

基于 Web Services 的城市规划信息系统集成框架研究*

王海^{1,2}, 王志强¹, 李琛^{1,3}, 盛科荣¹

(1. 中国科学院地理科学与资源研究所, 北京 100101; 2. 中国科学院研究生院, 北京 100049; 3. 北京联合大学 应用文理学院, 北京 100083)

摘要: Web Services 是一种面向服务的体系结构, 其突出优点是实现了真正意义上的平台无关性和语言独立性。从 Web Services 的核心思想出发, 分析了 Web Services 的关键技术和体系结构, 针对目前城市规划信息系统发展中存在的信息共享问题, 提出了一个基于 web Services 的城市规划信息系统集成框架。并给出了一个利用 ASP.NET 在 Visual Studio .NET 环境下实现 UPIS 系统 Web Services 开发的实例。

关键词: 城市规划信息系统; Web Services; 集成框架

中图分类号: TU984 **文献标识码:** A **文章编号:** 1006-7329(2006)04-0024-05

Study on the Integration Framework of Urban Planning Information System Based on Web Services

WANG Hai^{1,2}, WANG Zhi-qiang¹, LI Chen^{1,3}, SHENG Ke-rong¹

(1. Institute of Geographical Sciences and Natural Resources Research, Chinese Academy of Sciences, Beijing 100101, P. R. China; 2. Graduate School of the Chinese Academy of Science, Beijing 100049, P. R. China; 3. College of Arts and Science of Beijing Union University, Beijing 100083, P. R. China)

Abstract: Web Services is a service-oriented structure and which is really platform-independent and language-independent. In this paper, its system and key technology are analyzed, then aiming at the main problems in information sharing of urban planning information system (UPIS), a UPIS integration framework based on the Web Services is proposed and its principle is studied. Furthermore, an example with realization of UPIS system by ASP.NET in Visual Studio .NET is given.

Keywords: urban planning information system; Web Services; integration framework

作为人类生产和生活的重要空间, 城市的可持续发展 and 建设一直是倍受关注的焦点。城市规划作为调控城市发展的一个有效手段, 不断受到城市快速发展现实的挑战。特别是“数字城市”的建设理念提出以后, 如何利用快速发展的现代信息技术和网络平台, 建设好城市规划信息系统 (Urban Planning Information System, UPIS), 为增强城市的规划和管理提供辅助决策支撑成为亟待解决的一个现实问题。

1 城市规划信息系统 (UPIS) 建设及其存在的问题

1.1 城市规划信息系统 (UPIS) 建设现状

城市规划信息系统 (以下简称 UPIS) 是由用户 (规划设计管理人员、相关企事业单位、公众等)、计算机

等组成的进行城市规划相关信息收集、传送、储存、加工、发布、维护和使用的系统, 它不仅仅是一个技术系统, 而且是一个管理信息系统 (MIS) 和社会系统^[1]。UPIS 是以城市规划数据库为核心, 将计算机技术、通信技术、网络技术、3S (GIS、GPS、RS) 技术和城市规划的理论方法系统综合应用于城市规划事务的一个集成系统。主要满足对空间数据和属性数据以及部分文档的采集、存储、使用 (浏览查询、汇交发布、访问下载、在线分析)、动态更新功能, 进而在此基础上辅助于城市的科学决策。

UPIS 作为现代城市规划体系中的重要组成部分, 已经成为规划编制设计、规划政策法规、规划实施管理等环节中的重要技术支撑手段和连接城市规划诸多环

* 收入日期: 2005-12-28

基金项目: 国家自然科学基金项目 (40341008)

