

基于 OGC WPS 标准的处理服务实现研究

孙 雨^{1,2} 李国庆¹ 黄震春³

(中国科学院对地观测与数字地球科学中心 北京 100086)¹

(中国科学院研究生院 北京 100190)² (清华大学计算机科学与技术系 北京 100084)³

摘 要 Web Services 为空间信息处理功能的互操作提供了一种通过网络访问的解决方案,但 Web Services 标准中缺少对空间信息元数据的定义。致力于 GIS 资源共享和处理互操作的 Open Geospatial Consortium(OGC)组织,针对该问题制定了 Web Processing Services(WPS)标准。基于 WPS 3 种主要方法提出了一种可扩展的 WPS 实现体系结构,用于解决空间信息互操作的问题。根据本体系结构,实现了一个镶嵌处理的 demo WPS 平台。试验证明,本体系结构具有很好的合理性、灵活性以及可扩展性,能够更好地解决处理功能互操作的问题。

关键词 Web Services, OGC, WPS, 互操作, 镶嵌

中图分类号 TP311 **文献标识码** A

Research on Implementation of OGC Web Processing Service

SUN Yu^{1,2} LI Guo-qing¹ HUANG Zhen-chun³

(Center for Earth Observation and Digital Earth, Chinese Academy of Sciences, Beijing 100086, China)¹

(Graduate University of the Chinese Academy of Sciences, Beijing 100190, China)²

(Department of Computer and Science Technology, Tsinghua University, Beijing 100084, China)³

Abstract Web services provide a network-based solution to achieving the interoperability of geospatial information processing, but there is a lack of the definition of geospatial information metadata. Against this problem, Web Processing Services (WPS) was proposed by Open Geospatial Consortium (OGC) who is dedicated to researching GIS resources sharing and processes interoperability. This paper proposed an extensible WPS implementation architecture based on the three main operations of WPS to solve the problem of geospatial information interoperability. Furthermore, we proved the design to be reasonable, feasible and extensible and can solve the interoperability problem better by developing a demo WPS platform with merge process according to this architecture.

Keywords Web services, OGC, WPS, Interoperability, Merge

随着信息技术的进步,社会对地理空间信息数据的依赖性不断增强,空间数据在科学研究以及日常生活中都扮演着越来越重要的角色。随着空间信息数据在共享、交换和用途方面的不断推进,处理功能的共享与互操作也更加备受关注。在空间信息共享以及处理互操作方面,Web Service 提供了一个开放的平台,通过网络访问服务资源,所有的处理功能都可以以 Web Service 的方式通过网络提供给用户^[1]。Web Service 是一种分布在网络上的、通过标准的 XML 信息格式提供诸如空间信息分析这些可以通过网络访问的处理操作。通常情况下,一个 Web Service 包含一个或一系列处理,它使用标准的、通用的 XML 标签进行描述,为服务使用者提供消息格式和足够的信息来与服务提供者进行交互^[2]。基于 XML 的 SOAP, WSDL, UDDI 等标准协议的采用,使得 Web Service 的使用更加广泛。

尽管 Web Service 具有不少优点,但是 Web Service 所有的标准都是通用的,因此不能很好地解决具体某一学科领域

的专业问题。在诸如地理空间信息领域,由于其在传输协议中没有包含空间信息元数据信息以及对空间信息数据的标准化,使得 Web Services 在解决 GIS 功能互操作方面存在不足。这极大地鼓舞了很多空间信息方面的专家来研发地理空间信息领域应用的服务标准。

为了解决 Web Service 在空间信息方面的不足,许多计算机和地理空间信息领域的公司与专家联合,成立了一个 OGC 组织^[3]。该组织主要制定标准规范来实现许多不同消息格式和不同通信机制的地理信息处理软件间的资源共享与处理功能的互操作,其工作主要致力于空间信息方面各种标准的制定^[4]。OWS 是一个着重研究如何在 Web Service 的基础上解决 GIS 数据共享和互操作的项目,其目的是基于发布的标准,提出一个可扩展的框架来无缝地集成各种在线的空间处理和位置服务。WPS 是 OWS 的一个重要组成部分,主要用来处理空间数据,通过网络向客户端提供任何类型的 GIS 处理功能。本文提出了一个实现 WPS 规范标准的体系

