息	紫色椭圆形
服务节点	代表服务资源的一次使用，可以处理请求，传递请求，获取或处理空间信息	蓝色矩形
信息节点	代表一个信息资源	绿色圆柱形

信息流动图中有两种弧分为两种（见表 4-5）。

表 4-5 信息流动图中的弧

分类	说明	标记
信息访问弧	描述某个空间信息产品被某个服务资源直接访问了	兰色点划线
信息传递弧	描述服务资源以及用户之间的传递了空间信息	黑色实线

信息流动图可以用图形方式（如图 4-14）或使用 Backus-Naur 范式说明其 XML 表达格式（如表 4-6）。

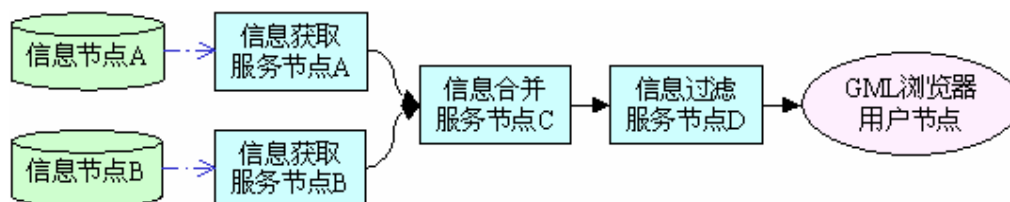


图 4-14 信息流动图

表 4-6 信息流动图的 Backus-Naur 范式定义

```

信息流动图 ::= 资源 节点 , 弧
*****资源*****
资源 ::= 用户 服务资源+ 信息资源+
用户 ::= ID 参数表
服务资源 ::= RID 参数表
信息资源 ::= RID
*****节点*****
节点 ::= 用户节点 服务节点+ 信息节点+
用户节点 ::= ID 用户 ID
服务节点 ::= ID 服务资源 RID
信息节点 ::= ID 信息资源 RID
*****弧*****
弧 ::= 信息访问弧+ 信息传递弧*
信息访问弧 ::= ID 信息节点 ID 服务节点 ID
信息传递弧 ::= ID 发送服务节点 接收服务节点
*****辅助信息*****
参数表 ::= 输入参数* 输出参数*
输入参数 ::= ID 参数类型
输出参数 ::= ID 参数类型
发送服务节点 ::= 节点 ID 输出参数 ID
接收服务节点 ::= 节点 ID 输入参数 ID

```

4.3.2 任务说明书的辅助生成

4.3.2.1 生成任务说明书

任务说明书是由特定的规格，不同的协调服务支持不同的任务说明书。生成任务说明书的要点就在于按照协调服务提供的要求补足信息。这种情况下，用户一般不需要了解协调服务的具体实现，更不必关系 OSIIF 中的资源分布情况。

4.3.2.2 生成信息流动图

信息流动图是以信息为中心的。它通过节点和弧描述了信息的流向，由于同一个服务资源可能在信息流动图的不同部分起不同的作用——其输入/输出的信息是不同的，所以需要不同的节点来区分之。信息流动图的生成过程大致分三步：

1. 确定其中包含哪些信息资源。用户可以直接说明信息资源的 RID，或者通过提出对信息资源的查询条件，在信息导航服务的帮助下查到 RID。信息导航服务可以支持多种信息发现与获取协议，如 Z39.50 协议或基于 XQuery 的查询协议。
2. 确定其中包含哪些信息获取服务并且描述信息访问弧。用户可以直接用 RID 指定信息获取服务，也可以依据信息资源的 RID，在信息导航服务和导航服务的帮助下自动获得相应服务的 RID。这个过程参见 4.2.5.1 的论述。因此，用户在生成信息流动图时，一般不必指定信息获取服务。
3. 确定其中包含哪些信息处理服务并且描述信息传递弧。用户可以直接用 RID 指定并且描述信息传递弧。这个过程是比较复杂的，其中要考虑如何找到具有恰当能力的处理服务，服务之间通信参数匹配（参见 4.5.2.2）等问题。因此，这一步需要人的参与：或者由用户预先指定，或者由用户通过与导航服务的交互动态获得。

4.4. 本章小结

根据提供服务的实体在粒度上的差异，我们认为服务可以划分成五个层次：应用层、系统集成层、领域原子层、构件层、对象层。OSIIF 是一种 Internet 上的大粒度的分布计算平台，其目标是帮助用户有效地在 Internet 上查找、获取并使用空间信息。用户可以基于这个计算平台构造自己在 Internet 上的应用。

OSIIF 服务可以分成五种：属于系统集成层的协调服务，以及属于领域原子层的信息导航服务、服务导航服务、空间信息获取服务和空间信息处理服务。

我们在 OSIIF 中提出并设计了“服务元数据”以管理并使用分布在 Internet 上的众多服务资源。它应当包括如下方面的信息：

- 提供服务标识来唯一地标记服务
- 提供输入/输出参数和服务分类以描述服务的能力
- 提供服务的接口描述以支持服务之间的自动通信。

OSIIF 以服务脚本的形式向用户提供服务。主要有两种服务脚本：任务说明书和信息流动图。其中，信息流动图表示服务过程中信息的抽象流动过程，服务在理解它的前提下，可以利用导航服务将抽象操作映射为具体操作，形成动态的服务协作来完成服务过程。同时，我们也说明了如何辅助生成这两种服务脚本。

第5章 用 OSIIF 构建空间信息领域服务体系

5.1. 增加空间信息产品实例——注册图层

OSIIF 支持信息提供者加入新的信息资源,注册空间信息产品的策略在 4.1.3.1 已经有所描述,这里以图层为例,说明如何注册一个框架中已有类型的空间信息产品的实例。

首先要考虑图层的元数据设计,其中要说明图层的规格和访问路径,这样才能支持用户获取图层的自动化。图 5-1 是相关图层元数据的设计,表 5-1 是其中图层访问路径特有属性的 BNF (Backus-Naur Form) 描述。

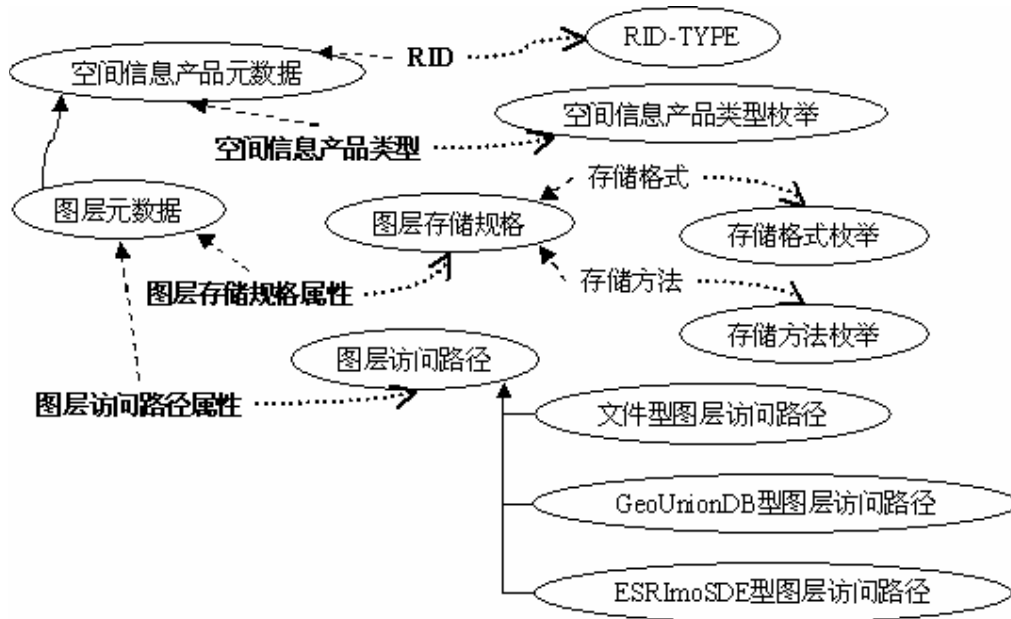


图 5-1 支持图层获取自动化的图层元数据 (RDF 图)

表 5-1 图层访问路径特有属性的 BNF 描述

文件型图层访问路径 ::=	IPv4 地址	协议类型	路径	文件名
GeoUnionDB 型图层访问路径 ::=	IPv4 地址	认证信息	数据源名	图层名
ESRImoSDE 型图层访问路径 ::=	IPv4 地址	认证信息	数据库名	实例名称
		拥有者	表名	空间字段名 空间特征类型
协议类型 ::=	"HTTP"		"FTP"	"FILE" ...
IPv4 地址 ::=	IP		PORT	
认证信息 ::=	USER		PASSWD	
空间信息产品类型枚举 ::=	"图层信息"		"地图"	"符号表" ...
存储方法枚举 ::=	"HTTP"		"FTP"	"FILE" "DATABASE" ...
存储格式枚举 ::=	"GML-XML-SCHEMA"		"GML-XML-DTD"	"GML-RDF-SCHEMA"
	"MIF"		"SHAPE"	"GeoUnionDB" "ESRImoSDE" ...
中介格式枚举 ::=	"GML-XML-SCHEMA"		"GML-XML-DTD"	"GML-RDF-SCHEMA"