作者简介: 王海 (1978-), 男, 甘肃人, 博士生, 主要从事城市与区域规划研究。

节的纽带,同时也为城市的开发建设、社会力量参与城市经营、公众参与城市规划决策提供了重要的信息平台(图1)。

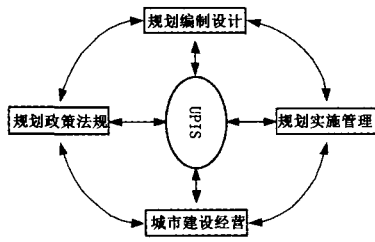


图1 UPIS与城市规划建设的关系

在现代技术,如面向对象技术、COM/DCOM技术、Internet/Intranet技术、Web GIS技术等,与城市发展中大量规划管理任务的现实需求双重推动下,UPIS不断深化发展。由原来简单的规划文档与图形/空间信息分隔处理的内部办公自动化系统逐步发展成面向对象的,以大型的分布式数据库建设为基础,并在此基础上构架了各类应用系统,乃至专家系统的图文一体化的综合信息系统。

1.2 城市规划信息的共享是发展中存在的最大问题

城市化进程迅速推进中带来的诸多矛盾和问题使得人们更加注重科学合理的城市规划,很多城市都从可持续发展的前景出发,构建了各自的UPIS,在推动城市规划科学决策中发挥了重要作用。但一个值得重视的突出问题是,许多城市的规划信息系统,发展建设处于不同层次和不同阶段,多局限在本部门使用,对外接口很少,规划信息共享程度不高。并且很多是在不同的平台上开发建设而成,开发语言不同,通讯协议也不同,大量不同来源的数据分布存贮,对外交换的数据格式存在很大差异,给C/S(Client/Server)结构中数据的共享和互操作带来很大的障碍,造成了规划信息孤岛。即使是以GIS技术为核心构建的UPIS,也因不同的GIS软件标准差异、数据格式差异而使得信息共享的难度很大^[2]。

但从城市规划发展的现实需求来看,规划信息的共享是发展的客观要求。城市规划信息因量大面广,涉及多部门多方面的利益,不仅对城市规划部门本身具有重要意义,而且与园林、消防、电力、水利、市政、环保等部门也密切相关,对于参与城市建设的企业和加强城市规划的公众参与也具有积极的作用。动态规划和弹性规划理念的提出和实践探索^[3],更强化了对信息共享的迫切需求。可以说,规划信息共享和互操作程度的提高与否,将直接影响到UPIS作用的有效发挥和深入发展。

因此,如何去解决语言差异、平台差异、协议差异、数据格式差异条件下的系统集成是UPIS实现信息共

享中的关键。而XML(eXtensible Markup Language)和Web Services作为Internet新技术的典型代表,为更广泛意义上的信息共享提供了强有力的支撑。

2 Web Services技术标准与体系结构

Web Services基于XML文档进行服务描述,服务请求和反馈结果,可以在Internet上通过HTTP(Hypertext Transfer Protocol)协议进行传递。同时,由于Web Services的相关标准都是W3C(World Wide Web Consortium)的开放协议^[4],与平台和操作系统无关,不同平台和操作系统上的Web Services的实现在很大程度上可以做到互操作,这使得异构平台上应用的集成变得易于实现,不同平台的客户端可以无缝的获取应用服务。

2.1 Web Services主要相关协议与技术

Web Services以技术栈的形式规范了体系中的各类关键技术^[5],包括服务的描述、发布、发现以及消息的传输等。主要有XML和HTTP、SOAP(Simple Object Access Protocol)、WSDL(Web Services Description Language)、UDDI(Universal Description, Discovery, and Integration)等。

XML是目前全球范围内用于描述数据和交换数据的方式,它是一种数据共享机制,具有自描述性、可扩展性、可校验性、层次结构、丰富的链接定义、多样的样式表支持。

SOAP是消息传递的协议^[6],用于在Web上传输XML格式的数据,并规定了PRC(Remote Procedure Call)的一系列规则。用户应用程序可以通过SOAP协议来访问Internet上的Web服务。SOAP协议可以构建在TCP(Transmission Control Protocol)、SMTP(Simple Mail Transmission Protocol)和HTTP等协议之上。因HTTP协议易于穿越防火墙,便于进行Internet上的分布式计算而在实际中采用的较多。

WSDL是一种描述Web服务的信息内容的XML语言,是用户与服务器之间的一个协约。它定义了描述Web服务接口规范的标准格式。有了WSDL,服务请求者就可以真正以一种语言无关和平台无关的方式自动产生Web服务的代理。