到稿日期:2008-09-04 返修日期:2008-12-11 本文受 863 国家重点项目(2006AA12Z203 和 2007AA12Z205)资助。

孙 雨(1983-),女,硕士,主要研究方向为空间信息网格、数据服务以及计算服务等,E-mail:yusun@ceode.ac.cn;李国庆(1968-),男,博士生导师,主要研究方向为空间信息网格技术以及高性能计算等;黄震春(1973-),男,讲师,主要研究方向为网格计算、空间信息系统和高性能计算。

结构,用来解决空间信息处理功能互操作的难题。

1 WPS 规范及主要方法

OWS 服务框架支持能够实现各种空间信息共享与互操作的服务、接口和交换协议。OWS 服务是一种遵循 OGC 标准规范实现的服务。OWS 包含的服务主要有应用服务、注册服务、数据服务、描述服务和处理服务。目前,OGC 组织最新制定了一个用来规定空间处理服务的标准化的接口 WPS,以标准的形式在网络上共享 GIS 功能处理服务^[5]。使用 WPS 可以把 GIS 所有的功能处理模块都发布成 Web Service,同时包含它们的输入、输出参数和触发方式。在 WPS 中,同时支持超文本传输协议中的 POST、GET 方法和 SOAP 传输协议。这样,客户端可以根据其意愿选择最合适的接口机制。尽管 WPS 最初主要面向地理空间信息相关的数据,但事实上,WPS 也支持其他类型的数据。

在 WPS 中,客户端和服务端采用基于 XML 的通信方式,在 WPS 接口中定义了 3 个主要操作,用于向客户端提供详细信息和查询部署在服务器上的服务。这些方法主要有: GetCapabilities 方法,用于客户端请求和接收描述具体服务实现方式的服务元数据文档,该方法支持在客户端与服务器交互中协商规范版本; DescribeProcess 方法,客户端可以请求和接收一个执行操作需要运行的进程的详细信息,包含输入、输出参数和格式; Execute 方法,客户端可以通过该方法使用提供的参数值来指定执行 WPS 实现的处理操作,并返回产生的结果。

2 基于 WPS 的体系结构及实现

作为一种专门用于解决空间信息共享与处理互操作的 Web 应用服务,OGC 的 WPS 规范可以有多种实现方式。三层或多层体系结构由于可以对显示层、业务逻辑层和数据层进行逻辑上和物理上的分离,便于系统的维护和升级,最大程度地减少了系统间的耦合,成为构建基于 Web 的应用系统的首选。并且,一个设计良好的体系结构应该易于扩展。具体来讲,应该使得算法或处理功能易于集成。基于上述考虑,本文根据 WPS 标准规范,探索性地研究了 WPS 实现,提出了关于实现算法可扩展的 WPS 服务的三层体系结构,如图 1 所示。

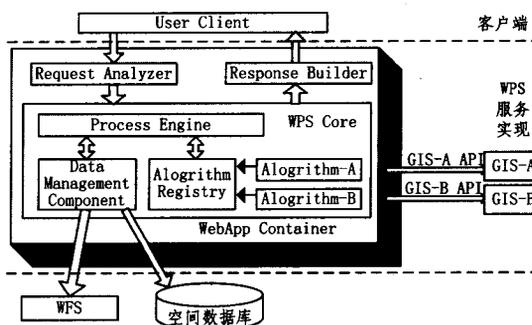


图 1 WPS 实现的体系结构

最上层为客户端,负责产生以及发送用户的请求。用户可以采用浏览器直接访问,也可以用标准的 Web 服务客户端调用服务层提供的服务。中间层为服务层,是整个体系结构的核心部分。服务层之下是数据源层,数据源层为上层提

供所需的数据,该层可以由一个 WFS (Web Feature Service) 服务实现,也可以由空间数据库直接提供数据。

服务端是基于现有的 Web 容器来实现的,底层的算法模块以插件的形式动态装载到服务器端。每个算法的实现对于服务代码都是透明的。通常情况下,算法是调用 GIS 的 API 来实现其功能的。