表 5-2, 5-3 分别举例说明了如何书写与图层访问路径相关的元数据。

表 5-2 一个以 GML2 (XML Schema) 形式存储的图层的部分元数据

```
<图层元数据>
  <RID> http://gis.pku.edu.cn/gis/:I:2001051101 </RID>
  <图层存储规格属性>
    <存储方法> HTTP </存储方法>
    <存储格式> GML-XML-SCHEMA </存储格式>
  </图层存储规格属性>
  <图层访问路径属性>
    <文件型访问路径>
      <协议类型> HTTP </协议类型>
      <主机地址属性>
        <IPv4 地址>
          <IP>   gis.pku.edu.cn   </IP>
          <PORT> 80 </PORT>
        </IPv4 地址>
      </主机地址属性>
      <路径>   gis\samples\gml2   </路径>
      <文件名> city </文件名>
    </文件型访问路径>
  </图层访问路径属性>
  ...
</图层元数据>
```

表 5-3 一个存储在 Geo-Union 5.0 data server 中的图层的部分元数据

```
<图层元数据>
  <RID> http://gis.pku.edu.cn/gis/:I:2001051104 </RID>
  <图层存储规格属性>
    <存储方法> DATABASE </存储方法>
    <存储格式> GeoUnionDB </存储格式>
  </图层存储规格属性>
  <图层访问路径属性>
    <GeoUnionDB 型访问路径>
      <主机地址属性>
        <IPv4 地址>
          <IP>   168.168.168.217   </IP>
          <PORT> 2000 </PORT>
        </IPv4 地址>
      </主机地址属性>
      <认证信息属性>
        <认证信息>
          <USER> demo </USER>
          <PASSWD> demo2001 </PASSWD>
        </认证信息>
      </认证信息属性>
    </GeoUnionDB 型访问路径>
  </图层访问路径属性>
</图层元数据>
```

```

    </认证信息>
    </认证信息属性>
    <数据源名> DigitalPKU </数据源名>
    <图层名> 燕园地块 </图层名>
    </GeoUnionDB 型访问路径>
    </图层访问路径属性>
    ...
  </图层元数据>

```

5.2. 增加空间信息产品类型—地图

OSIIF 也支持信息提供者加入新的空间信息产品类型。这涉及到用户如何利用 OSIIF 的空间信息模型描述空间信息产品, 以及用户如何组织新空间信息产品的元数据。对于前面一个问题, 可以参照 3.5 中规定的步骤完成; 对于后一个问题, 也要考虑空间信息产品的分类与访问路径设计, 这可以参照图层元数据来设计。下面主要讨论前一个问题。

5.2.1 明确问题域——定义地图

“地图”是向用户提供空间信息的可视化产品。它由若干图层构成的。它可以指派各个图层中实体的画法, 描述图层的上下关系。

5.2.2 提取共识——将地图纳入空间信息模型

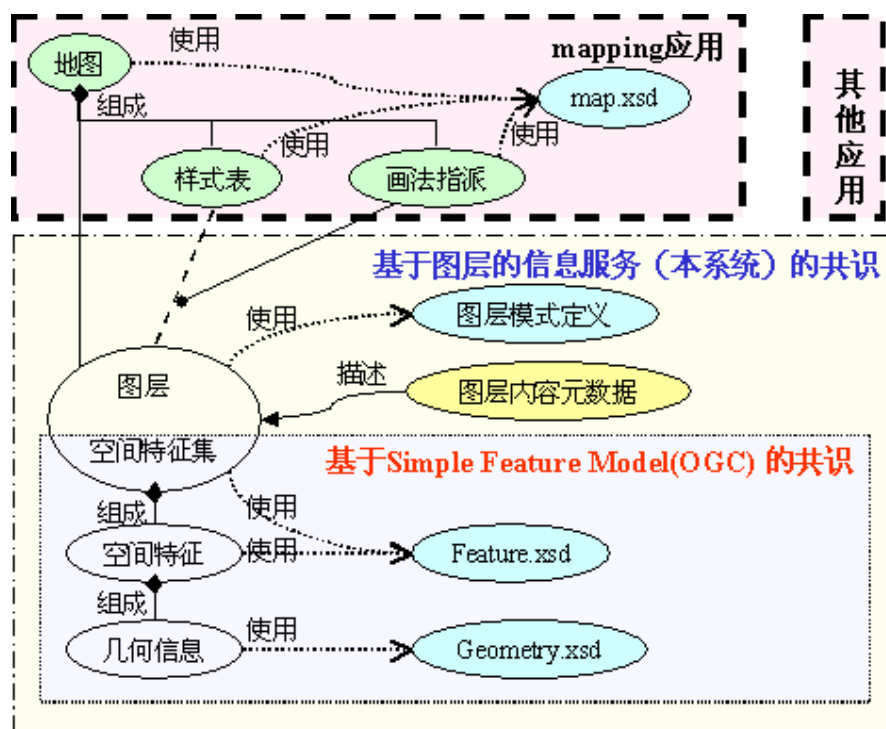


图 5-2 OSIIF 的空间信息共识及 mapping 应用中的共识

图 5-2 描述了 OSIIF 的共识信息体系与地图等应用信息之间的关系。其中包括三个层次的共识，首先是基于 OGC 的 Simple Feature Model 的共识，它提供了定义空间信息的基本要素；其次是 OSIIF 的共识，它提供了图层的概念，帮助一般用户理解空间信息的存储组织；最后是 mapping 应用的共识，它提供了地图、样式表和画法指派等概念。

通过识别地图与 OSIIF 的基于图层的共识体系的关系，可以提取出与地图相关的概念：地图、样式表和画法指派。因为图层是一种比较独立的资源，因此可以采用 Xlink 技术定义地图：地图是一种扩展连接（Extended Link），而图层和样式表是定位型连接（Locator Link），画法指派是一种弧型连接（Arc Link），每个画法指派可以关联一个样式表和一个图层。一个地图可以用图形方式表示，如下图所示。

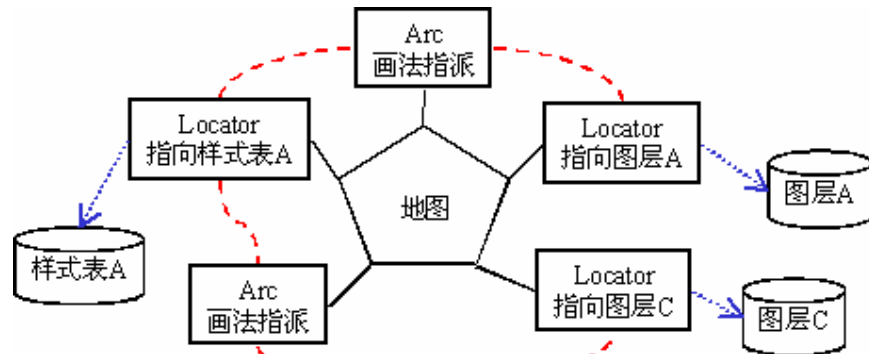


图 5-3 地图的 Xlink 定义

根据 OSIIF 提供的基础对象体系以及上面地图的 XLink 定义，可以提取地图及相关信息的抽象对象体系（如下图）。其中关于样式表的设计参照了 OGC 的 Styled Layer Descriptor ^[5.1]。

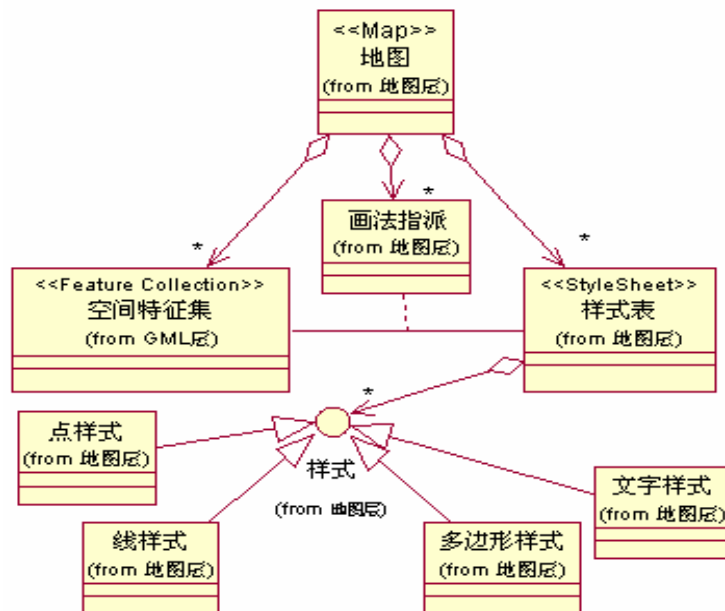


图 5-4 地图的抽象对象体系（概念级表示）

5.2.3 设计地图的对象系统

依据 OSIIF 的基于共识的分级空间信息模型的提供的对象表示体系，参考 GML2.0 的对象定义，可以把地图的抽象对象体系映射到实际的对象体系中，最终形成地图的对象级表示（如下图）。