UDDI^[6]是一套基于Web的、分布式的、为Web服务提供信息注册中心的实现标准规范。同时也包含一组使用户能将自身提供的Web服务注册以使别的用户能够发现访问协议的实现标准。即UDDI定义了一个注册和定位Web服务的方法,便于用户描述和注册Web服务,并发现其他用户的Web服务并与它集成。

2.2 Web Services体系结构

在 Web Services 体系中,所有的应用实体都被抽象成服务。主要有三个实体:服务提供者(Service Provider)、服务请求者(Service Requester)、服务注册表(Service Registry)和三种操作:发布(Publish)、查找(Find)、绑定(Binding)。典型的 Web 服务体系结构表现为通过三种操作进行三面向实体之间的交互^[7,8](图 2)。

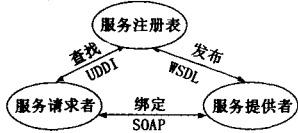


图 2 Web Services 体系结构

典型的 Web Services 实现模式为:服务提供者按照 UDDI 规范,使用 WSDL 定义对 Web Services 进行描述,并将其发布到服务注册上,保持注册信息的动态更新。服务请求者使用查找的操作,从服务注册表上检索到服务描述,获取与绑定相关的信息,与服务提供者进行绑定,并激活 Web Services 中的实现方法,完成调用。在这个过程中,服务请求者并不需要知道服务提供者提供服务的具体实现技术和分布式的拓扑结构。Web Services 架构很好的体现了跨平台可集成性和异构环境的互操作性的优点。

3 基于 Web Services 的 UPIS 集成框架

基于 Web Services 的 UPIS 集成框架如图 3 所示,其主要目的是为了实规划信息的共享以及现有系统的有机集成。这种共享不仅体现在系统原有各个模块之间的信息互操作,而且更多的体现在对与城市规划相关的政府部门、企事业单位、个人等外部用户的信息共享服务上。虽然各个 UPIS 发展程度不一,但多是按照层次结构架构的应用系统。因此基于 Web Services 的 UPIS 集成框架,主要采取的方法是将原有的 UPIS 的不同模块包装成 Web 服务,布署到 Web 服务器上,并在 UDDI 注册中心注册。这种集成方式,只需要将中间层组件简单封装一下即可,整个系统不需要做太大的改动。同时并不改变原有系统各个模块之间的逻辑结构,也不改变原有 UPIS 系统与后台数据库的连接关系,却可以实现多个规划应用系统之间数据、过程和服务的有机共享。当然,如果系统开发初期就按照 Web 服务的思想架构,更有利于系统间数据的共享和集成。以这种框架开发的系统可以方便的被 PC、PDA 等终端设备通过 Internet 进行访问,同时也很容易与其它系统进行互操作。

UPIS 的 UDDI 注册中心可采用 Windows Server 2003 提供的 UDDI 服务在内联网或外联网运行私有的

UDDI 目录服务。作为 Windows Server 2003 中 Web 服务构架的核心部分,UDDI 服务使发现、共享和重用 Web 服务与编程资源更容易。UDDI 服务支持通过 UDDI 应用开发接口(API)编程查询,也包含了一个具有查找、发布与协作功能的 Web 界面,能够与 Microsoft Internet Explorer 4.0 以上版本及 Netscape Navigator 4.5 或更新的版本兼容。UDDI 服务同时支持 1.0 与 2.0 版本的 UDDI 开发 API,使开发人员能够直接通过他们的开发工具与商务应用程序公布、发现、共享并相互交互 Web 服务。