服务实现主要包括两部分内容:第一部分是服务核心部分,第二部分是请求和响应 SOAP 消息的解析和构造。其中第二部分的实现可以借助于已有的 XML 文档解析器,比如 Java 下的 dom4j 和 JDOM 等。使用现有的 XML 文档解析器,将 OGC 发布的 WPS 请求和响应的 schema 文档转化为对应的 Class 定义,从而实现 XML 文档的解析和构造。前者则包括处理引擎、算法注册中心、数据管理以及具体的算法实现。以下重点讲述服务核心部分各模块的功能以及具体实现。

2.1 处理引擎

处理引擎是 WPS 服务的总控部分,由它驱动各模块,构造响应消息,完成服务。具体来讲,对于 GetCapabilities 请求,由处理引擎查询算法注册中心,获得注册的所有算法的元数据信息,包括算法的 Identifier、算法名称以及算法的说明等。对于 DescribeProcess 请求,处理引擎根据请求给出的处理 Identifier,查询算法注册中心。算法注册中心返回该处理的详细信息,包括算法的输入输出名称以及对每个输入输出的说明。对于 Execute 请求,处理引擎根据用户的请求,决定是否需要调用数据管理模块来获取数据。然后将请求中的算法 Identifier 在算法注册中心中查询,获得算法具体的实现类后,用 Java 的反射机制调用该类,完成处理。最后根据请求中的要求,在 SOAP 中直接返回结果数据,或者调用数据管理组件,将处理结果以 WFS 的形式发布。

2.2 算法注册中心

算法注册中心的主要功能是维护算法类的元数据信息,具体来讲,包括每个算法的标识符、实现类的 Qualified Name、算法所需的输入输出等。在服务初始化时,利用 Java 的反射机制动态地发现已有的算法,将其注册到算法注册表中。另外,算法注册中心还将类定义中的 Annotation 信息提取出来,储存在注册中心中,以供处理引擎的查询和调用。

2.3 数据管理组件

数据管理组件的主要功能是调用 WFS、获取输入文件以及将处理引擎生成的结果文件以 WFS 的形式返回给用户。

需要进行处理的矢量文件可以由客户端上传到服务器端,这其中有两种方式:第一种是直接基于 HTTP 发送矢量的二进制文件,比如 SHP 文件。另外一种办法是将输入文件在客户端转化为 GML 的格式,发送给服务端;而服务端处理后,仍然以 GML 格式返回给客户端^[6]。GML (Geography Markup Language)即地理标识语言由 OGC 于 1999 年提出,并得到了许多公司的大力支持。目前的最新版本是 OGC 在 2007 年 8 月提出的 3.2.1 版。从本质上讲,GML 是一种经过特殊化的 XML 语言。

除了由客户端上传以外,WPS 标准也支持通过调用 WFS 服务获取一个数据文件。在数据管理组件中包含 WFS 的客户端程序。如果在 ExecuteProcess 请求中要求通过 WFS 协议获取输入数据,数据管理组件则调用这部分代码来

获得相关的输入文件。

2.4 算法实现部分

每个算法的实现被封装到一个单独的类中。该类必须实现预先规定的特定接口,以便它能够注册到算法注册中心中。具体的算法实现不是本文重点要讨论的问题。通常是通过调用 GIS 提供的 API 来实现特定的处理功能。

3 实验结果与分析

本文以空间分析中的典型算法(矢量镶嵌)为例,根据上面的 WPS 体系框架,开发、部署、发布该处理功能的服务。镶嵌是 GIS 的一个典型处理操作,是指将两幅或者多幅图像拼接成一幅图像的过程^[7]。在栅格图像的矢量化操作中,由于仪器设备的限制,矢量化通常是分幅进行的,而分幅矢量化得到的矢量图形的拼接则是由矢量镶嵌完成的。