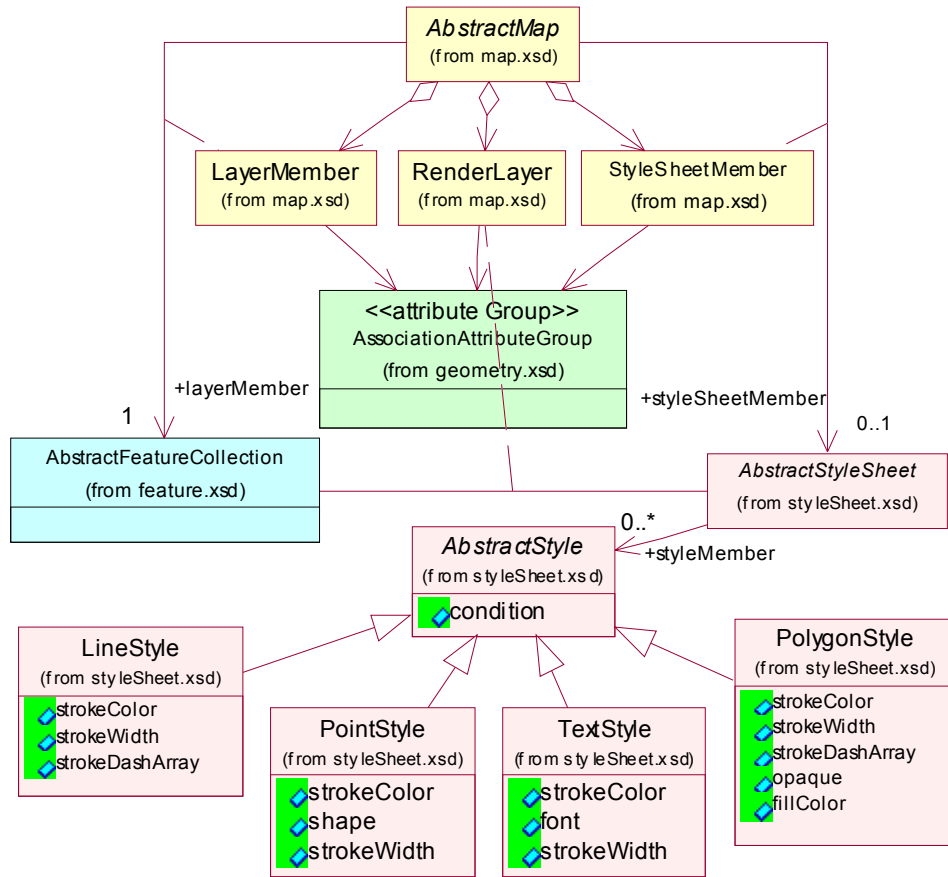


图 5-5 地图的对象级表示

同时，也可以依据地图的对象级表示设计中中介格式。这具体表现为地图的两个 XML Schema 定义文件（map.xsd，和 styleSheet.xsd）对 GML2.0 的依赖。

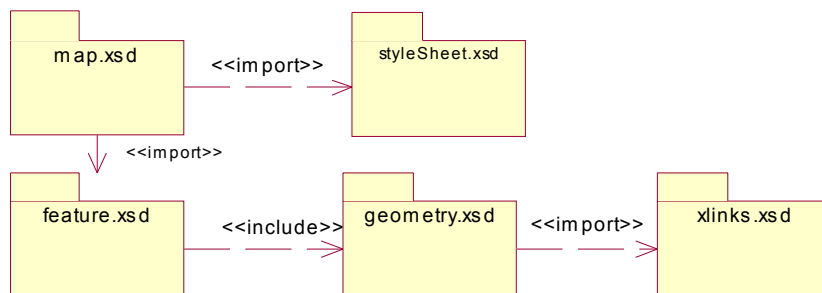


图 5-6 地图定义文件与 GML2.0 定义文件的依赖关系

5.2.4 设计地图的元数据

设计地图的元数据要考虑空间信息产品的分类与访问路径。地图应用中要用到两种的空间产品，“地图”和“样式表”。因为他们与遗产系统关系不大，可以直接规定他们的存储规格为：采用文件型存储方式，而且就用 XML Schema 格式作为存储规格。

5.3. 增加服务资源

5.3.1 直接获取空间信息产品

最简单的图层信息资源就是把空间信息产品存储为中间格式的文件，然后把它连接到 Web Server 上，用户就可以直接利用其 URL 获取它。例 5.1 中图层的访问路径就是一个图层的 URL^[5.2]。

5.3.2 动态获取空间信息产品

有的空间信息产品的存储规格并不能被用户直接使用，因此需要转换成用户可理解的中介格式，这可以通过提供相应的信息获取服务来达到。4.1.3.4 中已经讨论了动态获取信息的过程。这里以 GeoUnionDB 图层获取服务为例，说明如何实现一个支持动态获取空间信息产品的服务。

5.3.2.1 设计 GeoUnionDB 图层获取服务的元数据

GeoUnionDB 图层获取服务是一种图层获取服务。因此它应当遵循图层获取服务的元数据定义（如图 5-7），主要应当包括如下项目：

- RID
- 基本服务分类
- 空间信息产品类型
- 图层存储规格属性
- 输入参数。其值域为“参数”，而“参数”又包括以下项目：
 - 参数 ID
 - 参数类型
- 输出参数
- 服务接口描述属性。GeoUnionDB 图层获取服务采用“HTTP 服务接口描述”，包括以下项目：
 - 服务脚本规格定义
 - 消息处理特性
 - 服务访问路径

按照元数据格式定义，图层提供者可以很方便地编写该服务的服务元数据内容（如表 5-4）。

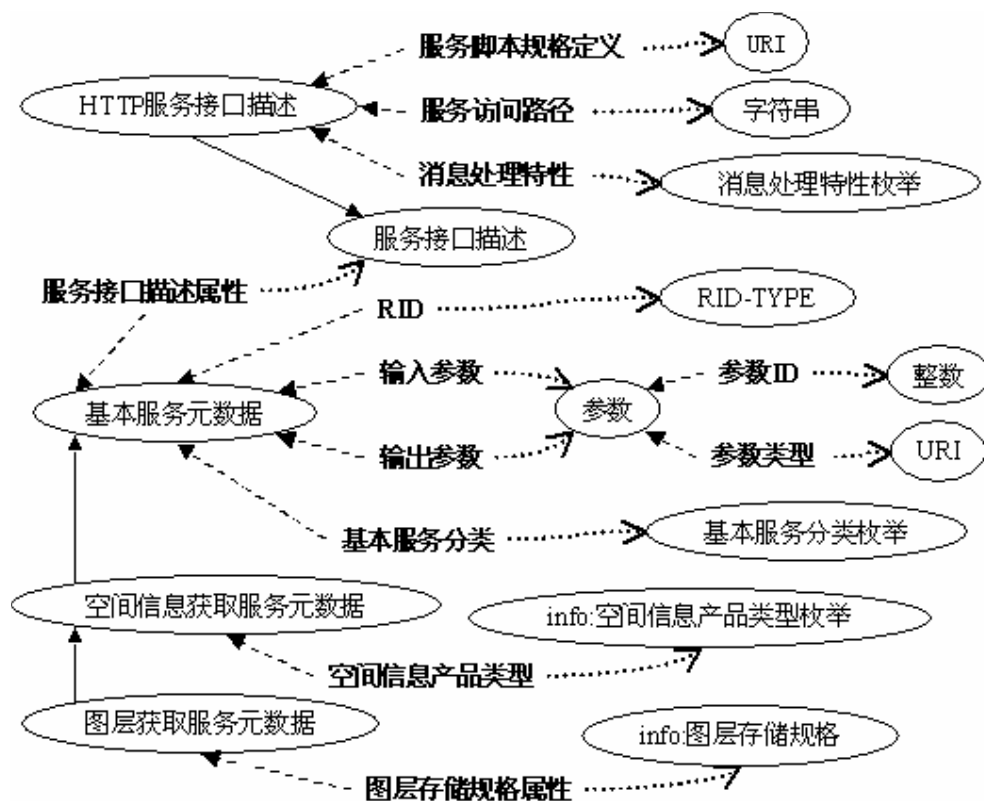


图 5-7 图层获取中相关的服务元数据 (RDF 图)

表 5-4 一个图层获取服务的元数据举例

```

xmlns:info = http://gis.pku.edu.cn/gis/samples/meta-info
<图层获取服务元数据>
  <RID> http://gis.pku.edu.cn/gis/ : S : 2001051105 </RID>
  <基本服务分类> 空间信息获取服务 </基本服务分类>
  <空间信息产品类型> 图层 </空间信息产品类型>
  <图层存储规格属性>
    <info:图层存储规格>
      <info:存储方法> DATABASE </info:存储方法>
      <info:存储格式> GeoUnionDB </info:存储格式>
    </info:图层存储规格>
  </图层存储规格属性>

  <输入参数>
    <参数>
      <参数 ID> 1 </参数 ID>
      <参数类型>
        http://gis.pku.edu.cn/gis/samples/OSIIF-InfoProduct-Layer.rdf#GeoUnionDB 型访问路径
      </参数类型>
    </参数>
  </输入参数>

```

```

    <参数>
      <参数 ID> 2 </参数 ID>
      <参数类型>
http://gis.pku.edu.cn/gis/samples/OSIIF-InfoProduct.rdf#中介格式枚举
      </参数类型>
    </参数>
  </输入参数>
  <输出参数>
    <参数>
      <参数 ID> 1 </参数 ID>
      <参数类型> http://www.opengis.net/gml#AbstractFeatureCollection </参数类型>
    </参数>
  </输出参数>

  <服务接口描述属性>
    <HTTP 服务接口描述>
      <服务脚本规格定义>
http://gis.pku.edu.cn/gis/samples/ServiceTemplate1.rdf
      </服务脚本规格定义>
      <消息处理特性> 单输入型 </消息处理特性>
      <服务访问路径>
http://gis.pku.edu.cn/gis/samples/GeoDBRetriever.exe
      </服务访问路径>
    </HTTP 服务接口描述>
  </服务接口描述属性>
  ...
</图层获取服务元数据>