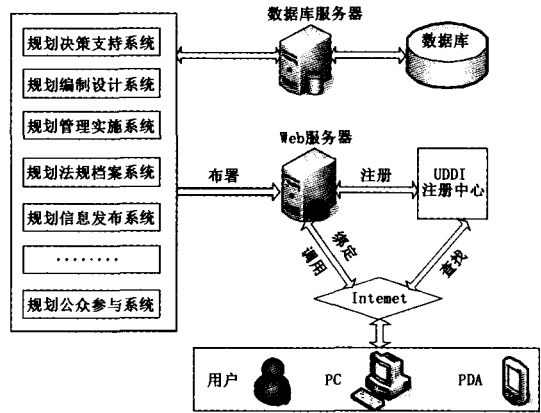


图 3 基于 Web Services 的 UPIS 集成框架

根据系统需要还可在上述框架中添加 SOAP 路由器和 Web Services 适配器等中间部件。作为集成引擎的 Web Services 适配器,是一个连接到后端服务器的应用程序,主要有接口、连接控制逻辑、数据转换器、消息路由器几个部分组成,针对不同的应用系统,适配器提供不同的接口^[9]。这种模式下的 UPIS 中,用户调用 Web Services 的过程如下:SOAP 请求作为一条 HTTP Post 发出后,交由 Web 服务器处理。Web 服务器分析 HTTP 头信息并找到 SOAP 路由器的名称,将请求的消息传递到制定的 SOAP 路由器。SOAP 路由器分析 HTTP 头找出某个 Web Services 适配器的位置,将该请求传递到所请求的适配器。适配器激活应用,具体的 UPIS 应用系统处理请求,并将结果返回到适配器。适配器将得到的结果打包成 SOAP 消息,返回 SOAP 路由器,再返回到 Web 服务器,用户最终得到包含了处理结果的 SOAP 消息。

这种架构可以较好的保证在原有应用系统上的性能扩展,在功能上也具有开放性和可延展性,能够在很大程度上满足规划信息的共享。基于 Web Services 的 UPIS 集成框架的优点主要体现在以下方面:

1) 开放性。Web Services 技术是基于开放标准的,这使得用 Web Services 技术集成的 UPIS 能够屏蔽不同应用平台的差异而被广泛接受。

2)简单性。对于实际的开发者,在这个框架下,能较好的利用 Web Services 本身便于开发的特性,易于创建满足规划实际工作的多种不同服务。

3)灵活性。传统的应用集成是面向具体的应用逻辑,是一种紧密耦合的集成方式,对于需要集成的两个城市规划应用程序来说,任何一端的改变都必须相应的改变另一端,这必然会增加集成的开发成本。而基于 Web Services 的应用集成,因待集成的两个规划应用程序是建立在一种松散耦合的关系之上的,具有灵活性且能节约开发成本。

4)高效性。Web Services 允许把应用程序划分成更小粒度的逻辑组件,在小粒度基础上的应用集成对于 Web Services 来说更为容易,这就使基于 Web Services 的 UPIS 的应用集成比传统应用集成更为高效,更能满足未来多样个性化的需求。

4 应用实例及其探讨

Web 服务是基于 SOAP 的,SOAP 独立于服务的具体实现方式,开发 Web Services 的方式也多种多样,有很多开发工具包都可以实现。例如:IBM 有 Web Services Toolkit、WebSphere Software Developer Kit for Web Services;Apache 提供 Web Services Project;Microsoft 公司提供了 SOAP 工具包、.NET Remoting 以及 ASP.NET 多种开发工具。

Visual Studio .NET 是惟一一个从一开始就是为 XML Web Services 创建的开发环境。Visual Studio .NET 和 XML Web Services 提供简单、灵活和基于标准的模型,允许开发人员从新的和现有的代码汇编应用程序,而与平台、编程语言或对象模型无关。