服务器端代码采用 Java 语言实现。Tomcat 5.5 作为 Web 容器,底层的镶嵌算法基于 ArcGIS 9 的 ArcGIS Engine 提供的 API 实现。客户端采用 52North^[8] 改造的 JUMP (JUMP Unified Mapping Platform)^[9]。JUMP 给用户提供了可扩展的图形用户界面以及 API,以供用户开发和调用自定义的处理函数。52North 对 JUMP 进行了扩展,使其能够调用远程的 WPS 服务,并显示服务返回的结果文件。

WPS 服务调用的 UML 时序图如图 2 所示。用户首先将 WPS 服务的 URL 添加到 JUMP 客户端中,客户端将调用 GetCapabilities 方法。用户根据客户端返回的操作种类和类型选择一种操作后,界面将自动发出 DescribeProcess 请求,并根据响应生成处理参数的输入界面。用户选择输入文件后,调用 Execute 方法,由 WPS server 调用适当的 WFS 服务以及算法模块,并将运行结果返回给 JUMP 客户端。

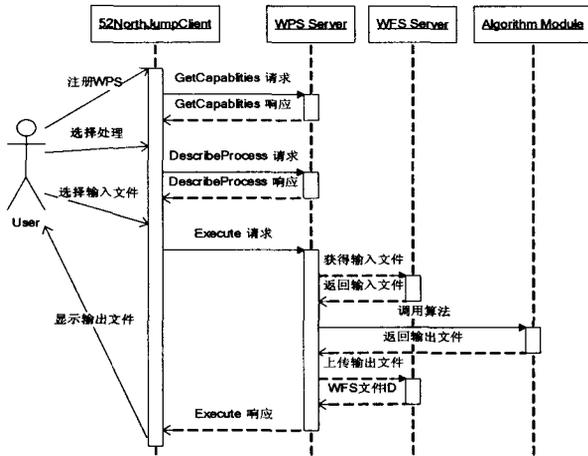


图 2 WPS 的 UML 时序图

实验输入矢量数据如图 3(a)和图 3(b)所示。客户端提交执行请求的用户界面如图 4 所示,该界面由改造后的 JUMP 读取 WPS DescribeProcess 响应后,动态构建而成。WPS 返回的镶嵌处理结果如图 5 所示。

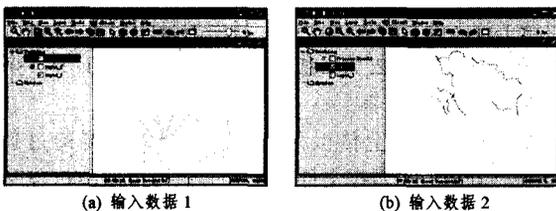


图 3

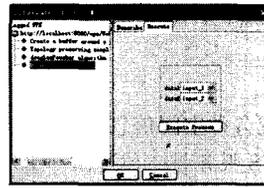


图 4 参数输入的图形界面

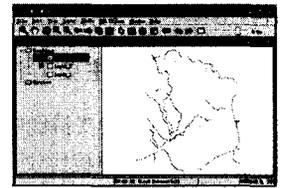


图 5 处理结果

使用 WPS 体系结构发布的 merge 处理结果与本地 GIS 软件上处理得到的结果是完全一致的。但与后者相比,利用 WPS 服务体系发布 GIS 处理功能,用户可以方便、快捷、广泛地访问地理信息,既可以降低系统运行成本,又达到计算机负载均衡高效的优点。另外,该 WPS 体系框架具有可扩展性,可以提供的处理功能不局限于某个特定的 GIS 软件而且可以在一个 WPS 服务中提供多个 GIS 处理功能以及根据需求快速部署实现开发处理算法服务。最重要的是,从系统的实现架构上讲,它是开放的、基于公共标准的平台,开发人员很容易将自己开发的算法集成到该平台中,从而可以很方便地与其它遵循 OGC WPS 以及 WFS 等标准的平台进行互操作,实现数据共享与处理共享。

结束语 功能强大的 Web Services 在处理地理空间信息领域问题方面存在严重的不足,而 OGC WPS 标准规范的发布使该问题迎刃而解。本文在充分分析、评价 WPS 标准的基础上,提出了一种可扩展的 WPS 服务实现架构。基于这种实现框架,用户可以将研究开发的算法添加到 WPS 服务中,作为其中的一个处理。通过实现一个提供矢量镶嵌的 WPS 服务,证明本实现框架具有可行性、可扩展性,能够解决 GIS 领域处理互操作的问题。