```

服务脚本规格定义的规格如图 5-8，其实例参见^[5.3]。通过这个文件和服务元数据可以帮助服务自动生成 HTTP 请求消息。其中 Request1 表明服务可以接收的一种请求的服务层消息的格式。这样只要把输入消息填写进该服务层消息即可。

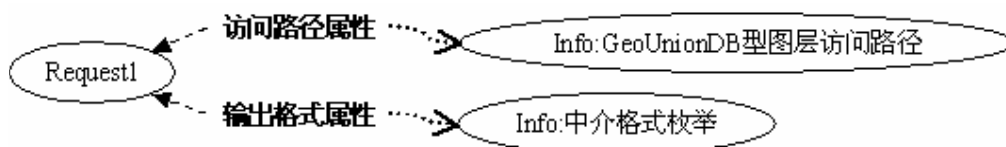


图 5-8 服务脚本规格定义举例（RDF 图）

5.3.2.2 设计图层获取服务

图层获取服务的一个最简单的也最有实践意义的实现就是在利用 Web Server CGI 实现，因为这样可以最大限度地达到在 Web 上的可达性，避免防火墙等信息过滤因素的干扰。

由于 Geo-Union5.0 data server 不能被直接访问，需要通过特定的客户端构件（如图 5-9）来访问 Geo-Union5.0 data server 中的空间信息。

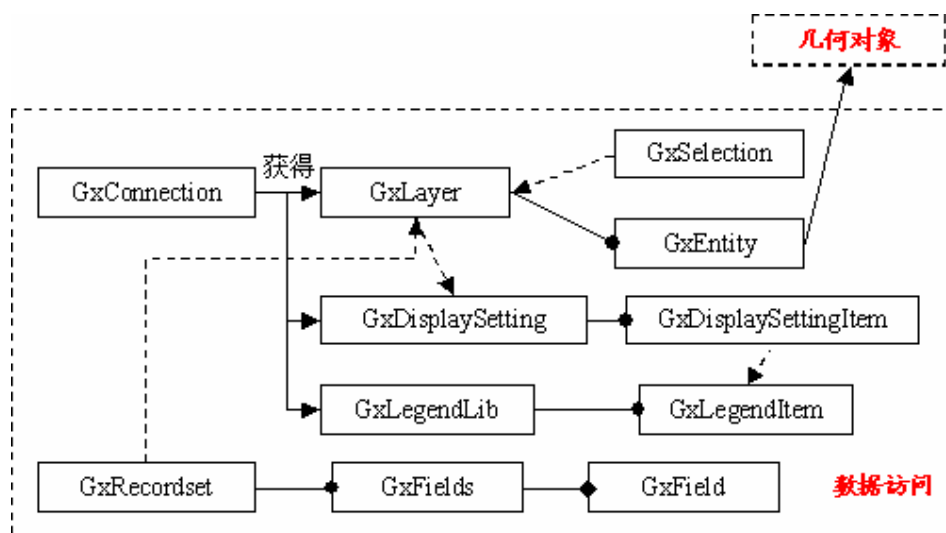


图 5-9 Geo-Union5.0 的客户端空间信息访问构件结构

这样，问题就转化为如何使用构件生成中介格式信息的问题，这只需要规定一些转换规则即可，下面是信息转换的基本算法描述。

1. 根据 Request 消息输入的服务层消息，服务可以得到一些可理解的输入信息——“GeoUnionDB 型图层访问路径”和“输出格式”（参见 5.1）。
2. 服务利用输入信息，用 GxConnection 建立与 Geo-Union5.0 data server 的连接
3. 服务利用输入信息，用 GxConnection 获得指定的图层 GxLayer
4. 服务读取 GxLayer 的信息，按照指定的输出格式进行信息转换。本例的服务只有一种输出——GML-XML-SCHEMA。首先形成映射规则，创建新类型：
 - a) 图层名: GxLayer.name => 新建 gml: AbstractFeatureCollectionType 的派生类
 - b) 实体: GxEntity => 新建 gml: AbstractFeatureType 的派生类
 - c) 几何信息: 几何对象 => gml: AbstractGeometry 的派生类
 - d) 属性信息: GxFields (名称, 类型) => 新建 gml: AbstractFeatureType 的派生类的子 element。
5. 然后按照映射规则，使用 GxLayer, GxEntity, GxField, GxGeoemtry 将图层的具体内容转换到目标格式。
6. 将生成的以目标格式组织得结果以 Response 消息反馈给请求者。

5.4. 一个框架服务的应用——获取图层

应用问题很简单，就是一个 Internet 上的用户想在 Internet 查找一个名字叫“北京地图”的图层，这个图层要以 GML-XML-SCHEMA 形式返回。如果能用一般浏览器获得图层信息资源的 XML 表示，就可以在 Internet 上使用特制的浏览器、或在普通浏览器中安装插件来观看图层了。而这要求从设计从浏览器发给 web server 的信息，以便 web server 能理解并将请求转发给相应的处理程序，并由该处理程序再把结果返回给浏览器。

5.4.1.1 初始状态

对于 Internet 用户来说，他只知道需要一个名字为“北京交通”的图层，还知道他可以使用一个可以按名获取图层的协调服务（如图 5-10）。

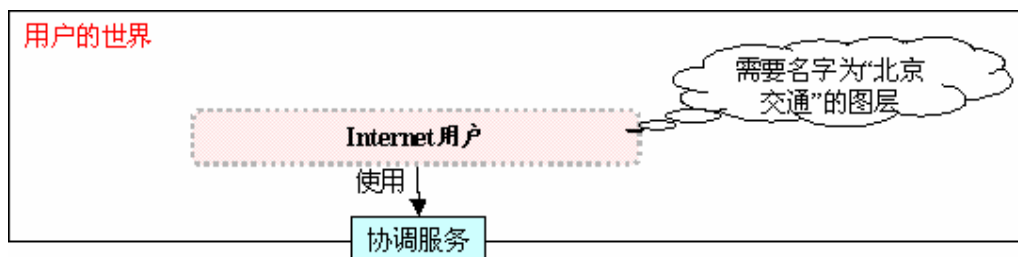


图 5-10 Internet 用户掌握的资源

对于这个协调服务来说，它可以把用户的请求转化为对服务导航服务的请求，它知道服务导航服务的地址，并且它可以使用服务导航服务和信息导航服务。当然两个导航服务都可以访问相应的元数据库（如图 5-11）。

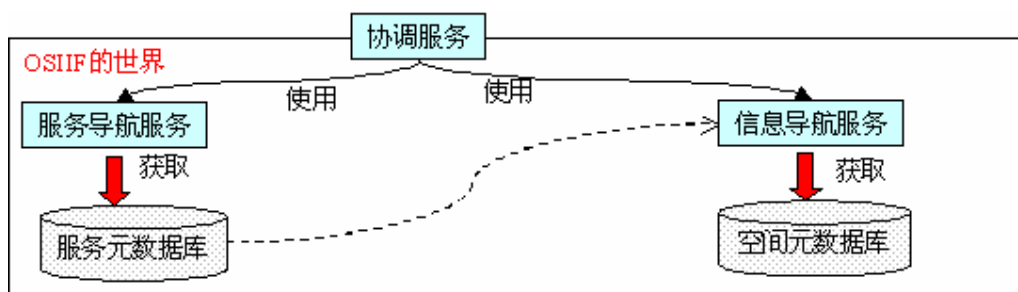


图 5-11 协调服务掌握的资源

5.4.1.2 查找信息

当用户发出查询名字为“北京交通”图层的请求时，协调服务将用户请求转化为对信息导航服务的请求，查找满足要求的图层，并获取它们的元数据。这时协调服务就所掌握的信息为（如图 5-12）：有两个图层资源满足条件，一个是以 GeoUnionDB 格式存储的，另一个是以 MIF 格式存储的。

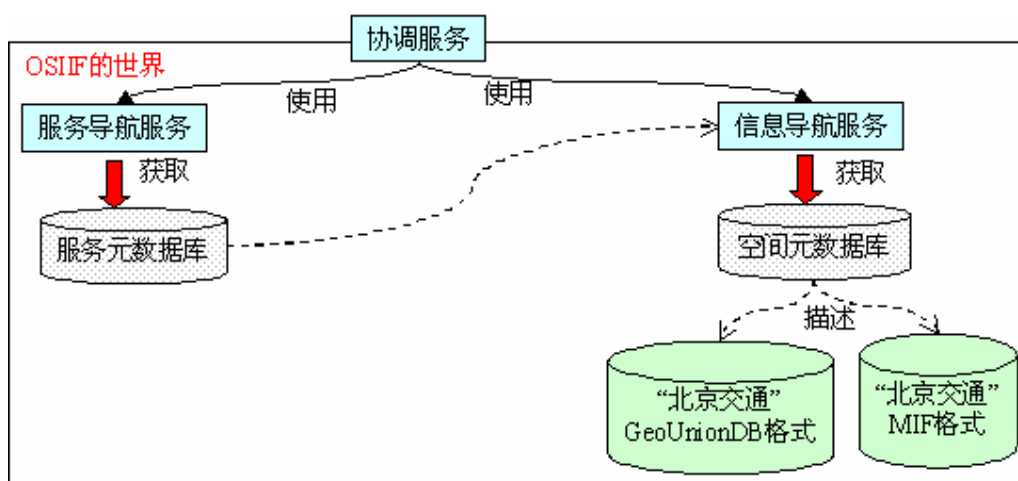


图 5-12 协调服务查找信息

5.4.1.3 生成信息流动图

当协调服务掌握了可用的空间信息产品的元数据列表后,就可以使用服务导航服务来查找能够访问这些空间信息产品的服务。具体匹配方法可以参考 4.2.5.1 中说明的策略。本例中协调服务只找到一个可以访问 GeoUnionDB 信息的空间信息获取服务,于是就选择使用 GeoUnionDB 格式的“北京交通”图层为数据来源,用刚找到的服务来获取它,从而形成信息流动图(如图 5-13)。