4.1 规划信息发布系统中 Web 服务的 WSDL 描述

利用 ASP.NET 在 Visual Studio .NET 环境下实现 UPIS 系统 Web Services 的开发,数据库选用 SQL Server 2000,Web 服务对数据库的访问采用 ADO.NET 技术,.NET 平台对 XML 文档的解析采用 MSXML 4.0。下面是规划信息发布系统中一个 Web 服务的 WSDL 描述。

```
<? xml version = "1.0" encoding = "utf-8" ? >
< wsdl: definitions xmlns: http = " http://schemas. xmlsoap. org/
wsdl/http/"
xmlns: soap = " http://schemas. xmlsoap. org/wsdl/soap/" xmlns: s
= " http://www. w3. org/2001/XMLSchema"
xmlns: soapenc = " http://schemas. xmlsoap. org/soap/encoding/"
xmlns: tns = " http://www. UPIS. com/"
xmlns: tm = " http://microsoft. com/wsdl/mime/textMatching/"
xmlns: mime = " http://schemas. xmlsoap. org/wsdl/mime/" tar-
getNamespace = " http://www. UPIS. com/"
```

```
xmlns: wsdl = " http://schemas. xmlsoap. org/wsdl/" >
< wsdl: types >
< s: schema elementFormDefault = " qualified" targetNam-
espace = " http://www. UPIS. com/" >
< s: element name = " AddUPMsg" >
< s: complexType >
< s: sequence >
< s: element minOccurs = " 0" maxOccurs = " 1"
name = " msg_ Title" type = " s: string" / >
< s: element minOccurs = " 0" maxOccurs = " 1"
name = " msg_ Date" type = " s: string" / >
< s: element minOccurs = " 0" maxOccurs = " 1"
name = " KeyWord" type = " s: string" / >
< s: element minOccurs = " 0" maxOccurs = " 1"
name = " msg_ Body" type = " s: string" / >
</s: sequence >
</s: complexType >
</s: element >
< s: element name = " AddUPMsgResponse" >
< s: complexType / >
</s: element >
</s: schema >
</wsdl: types >
< wsdl: message name = " AddUPMsgSoapIn" >
< wsdl: part name = " parameters" element = " tns:
AddUPMsg" / >
</wsdl: message >
< wsdl: message name = " AddUPMsgSoapOut" >
< wsdl: part name = " parameters" element = " tns: AddUPMs-
gResponse" / >
</wsdl: message >
< wsdl: portType name = " upmsgSoap" >
< wsdl: operation name = " AddUPMsg" >
< documentation xmlns = " http://schemas. xmlsoap. org/
wsdl/" > 发布规划信息 </documentation >
< wsdl: input message = " tns: AddUPMsgSoapIn" / >
< wsdl: output message = " tns: AddUPMsgSoapOut" / >
</wsdl: operation >
</wsdl: portType >
< wsdl: binding name = " upmsgSoap" type = " tns: upmsgSoap"
>
< soap: binding transport = " http://schemas. xmlsoap. org/
soap/http" style = " document" / >
< wsdl: operation name = " AddUPMsg" >
< soap: operation soapAction = " http://www. UPIS. com/
AddUPMsg" style = " document" / >
< wsdl: input >
< soap: body use = " literal" / >
</wsdl: input >
```

```

    < wsdl:output >
      < soap:body use = "literal" />
    </ wsdl:output >
  </ wsdl:operation >
</ wsdl:binding >
< wsdl:service name = "upmsg" >
  < documentation xmlns = " http://schemas.xmlsoap.org/wsdl/" > 规划信息发布 </documentation >
  < wsdl:port name = " upmsgSoap" binding = " tns: upmsg-Soap" >
    < soap: address location = " http://localhost/UPIS/upmsg.asmx" />
  </ wsdl:port >
</ wsdl:service >
</ wsdl:definitions >

```

同样,可以将 UPIS 中的其它系统进行封装,生成基于 SOAP 的服务框架,在此基础上开发适用于已有系统的适配器,并将服务发布到 UDDI 注册中心。开发后的 UPIS Web 服务,在向 UDDI 注册中心注册之前,需要先对服务进行测试以确保它能够正常运行。在 Visual Studio .NET 开发环境下对于简单的 Web 服务可以直接利用浏览器进行测试。测试后的 UPIS Web 服务,可以通过 UDDI 提供的 Web 页面进行注册,也可以利用 UDDI API 进行编码注册。如果开发者仅集成 UPIS 内部的应用系统,一般采用私有的 UDDI 注册中心。经过封装后的各系统利用 Web 服务通过 Internet 进行相互访问,从而实现基于 Web Services 的 UPIS 系统的内部或外部集成。