在 WPS 这种基于 Web 服务的模式下,软件证书如何管理和控制是该服务下一步需要考虑和解决的问题。传统 GIS 软件通常是根据组件个数、使用期限以及并发用户数等因素进行证书的发放。将 GIS 的处理功能发布成为 Web 处理服务带来的问题是如何对用户进行身份认证和授权控制。认证方案可以采用 X.509 证书进行认证。可是,如何实现细粒度的授权,即可以控制用户是否能够使用某一具体处理功能,目前似乎还没有比较完善的解决方案。

另外,随着 OGC 的 WPS 标准被越来越多的厂商和用户所接受,将伴随越来越多的 WPS 服务实现的出现,其中可能包括针对不同类型数据、不同应用的处理操作。因此,如何去管理这些 WPS 服务实现是目前亟待解决的问题之一。同时,该问题涉及到如何根据用户对服务质量以及服务内容的需求,让用户找到一个适合于自己的 WPS 服务。而对于一般化的 Web 服务来说,UDDI (Universal Description, Discovery, and Integration) 是其中的一个解决方案,但是将其应用到具体的空间信息领域,还需要对其进行研究以及扩展。

参考文献

- [1] Gottschalk K, Graham S, Kreger H, et al. Introduction to Web Services Architecture[J]. IBM Sys. Journal, 2002, 41(2): 170-177
- [2] Foerster T, Stoter J. Establishing an OGC Web Processing Service for generalization processes[A]// Workshop of the ICA Commission on Map Generalisation and Multiple Representation [C]. June 2006

(下转第 137 页)

+0x174 ImageFileName : [16] UChar

.....

+0x1b0 Peb : Ptr32_PEB

.....

其中 ExitTime 可以作为判断 EPROCESS 是否合法的依据: ExitTime 是进程的退出时间,对于所有退出运行的进程来说,这一项都是 0。

4 实现及实验分析

4.1 工具实现

目前,我们已经根据本文提出的基于交叉视图的进程检测方法实现了一个基于隐藏进程检测的 Windows rootkit 检测工具 VITAL。工具采用了基于 DLL 插件机制的结构,分为 GUI 层、DLL 层和内核层,插件结构使得检测工具便于升级,层次结构使得工具有一部分运行在 Windows 内核中,提高了检测的能力。

该检测工具在 Windows XP SP2 下开发完成,并在 Windows Vista ultimate 操作系统上完成部分测试。GUI 层和 DLL 层使用 VC 6.0 作为开发工具,使用 Windows DDK 2600 进行内核层的开发,内核层由若干驱动程序组成。在开发中,使用了 Windbg 和 IDA 等工具进行辅助。

4.2 实验及分析

选用 Windows XP Professional SP2 作为实验平台,选用 Windows 操作系统上比较典型和常见的几个 rootkit 进行测试,包括 Hacker Defender, NtRootkit, HideProcessHook 以及 FU Rootkit,同时,选取 Windows 工具 Processes Explorer 和 rootkit 检测工具 Rootkit Unhooker^[13]作为对照。各个 rootkit 的原理及实验结果如表 1 所列。

表 1 VITAL 和部分工具的检测结果对比

	Process Explorer	Rootkit Unhooker	VITAL
Hacker Defender	×	√	×
NtRootkit	×	√	√
HideProcessHook	×	√	√
FU Rootkit	×	×	√

Process Explorer 是微软公司提供的查看 Windows 上正在运行的进程列表的工具,其原理和 Windows 自带的任务管理器相似,都是通过调用系统 API 来查看进程列表,没有 rootkit 检测的功能,所以 rootkit 一般都能对其达到隐藏的效果。Rootkit Unhooker 采用钩子检测的方式进行 rootkit 检测,这种工具能检测大部分用户态 rootkit 以及使用 SSDT 钩子进行进程隐藏的 rootkit,但是不能检测到类似 FU Rootkit 这种使用直接内核对象操纵的 rootkit。Rootkit Unhooker 是一种从 rootkit 使用的技术入手进行 rootkit 检测的方法,由