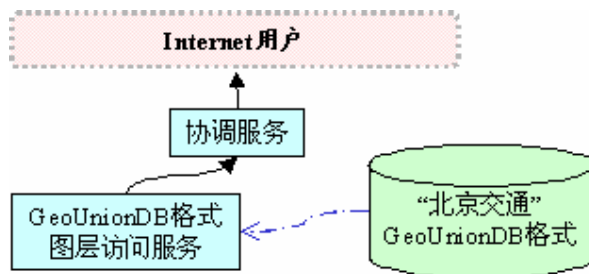


图 5-13 本例的信息流动图

当然,匹配可能得到多条可能路径,例如同时找到 GeoUnionDB 格式的图层获取路径和 MIF 格式的图层获取路径,两种路径都能满足用户需要。路径的选择可以由协调服务或用户决定。此外,用户还可以跳过前几步,直接以信息流动图的方式提交任务,从而加强对服务过程的控制并提高服务效率。

5.4.1.4 执行服务过程

通过理解上面生成信息流动图,首先形成了控制信息(如表 5-5)。随后协调服务按如下步骤执行

1. 将控制信息发送给特定的信息获取服务——GeoUnionDB 格式图层访问服务
2. 信息获取服务理解请求中的空间信息的访问路径并据此获取指定的空间信息——GeoUnionDB 格式的图层“北京交通”
3. 信息获取服务将找到的信息按指定格式(GML-XML-SCHEMA)返回给协调服务
4. 最后协调服务将结果转交给 Internet 用户(如图 5-14)。

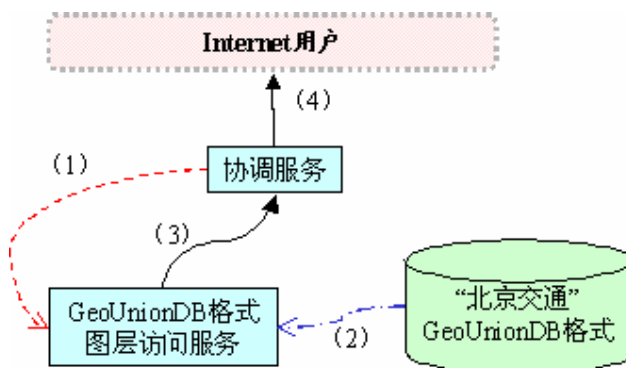


图 5-14 信息流动图的具体执行

表 5-5 发送给信息获取服务的服务层消息

```

<?xml version="1.0" encoding="UTF-8"?>
<Request1 xmlns:info="http://gis.pku.edu.cn/samples/rdf/info#">
  <访问路径属性>
    <info:GeoUnionDB 型访问路径>
      <info:主机地址属性>
        <info:IPv4 地址>
          <info:IP> 168.168.168.217 </info:IP>
          <info:PORT> 2000 </info:PORT>
        </info:IPv4 地址>
      </info:主机地址属性>
      <info:认证信息属性>
        <info:认证信息>
          <info:USER> demo </info:USER>
          <info:PASSWD> demo2001 </info:PASSWD>
        </info:认证信息>
      </info:认证信息属性>
      <info:数据源名> DigitalPKU </info:数据源名>
      <info:图层名> 燕园地块 </info:图层名>
    </info:GeoUnionDB 型访问路径>
  </访问路径属性>
  <输出格式属性>
    <info:中介格式枚举> GML-XML-SCHEMA </info:中介格式枚举>
  </输出格式属性>
</Request1>

```

5.5. 本章小结

本章通过具体的例子说明了 OSIIF 的工作机制。

- 以添加图层为例，说明如何在 OSIIF 中增加并注册一个已有的空间信息产品的实例。
- 以添加地图为例，说明如何在 OSIIF 中创建一个新的空间信息产品
- 以添加图层获取服务为例，说明如何在 OSIIF 中增加并注册一个空间信息领域原子服务，如何设计和填写相应的服务元数据。
- 以获取图层为例，说明 OSIIF 的协调服务是如何通过服务过程完成用户的任务。

第6章 结论与展望

6.1. 结论

本文针对目前空间信息共享中的可达性、可用性和易用性三个问题,提出了开放空间信息互操作框架 OSIIF,并且面向 Semantic Web,以开放、动态、简单、自动和灵活为原则,有重点地研究了 OSIIF 的体系结构、信息模型和服务模型。

在 OSIIF 中,我们把信息和服务统一看作 Internet 上的资源。信息资源可以分为空间信息产品和元数据库两大类;服务可分为协调服务,导航服务和领域原子服务三大类。互操作问题体现在信息获取活动和服务协作活动之中。元数据提供了对资源的描述。

我们采用 XML 和 RDF 技术来描述 OSIIF 中基于共识的分级空间信息交换机制。通过界定空间信息领域中的共识(概念模型级共识,领域级共识和应用级共识),划分空间信息交换的层次(结构层,对象层和语义层),我们设计了空间信息交换的分级共识矩阵,具体说明 OSIIF 的空间信息的交换格式。

我们把服务按服务实体粒度划分成五个层次:应用层、系统集成层、领域原子层、构件层、对象层。OSIIF 是一种 Internet 上的大粒度的分布计算平台,包括五种服务:属于系统集成层的协调服务,以及属于领域原子层的信息导航服务、服务导航服务、空间信息获取服务和空间信息处理服务。

我们在 OSIIF 中提出并设计了“服务元数据”以管理并使用分布在 Internet 上的众多服务资源。它可以描述服务的唯一标识,服务的能力以及服务的接口描述。OSIIF 以服务脚本的形式向用户提供服务,包括任务说明书和信息流动图等两种服务脚本。

在 OSIIF 建设中,参与人员拥有很大的自由度:

- 服务提供者可以采用不同的先进技术实现服务,或者开发领域应用相关的增殖服务,提高框架运行的自动化程度,从各个方面提高它的服务能力;
- 空间信息提供者可以很方便地向其中增加新的空间信息,从而提高它的信息总量;
- 用户可以利用 OSIIF 中丰富的信息、服务资源开发应用,并且可以通过共享服务脚本来最大程度地分享开发经验。

6.2. 进一步的工作

OSIIF 的建设刚刚起步,不过它开放的特性非常有利于进一步发展。从建设 OSIIF 的角度来看,应当首先提供足够的资源:空间信息获取服务及空间信息产品,尤其是增加图层。作为最基本的空间信息产品,图层被广大用户和信息提供者广泛接受。在此之上,可以逐步考虑扩展信息获取服务能力,增加新的空间信息产品,开发各式各样的空间信息处理服务。

同时,很重要的一点就是设计好导航服务。系统的易用性对它有极强的依赖,没有好的导航服务就无法真正推广整个框架的应用。而导航服务不但考虑如何设计和组织元数据,还需要提供一个友好的用户界面。可能有很多户从来没有使用

空间信息的经验，导航服务的用户界面要充分考虑到与人们的常识结合。可视化用户界面是很重要的，比如说可视化的信息导航服务，可以帮助用户从世界地图到中国地图，再由中国地图找到北京地图，采用形象的手段辅助一般用户找到需要的信息。此外，也存在一些专业的用户，他们是空间信息的主要使用者，他们面临的问题是如何利用框架提供的信息和服务资源来开发更好的应用，以便解决空间信息相关的问题。提供一个形象而简易的辅助开发工具就是很重要的，比如实现一个服务脚本编辑工具，不但可以从目录中找到服务和信息的元数据，还可以通过共享已有的服务脚本或开发经验来提高开发效率，框架应用的难点往往就在于如何组织各种资源来完成任务。

信息安全也是信息共享中的重要因素。信息的提供者有保护自己知识产权的权利，尤其是把空间信息当作商品再网络上交易时，信息安全不可忽视。目前框架中的安全问题主要有两个解决途径：或者采用安全的通信协议实现服务之间的通信，例如 `https` 等；或者实现认证服务器机制，由认证服务器来负责验证用户的真实性，而具体服务只考虑向经过认证的用户提供服务。这些只是空间信息共享中的一部分安全问题，其他如“信任链”等问题还需要深入研究。