按照 Web Services 架构实现的 UPIS 各系统使用基于 XML 的消息作为一种基本的数据通信方法,这搭起了使用不同组件模型、操作系统和编程语言的系统之间的桥梁,共享各系统的数据和操作,可以很容易实现跨平台的分布式体系结构。

4.2 安全性探讨

从简单性和可扩展性出发设计的 SOAP 协议在制定时并没有过多的考虑安全性,但安全性却是基于 Web Services 集成架构的系统实际运行中的一项重要内容,UPIS 系统也不例外。基于 Web Services 架构的 UPIS,其安全性需要考虑的主要因素有身份验证、授权、机密性和完整性。目前,SSL(Secure Socket Layer)和 TLS(Transport Layer Security)被用来提供传输层的 Web Services 安全,SSL/TLS 在点对点的对话中,可以完成身份验证,数据完整性、机密性等基本要求^[10]。网络层的 IPsec 同 SSL/TLS 一样,提供主机身份验证,数据完整性、机密性的功能,但仅有传输层和网络层的安全机制是不够的。在 UPIS 中,还存在规划消息数据

接受和中转被泄漏或丢失的可能性,需要 UPIS 的请求者/提供者必须信任那些中间节点对消息的获得和处理。另外,还要考虑到权限策略控制,数据不仅未被更改,还要求只能由经过正确授权的 UPIS 用户查看。所以,UPIS 的一个完整的 Web Services 安全解决方案应利用 Web Services 模型核心组件的可扩展性,建立一整套基于 SOAP、WSDL、XML 数字签名(XML Digital Signature)、XML 加密(XML Encryption)和 SSL/TLS 等基础技术之上的安全规范。目前,身份和许可权作为常用的两个安全特性,有多种方法可以实现,如用户名/密码、X.509 证书和 Kerberos 等^[11]。下面是一个 XML 数字签名的主要步骤,应用 XML 数字签名可以较好的保证 UPIS 中信息的完整性。当然,XML 数字签名、XML 加密作为一个快速发展的技术,在确保应用系统安全和标准接口方面尚有一些漏洞和问题还需要进一步研究和探索^[12]。

```

< Signature >
  < SignedInfo />
  ( CanonicalMethod )
  ( SignatureMethod )
  ( < Reference( URI = )? >
    ( Transforms )?
    ( DigestMethod )
    ( DigestValue )
  </ Reference > ) +
</ SignedInfo >
( SignatureValue )
( keyInfo )?
( Object ) *
</ Signature >

```

5 结论

Web Services 的最大特点是具有真正意义上的与平台和语言无关的特性。基于 Web Services 的 UPIS 集成框架,较好的克服了传统集成方式的缺陷,可以方便地实现已有系统、新开发的 Web Services 应用等各种系统的集成。这种松散耦合和动态集成,对未来建立无缝跨平台互操作的 UPIS,特别是空间信息的共享与数据交换具有极大的优势,有利于推进城市规划工作的信息化进程,具有很好的发展前景,是未来发展的一个重要趋势。但由于 Web Services 是一项迅速发展的新技术,并不是十分完善,其在互操作、发现机制和安全性等方面还存在一些问题,这些将在 Web 技术不断发展成熟的进程中逐步得以探索和解决。

(下转第 37 页)