于 rootkit 使用的技术较多,变化较快,使得这种方法具有一定的局限;而不管 rootkit 使用何种技术,它表现的行为都是隐藏,本文的方法从 rootkit 的行为入手,取得了比较好的效果。

结束语 本文以对 Windows rootkit 的研究为基础,提出了通过隐藏进程检测进行 rootkit 检测的机制,并设计了一种基于交叉视图的隐藏进程检测方法,最终实现了一个 Windows rootkit 检测工具。通过实验,验证了该工具具有较强的检测能力,在某些方面优于目前出现的检测工具。

随着 Internet 的进一步发展,新的 rootkit 将会不断涌现,根据目前对于 rootkit 发展趋势的研究,我们认为进一步的工作将会包含这些内容:提出功能更强的检测方法,增加钩子、文件、注册表项等的检测以完善检测工具的能力;对 Windows 新推出的操作系统如 Windows Vista 提供 rootkit 检测;对 rootkit 监测技术进行研究,变事后发现为事先监测;引入虚拟机等技术对检测工具进行保护,防止 rootkit 对其的破坏。

参考文献

- [1] CERT® Advisory CA-1994-01 Ongoing Network Monitoring Attacks[OL]. <http://www.cert.org/advisories/CA-1994-01.html>
- [2] Rutkowska J. Subverting Vista™ Kernel For Fun And Profit [J]. Blackhat Presentation, August 2006
- [3] Uty. 搜索内存枚举进程[OL]. <http://blog.donews.com/uuty/archive/2006/03/15/769472.aspx>
- [4] Wiki[OL]. <http://en.wikipedia.org/wiki/Rootkit>
- [5] King S T, Chen P M. SubVirt: Implementing malware with virtual machines[C]// Security and Privacy, IEEE Symposium, 2006
- [6] Heasman J. Implementing and Detecting an ACPI Rootkit[M]. BlackHat Federal, 2006
- [7] Heasman J. Implementing and Detecting a PCI Rootkit[M]. November 2006
- [8] Hoglund G, Butler J. Rootkits: Subverting the windows kernel. 2007
- [9] FUrootkit[OL]. http://www.rootkit.com/vault/fuzen_op/FU_Rootkit.zip
- [10] Russinovich M, Solomon D. Microsoft Windows Internals[M]. MicroSoft Press, 2005; 289-295
- [11] Baker A. Windows NT 设备驱动程序设计指南[M]. 北京:机械工业出版社, 1998; 44-45
- [12] Jimhotkin. Detection of the hidden processes[OL]. <https://www.rootkit.com/newsread.php?newsid=434>
- [13] Rootkit Unhooker[OL]. <http://www.antirootkit.com/software/RootKit-Unhooker.htm>

(上接第 88 页)

- [3] Open Geospatial Consortium, Inc. Vision and Mission. WWW document[OL]. <http://www.opengeospatial.org/about/?page=vision,2006>
- [4] Michaelis D C, Ames P D. Evaluation and Implementation of the OGC Web Processing Service for Use in Client-Side GIS. Geoinformatica DOI 10. 1007/s10707-008-0048-1
- [5] Open Geospatial Consortium, Inc. OpenGIS® Web Processing Service (WPS) Specification. WWW document[OL]. http://portal.opengeospatial.org/files/?artifact_id=24151,2007
- [6] Kim Hak-Hoon, LEE Kiwon. Web-based GIS- transportation

framework data services using GML, SVG and X3D. Dept. of Information System Engineering, Hansung University Seoul, Korea; 136-792

- [7] 何强, 马颂德. 图像镶嵌技术理论、难点及应用[J]. 高技术通讯, 1998(3)
- [8] WebProcessingService (WPS). WWW document[OL]. http://52north.org/index.php?option=com_projects&task=showProject&id=21&Itemid=127
- [9] <http://www.jump-project.org/>
- [10] Standards for XML and Web Services Security [J]. Martin Naedele, Computer, 2003, 36(4): 96-98