通过进一步完善 OSIIF，它将不但为网上空间信息的管理和导航提供有效的方法，而且对网上其他信息的管理和导航也会产生重要借鉴意义。

参考文献

- [1.1] Al Gore, The Digital Earth: Understanding our planet in the 21th century, 1998-1-31
- [1.2] 数字地球北京宣言, [EB/OL]
<http://www.digitalearth.net.cn/de99/isde-declaration.htm>, 1999-12-2
- [1.3] <http://www.semanticWeb.org>
- [1.4] OGC, Inc., A Request for Technology In Support of an OGC Web Services Initiative [EB/OL], <http://ip.opengis.org/ows/index.html>, 2001-3-23
- [1.5] Buehler K, Mckee L, editors. The Open GIS Guide (Third Edition) [EB/OL], OpenGIS Consortium, Inc., 1998-6
- [1.6] <http://www.opengis.org>
- [1.7] <http://www.statkart.no>
- [1.8] <http://www.esri.com/software/internetmaps/index.html>
- [1.9] <http://www.mapinfo.com/>
- [1.10] <http://www.intergraph.com/gis/gmwm/>
- [1.11] Max J. Egenhofer, Introduction: Theory and concepts [A], Interoperating Geographic Information Systems[C], Kluwer Academic Publishers, p.1,1999
- [1.12] ISO/TC211 WG4. Geospatial services N042 [EB/OL]. Open Geographic Datastore Interface (OGDI).
- [1.13] Bishr Y A, Pundt H, Probing the concept of information communities——A road towards semantic interoperability [EB/OL], <http://www.ncgia.edu.>, 1997
- [1.14] Michael F. Goodchild, Interoperating GISs, [EB/OL]
<http://www.ncgia.ucsb.edu/conf/interop97/report.html>, 1997-12.
- [1.15] 黄裕霞, 陈常松, 何建邦. GIS 互操作及其体系结构 [J]. 地理研究, 2000,19 (1): 86~92.
- [1.16] 黄裕霞, 陈常松, 何建邦. GIS 的互操作[A]. 中国地理信息系统协会. 中国海外地理信息系统协会 1998 年年会论文集[C], 1998.108~113.
- [1.17] Michael F. Goodchild, etc., "Interoperating Geographic Information Systems" [C], Kluwer Academic Publishers, 1999
- [1.18] Andrew S. Tanenbaum, Computer Networks (3rd Edition)[C], Prentice Hall, pp22~28, 1996
- [1.19] 罗英伟等, WebGIS 的构件设计 [J], 中国图形图象学报, 1999, Vol.4 (A): 79-84
- [1.20] 罗英伟, 基于 Agent 的分布式地理信息系统研究 [D], 北京大学博士研究生学位论文, 北京, 1999-6
- [1.21] Tim Berners-Lee, etc., Weaving the Web, The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor [C], Harper Collins Publishers Inc. 1999
- [1.22] www.semanticWeb.org
- [1.23] Introduction, [EB/OL], <http://www.semanticWeb.org/introduction.html>

- [1.24] Tim Berners-lee, James Hendler and Ora Lassila, The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities, [J], Scientific American, <http://www.sciam.com/2001/0501issue/0501berners-lee>, 2001-05-01
- [1.25] Robert Kahn, and Robert Wilensky, “A Framework for Distributed Digital Object Services“, [EB/OL], http://www.cnri.reston.va.us/tmp_hp/k-w.html, 1995-05-13
- [1.26] <http://www.handle.net>
- [1.27] OGC, Inc., The OpenGIS Abstract Specification Topic 5: Features (version 4) [EB/OL], <http://www.opengis.org/public/abstract/99-105r2.pdf>, 1999
- [1.28] OGC, Inc., Interoperation Program, [EB/OL], <http://ip.opengis.org/>
- [1.29] Jeff de La Beaujardiere (editor), Basic Services Model Draft Candidate Implementation Specification 0.0.8 [EB/OL], <http://www.opengis.org/techno/discussions/01-022r1.pdf>, 2001-03-05
- [1.30] ISO/CD 19119 (ISO TC 211 N 1044) Geographic Information – Services, [EB/OL] <http://www.statkart.no/isotc211/scope.htm#19119>, 2001-01-29
- [1.31] ISO/CD 19115 (ISO TC 211 N 1024) Geographic information – Metadata, [EB/OL] <http://www.statkart.no/isotc211/scope.htm#19115>, 2001-01-29
- [1.32] Microsoft, DCOM Technical overview, Microsoft white paper, 1996
- [1.33] 丁力, DCOM 初探, <http://gis.pku.edu.cn/gis/TR/19981120-DCOM.doc>, (研究小组技术报告)
- [1.34] OMG, The Common Object Request Broker: Architecture and Specification, Version.2.2., February, 1998
- [1.35] <http://java.sun.com/products/javabeans/>
- [1.36] 丁力, 汪小林, 许卓群, 数据库在地理信息系统中的应用研究 [J], 计算机应用与软件, (vol.18).pp.1-pp.4, 2000-1

- [2.1] Buehler K, Mckee L, editors. The Open GIS Guide (Third Edition) [EB/OL], OpenGIS Consortium, Inc., 1998-06

- [3.1] Tim Berners-lee, James Hendler and Ora Lassila, The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities, [J], Scientific American, <http://www.sciam.com/2001/0501issue/0501berners-lee>, 2001-05-01
- [3.2] <http://www.cs.wisc.edu/niagara>
- [3.3] Extensible Markup Language (XML) 1.0 (Second Edition), W3C, eds. Tim Bray, Jean Paoli, C.M. Sperberg-McQueen, Eve Maler., [EB/OL], <http://www.w3.org/TR/REC-xml>. 2000-10-06.
- [3.4] Namespaces in XML, W3C, eds. Tim Bray, Dave Hollander, Andrew Layman. <http://www.w3.org/TR/REC-xml-names/>, 1999-01-14.
- [3.5] David C. Fallside, XML Schema Part 0: Primer [EB/OL], W3C (PR), <http://www.w3.org/TR/xmlschema-0/>, 2001-03-30
- [3.6] Henry S. Thompson, David Beech, Murray Maloney, Noah Mendelsohn, XML Schema Part 1: Structures [EB/OL], W3C (PR),

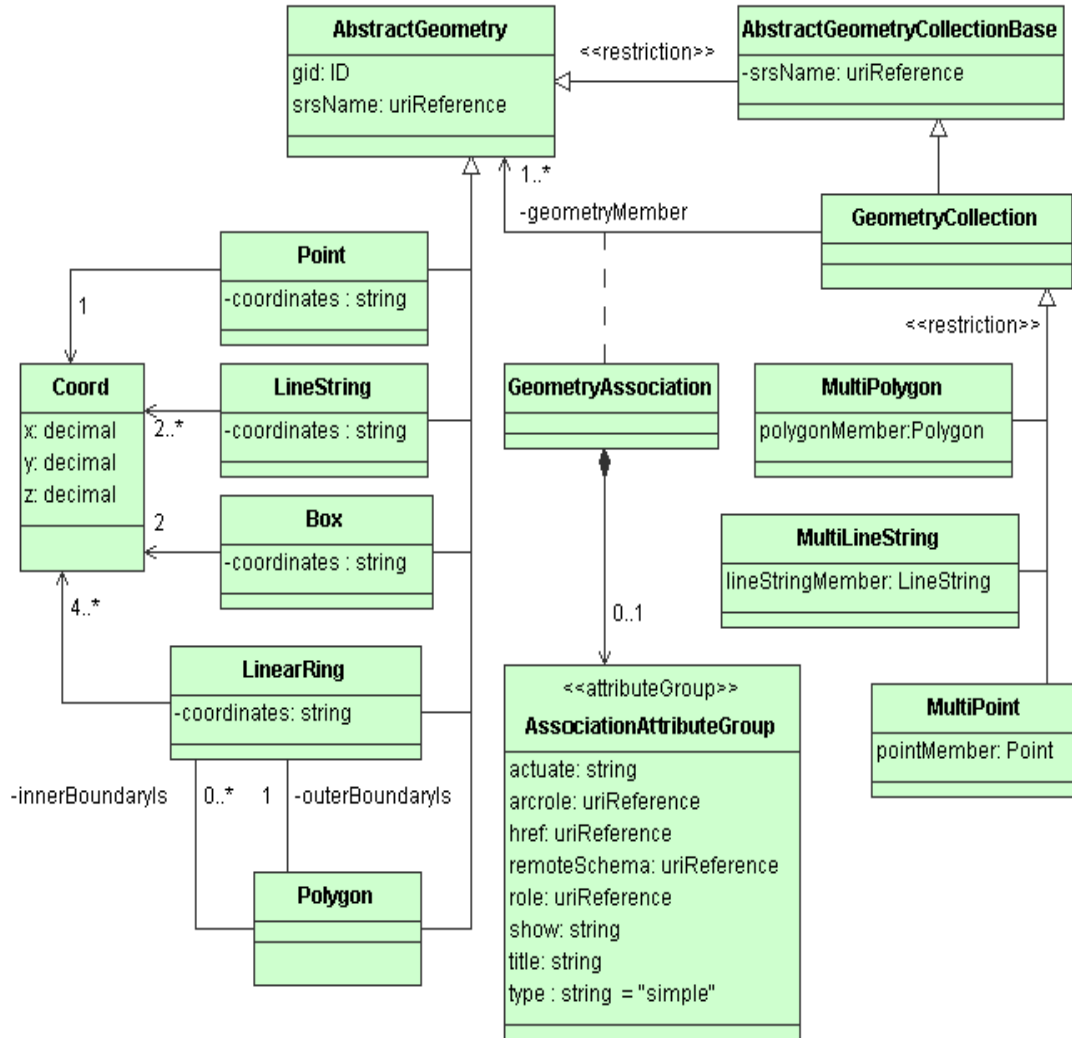
- <http://www.w3.org/TR/xmlschema-1/> , 2001-03-30
- [3.7] Paul V. Biron, Ashok Malhotra, XML Schema Part 2: Datatypes [EB/OL], W3C (PR), <http://www.w3.org/TR/xmlschema-2/> , 2001-03-30
- [3.8] Ora Lassila, Ralph R. Swick (editor), Resource Description Framework (RDF) Model and Syntax Specification [EB/OL], <http://www.w3.org/TR/REC-rdf-syntax>, 1999-02-22
- [3.9] Dan Brickley, R.V. Guha (editor), Resource Description Framework (RDF) Schema Specification [EB/OL], <http://www.w3.org/TR/rdf-schema>, 2000-03-27
- [3.10] XML Information Set, eds: John Cowan, Richard Tobin, <http://www.w3.org/TR/xml-infoset>, 2001-03-16
- [3.11] XML Linking Language (XLink) Version 1.0, eds, Steve DeRose, Eve Maler, David Orchard, <http://www.w3.org/TR/xlink>, 2000-12-20
- [3.12] XML1.0(2NE) Overview, <http://gis.pku.edu.cn/gis/TR/20010309-XML.htm> , (研究小组技术报告), 2001-03-09
- [3.13] 丁力, RDF 读书报告, <http://gis.pku.edu.cn/gis/TR/20010226-rdf.doc> , (研究小组技术报告), 2001-02-26
- [3.14] J.F.Sowa, Concept Graph, <http://www.bestweb.net/~sowa/cg/index.htm>, 2001-04-02
- [3.15] ISO/IEC 13250(ISO/IEC JTC 1) Topic Map [EB/OL], <http://www.topicmaps.net/>, 1999-12-03
- [3.16] UML 1.3 Specification [EB/OL], <ftp://ftp.omg.org/pub/docs/ad/99-06-08.pdf>, 1999-06-08
- [3.17] <http://www.ontoknowledge.org/oil/>
- [3.18] Sergey Melnik, Stefan Decker, A Layered Approach to Information Modeling and Interoperability on the Web, [EB/OL] <http://www-db.stanford.edu/~melnik/publications.html>, Sep 4, 2000
- [3.19] Gruber, T. R.. A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, 5(2), 199-220, [EB/OL] http://ksl-web.stanford.edu/KSL_Abstracts/KSL-92-71.html, 1992
- [3.20] Ron Lake, Adrian Cuthbert(editor), Geography Markup Language (GML) v1.0 [EB/OL], <http://www.opengis.net/gml/00-029.pdf>, 2000-05-12
- [3.21] Simon Cox, Adrian Cuthbert, Ron Lake, Richard Martell (editor), Geography Markup Language (GML) 2.0 [EB/OL], <http://www.opengis.net/gml/01-029/GML2.html>, 2001-02-20
- [4.1] Andrew S. Tanenbaum, Computer Networks(3rd Edition)[C], Prentice Hall, 1996, pp22~28
- [4.2] Open GIS Consortium. The OpenGISTM Abstract Specification Topic 12: OpenGISTM Service Architecture. <http://www.opengis.org/public/abstract/01-112.pdf>, 2001
- [4.3] 汪小林, “分布式 GIS 中的 QoS 问题及关键技术研究”, 北京大学博士论文, 2001
- [4.4] Don Chamberlin, etc., XQuery: A Query Language for XML, [EB/OL],