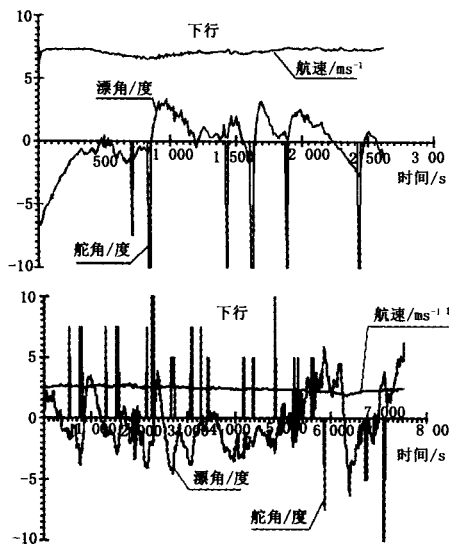


图5 石埠桥桥位建桥后船舶上下行航行参数图

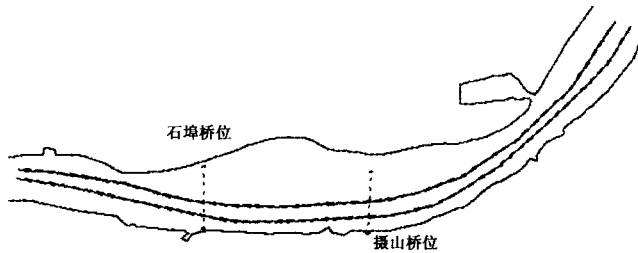


图6 石埠桥桥位建桥后船舶上下行航行迹线图

2) 计算表明,建桥对航线上水流条件几乎没有影

响。从局部来看,石埠桥桥位,桥墩周围最大水位壅高小于7.0 cm,流速最大增加约30 cm/s。摄山桥位,桥墩周围最大水位壅高12.3 cm,流速最大增加约50 cm/s。

3) 综合分析船队的航行参数表明,各桥位、桥型方案对舵角、漂角及对岸航速的影响均较小。

参考文献:

- [1] 文岑,陈桂馥. 船舶航行的数值模拟[J]. 重庆交通学院学报,2000,19(4):102-105.
- [2] 陈明栋,杨胜发,文岑,等. 苏通长江大桥建桥后航线规划研究[J]. 重庆建筑大学学报,2001,23(5):45-49.
- [3] 杨胜发,文岑,陈明栋,等. 长江南京三桥建桥后船舶航行过程数值模拟[J]. 四川大学学报,2003,34(2):39-43.
- [4] 刘应中. 船舶兴波阻力理论[M]. 北京:国防工业出版社,2003.
- [5] 赵月林. 船舶操纵性[M]. 大连:大连海事大学出版社,2000.
- [6] 吴江航,韩庆书. 计算流体力学的理论、方法及应用[M]. 北京:科学出版社,1988.
- [7] 宣国祥,杨朝东,黄岳,等. 船舶航行条件的实时模拟[J]. 水利学报,1999,(8):77-80.
- [8] 陈景秋,赵万里,季振刚,等. 滨江路对两江水流条件影响的研究[J]. 重庆建筑大学学报,2005,27(4):47-51.

(上接第28页)

参考文献:

- [1] 丁建伟,谢明,黄伟. 城市规划信息系统的可持续发展[J]. 城市规划,2001,25(8):19-24.
- [2] 钱聿铭,碳海樵. 基于GIS的数字城乡规划基础平台软件[J]. 重庆建筑大学学报,2005,27(1):26-32.
- [3] 董颖. 混沌带给城市规划的启示[J]. 重庆建筑大学学报,2002,24(1):4-8.
- [4] W3C. Web Service 相关标准[EB/OL]. <http://www.w3c.org>,2005-05.
- [5] 江聪世. 基于Web Services和FETTL的城市空间信息共享与交换技术研究[J]. 武汉大学学报(信息科学版),2004,29(2):105-111.
- [6] Scott Seely. SAOP XML跨平台Web Services开发技术[M]. 北京:机械工业出版社,2002.
- [7] 柴晓路,梁宇奇. Web Service技术、构架和应用[M]. 北京:电子工业出版社,2003.
- [8] 李严,袁一平. 基于Web Services的医疗资源共享设计与实现[J]. 长春工业大学学报(自然科学版),2004,25(1):5-9.
- [9] 杨洁. 基于Web Services的企业应用集成解决方案研究[D]. 兰州:兰州理工大学,2003.
- [10] 田虹,曾德浩. Web Services安全机制研究[J]. 武汉理工大学学报,2005,27(6):95-98.
- [11] 王强,陈平. Web Services安全性及其应用研究[J]. 微机发展,2005,15(6):102-105.
- [12] 卢文锋,张永胜. Web Services及其安全机制研究[J]. 山东科学,2005,18(1):57-62.