- <http://www.w3.org/TR/xquery>
- [4.5] T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, MIT/LCS, U.C. Irvine, Xerox Corporation, 1998-08.
 - [4.6] Tim Bray, etc. eds., Namespaces in XML. W3C Recommendation (14 January 1999). <http://www.w3.org/TR/1999/REC-xml-names-19990114/>
 - [4.7] Mockapetris, P., "Domain Names - Concepts and Facilities", STD 13, RFC 1034, 1987-11.
 - [4.8] Mockapetris, P., "Domain Names - Implementation and Specifications", STD 13, RFC 1035, 1987-11.
 - [4.9] <http://www.handle.net>
 - [4.10] OpenGIS Project Document **01-021r1**: Web Map Server
Editor: Jeff de La Beaujardiere (NASA) issued by the WWW Mapping SIG
<http://www.opengis.org/techno/discussions/01-021r1.pdf>, 2001
 - [4.11] OpenGIS Project Document **01-023**: Web Feature Server
Editor: Panagiotis A. Vretanos (CubeWerx Inc.) issued by the WWW Mapping SIG <http://www.opengis.org/techno/discussions/01-023.pdf>, 2001
 - [5.1] OpenGIS Project Document **01-028**: Styled Layer Descriptor.
Editor: Adrian Cuthbert,
<http://www.opengis.org/techno/discussions/01-028.pdf>, 2001
 - [5.2] <http://gis.pku.edu.cn/gis/samples/gml2/city.xml>, 2001
 - [5.3] <http://gis.pku.edu.cn/gis/samples/rdf/ServiceTemplate1.rdf>, 2001

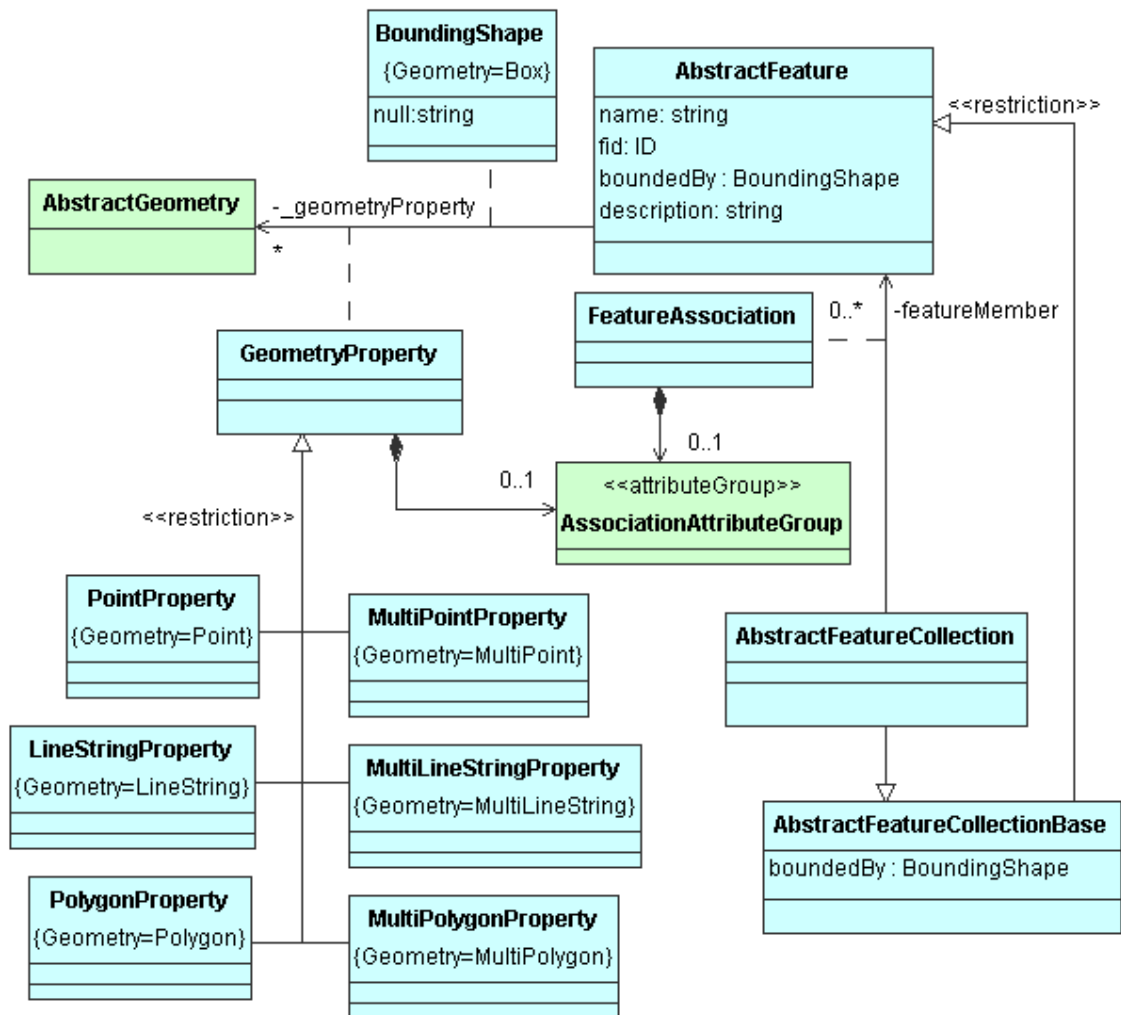
附录一：GML2.0 中对象类的类图

GML2.0 中采用 UML 描述了 Simple Feature Model, 并且用 XML Schema 定义了具体的 XML 化方法。下面是其中两个文件的对象类的类图。

附图 1 geometry.xsd 中对象类的类图



附图 2 feature.xsd 中对象类的类图



附录二：Map.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.5 NT (http://www.xmlspy.com) by () -->
<!-- File: map.xsd -->
<schema targetNamespace="http://gis.pku.edu.cn/map"
xmlns="http://www.w3.org/2000/10/XMLSchema" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:gml="http://www.opengis.net/gml" xmlns:map="http://gis.pku.edu.cn/map"
elementFormDefault="qualified" version="0.01">
  <annotation>
    <appinfo>map.xsd v0.01 2001-04</appinfo>
    <documentation xml:lang="en">extension of GML for mapping service, defined map and style
sheet</documentation>
  </annotation>
  <!-- import constructs from the GML Feature and Geometry schemas -->
  <import namespace="http://www.opengis.net/gml" schemaLocation="feature.xsd"/>
  <!-- =====
      element section
  ===== -->
  <element name="AbstractMap" type="map:AbstractMapType" abstract="true"/>
  <element name="MapLayerMemeber" type="map:MapLayerMemeberType"/>
  <element name="StyleSheetMember" type="map:StyleSheetMemberType"/>
  <element name="RenderLayer" type="map:RenderLayerType"/>
  <element name="AbstractStyleSheet" type="map:AbstractStyleSheetType" abstract="true"/>
  <element name="AbstractStyle" type="map:AbstractStyleType" abstract="true"/>
  <element name="Condition" type="string"/>
  <!-- common style descriptors -->
  <element name="Mark" type="map:MarkType"/>
  <element name="MarkColor" type="map:ColorRefType"/>
  <element name="MarkSize" type="map:SizeType"/>
  <element name="Stroke" type="map:StrokeType"/>
  <element name="StrokeColor" type="map:ColorRefType"/>
  <element name="StrokeWidth" type="integer"/>
  <element name="FillColor" type="map:ColorRefType"/>
  <element name="FontFamily" type="map:FontFamilyType"/>
  <element name="FontColor" type="map:ColorRefType"/>
  <element name="FontStyle" type="map:FontStyleType"/>
  <element name="FontSize" type="map:SizeType"/>
  <!-- =====
      map type section
  ===== -->

```

```

<complexType name="AbstractMapType" abstract="true">
  <sequence>
    <element ref="map:MapLayerMemeber" minOccurs="0" maxOccurs="unbounded"/>
    <element ref="map:StyleSheetMember" minOccurs="0" maxOccurs="unbounded"/>
    <element ref="map:RenderLayer" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
  <attributeGroup ref="xlink:extendedLink"/>
  <attribute ref="gml:remoteSchema" use="optional"/>
</complexType>
<complexType name="StyleSheetMemberType">
  <sequence>
    <element ref="map:AbstractStyleSheet" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
  <attributeGroup ref="xlink:locatorLink"/>
  <attribute ref="gml:remoteSchema" use="optional"/>
</complexType>
<complexType name="MapLayerMemeberType">
  <attributeGroup ref="xlink:locatorLink"/>
  <attribute ref="gml:remoteSchema" use="optional"/>
</complexType>
<complexType name="RenderLayerType">
  <attributeGroup ref="xlink:arcLink"/>
  <attribute ref="gml:remoteSchema" use="optional"/>
</complexType>
<!-- =====
      Style Sheet type section
===== -->
<complexType name="AbstractStyleSheetType" abstract="true">
  <sequence>
    <element ref="map:AbstractStyle" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>
<complexType name="AbstractStyleType" abstract="true">
  <sequence>
    <element ref="map:Condition"/>
  </sequence>
  <attribute name="sid" type="ID" use="optional"/>
</complexType>
<complexType name="PointStyleType">
  <complexContent>
    <extension base="map:AbstractStyleType">
      <sequence>
        <element ref="map:Mark" minOccurs="0"/>
        <element ref="map:MarkColor" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

```

        <element ref="map:MarkSize" minOccurs="0"/>
    </sequence>
</extension>
</complexContent>
<!-- single color, one internal -->
</complexType>
<complexType name="LineStyleType">
    <complexContent>
        <extension base="map:AbstractStyleType">
            <sequence>
                <element ref="map:Stroke" minOccurs="0"/>
                <element ref="map:StrokeColor" minOccurs="0"/>
                <element ref="map:StrokeWidth" minOccurs="0"/>
            </sequence>
        </extension>
    </complexContent>
    <!-- single color, one internal -->
</complexType>
<complexType name="PolygonStyleType">
    <complexContent>
        <extension base="map:AbstractStyleType">
            <sequence>
                <element ref="map:Stroke" minOccurs="0"/>
                <element ref="map:StrokeColor" minOccurs="0"/>
                <element ref="map:StrokeWidth" minOccurs="0"/>
                <element ref="map:FillColor" minOccurs="0"/>
            </sequence>
        </extension>
    </complexContent>
    <!-- single color, one internal -->
</complexType>
<complexType name="FontStyleType">
    <complexContent>
        <extension base="map:AbstractStyleType">
            <sequence>
                <element ref="map:FontFamily" minOccurs="0"/>
                <element ref="map:FontStyle" minOccurs="0"/>
                <element ref="map:FontColor" minOccurs="0"/>
                <element ref="map:FontSize" minOccurs="0"/>
            </sequence>
        </extension>
    </complexContent>
    <!-- single color, one internal -->
</complexType>

```

```

<simpleType name="ColorRefType">
  <restriction base="integer">
    <minInclusive value="0x000000"/>
    <maxInclusive value="0xFFFFFFFF"/>
  </restriction>
</simpleType>
<!-- =====
      basic type  section
===== -->

<complexType name="SizeType">
  <sequence>
    <element name="Width" type="integer"/>
    <element name="Heigh" type="integer"/>
  </sequence>
</complexType>
<simpleType name="MarkType">
  <restriction base="string">
    <enumeration value="Star"/>
    <enumeration value="Rect"/>
    <enumeration value="Ellipse"/>
  </restriction>
</simpleType>
<simpleType name="StrokeType">
  <restriction base="string">
    <enumeration value="Normal"/>
    <enumeration value="Dash"/>
    <enumeration value="DotDash"/>
  </restriction>
</simpleType>
<simpleType name="FontFamilyType">
  <restriction base="string">
    <enumeration value="Times New Roman"/>
    <enumeration value="Arial"/>
  </restriction>
</simpleType>
<simpleType name="FontSytleType">
  <restriction base="string">
    <enumeration value="Normal"/>
    <enumeration value="Italic"/>
    <enumeration value="Oblique"/>
  </restriction>
</simpleType>
</schema>

```


硕士期间发表的论文

1. 丁力、汪小林、许卓群, 数据库在地理信息系统中的应用研究, 计算机应用与软件, Vol.18, No.1, 2001.1。
2. Ding Li, Qu Lei, Zhang Ying, Luo Ying-wei, Wang Xiao-lin, Xu Zhuo-qun, Geo-Agents: Design and Implement, 武汉大学学报 (自然科学版), Vol.6, No.1-2, 2001.3 (International Software Engineering Symposium 2001)。
3. 罗英伟、汪小林、张颖、丁力、曲磊、丛升日、许卓群, 基于 Agent 的 WebGIS 信息搜索与分布计算模型, 工程图学学报, 2001, 增刊 (中国图象图形学会第十届学术会议)。
4. 丁力、汪小林、罗英伟、张颖、曲磊、许卓群, 地理信息系统与数据库结合研究, 中国图像图形学报, 2001。(已接收即将发表)
5. 罗英伟、汪小林、曲磊、张颖、丁力、许卓群, 构建分布式 GIS 的新基石——Agent 技术, 教育部 21 世纪高校 GIS 战略研讨会论文集, 武汉, 中国地质大学, 2001 年 5 月 22-24 日。
6. 曲磊、罗英伟、汪小林、熊国民、庄伟、丁力、张颖、许卓群, PAS: 一个多 Agent 系统的设计与实现, 同上。
7. 丁力、罗英伟、汪小林、张颖、曲磊、许卓群, Semantic Web 中空间信息共享框架的研究与设计, 同上。
8. 张颖、罗英伟、汪小林、丁力、曲磊、许卓群, 基于 XML 的地理信息元数据系统的设计与实现, 同上。
9. 罗英伟、马坚、汪小林、丁力、许卓群, 基于 GML 的大众化 WebGIS 应用系统研究, 同上。

致谢

首先要感谢关怀、教育我四年的导师许卓群教授和陆钟辉教授。二位导师不但在学习和科研方面给了我无私的教诲,而且在平时的生活中,他们的言传身教也让我受益匪浅。尤其是他们精深的学术造诣、勇攀科学高峰的精神、一丝不苟学术态度一直激励着我去探索世间的真理。敬祝两位老师身体健康、工作顺利、桃李满天下!

在论文设计与写作一百多个日日夜夜中,课题组的罗英伟博士后、汪小林博士,他们就论文中的不少问题与我进行了深入而富有成效的讨论,给了我许多宝贵的意见和启发。本文的顺利完成很大程度上离不开他们对我的关心与支持。

还要感谢张颖硕士、曲磊硕士和已毕业的师姐李慕华硕士和伍键硕士,多年来的精诚合作形成了这个默契而充满活力的研究开发团队,我们不仅是工作上最好的同志,也是生活中亲密的朋友。再次感谢各位同门对我各方面无微不至的关怀与支持。

同时还要感谢教研室的丛升日老师、张树云老师、谢桂芳老师,余华山博士这几年来对我的关怀和帮助。也要感谢新加入的马坚、刘昕鹏、喻丽珊同学对我的关心,他们给课题组带来了活力与新意。

最后还要感谢我的父母及全家人,是他们的谆谆教诲和无微不至的关怀使我有机会踏入北大这块圣地,并使我对未来充满信心。

感谢所有关心和爱护我的人们!