

基于 Web 服务数据集成框架的设计与应用

周爱武, 李孙长

(安徽大学 计算机科学与技术学院, 安徽 合肥 230039)

摘 要:随着企业级应用的不断发展,企业越来越重视自己的数据信息,迫切需要解决企业内部之间的“信息孤岛”。针对这些问题,提出了一种基于 Web 服务数据集成框架的设计。通过研究异构关系数据库集成的常用方法以及 XML 与数据库之间的映射之间的方法,从而设计出一种数据集成框架,并运用 J2EE 和 Web Services 相关技术实现该框架。将该框架应用到某供电公司综合在线系统的项目中,通过对该系统的运行情况的分析,它可以有效解决该公司内部之间的“信息孤岛”问题,从而充分验证了该框架的可行性、有效性、实用性。

关键词:信息孤岛;数据集成;异构关系数据库;XML;Web Services

中图分类号:TP311

文献标识码:A

文章编号:1673-629X(2010)11-0246-04

Design and Application of Data Integration Framework Based on Web Services

ZHOU Ai-wu, LI Sun-chang

(College of Computer Science & Technology, Anhui University, Hefei 230039, China)

Abstract: With the growing development of enterprise applications, companies take more and more attention to their own data, and also need to address “Information-islands” between themselves urgently. To solve these problems, a design of Data Integration Framework Based on Web Services has been proposed. Through analyzing the method of Integration of Heterogeneous Relational Databases and the Mapping method between XML and database, a data integration framework which is achieved by J2EE and Web Services has been designed. Finally, the framework is applied to an online integrated system of a power company. Through the analysis of the system, it can effectively solve the “information island” problem of the company. It can prove that the framework is feasible, effective and practical.

Key words: information-islands; data integration; heterogeneous relational databases; XML; Web services

0 引言

由于竞争、合并、收购、地域分布和扩展中不可避免的分散等各方面因素,导致了企业中的各部门常常使用不同的数据库管理系统来存储和管理其重要数据,数据共享很困难,常常形成企业部门内部的信息孤岛。只有将这些系统中的数据信息集成起来,企业才会认识到这些系统所包含数据的整体价值。因此,如何对数据进行有效的集成管理已经成为增强企业商业竞争力、提高企业生产效率的重要因素。国内外高校和企业早就开始对数据集成进行研究,也取得了一些成果。文献[1]介绍了将异构数据库集成应用到电力行业,解决了电厂 IT 系统信息集成的问题,提高了电厂的生产效率和竞争力。文献[2]介绍了将异构数据库集成应用到电子政务,建设一个高效的以人为

本的电子政务系统,从而提高了各级政府部门的办事效率,为市民提供更好更快的服务,让政务公开化,提高了政府形象。文献[3]介绍了基于 Web 服务的 EAI,将数据集成应用到企业内部、企业之间等方面,从而有效地整合了现有的资源。

文中运用 J2EE 平台和 Web Service 技术,设计并实现一个异构关系数据库的数据集成框架,最后将其应用到某电力系统的系统集成项目中,并取得了很好的集成效果。

1 SOA 与 Web Services 简介

一般认为:SOA 是一个组件模型,它将应用程序的不同功能单元——服务,通过服务间定义良好的接口和契约进行联系。

SOA 与 Web Services^[4]的联系:SOA 不是 Web Services, Web Services 只是实现 SOA 的方式之一。SOA 构架是独立于技术实现的,SOA 并不是必须用 Web Services 来实现,同理,Web Services 也不需要完全

收稿日期:2010-03-26;修回日期:2010-06-16

基金项目:国家自然科学基金项目(60773114)

作者简介:周爱武(1965-),女,安徽舒城人,副教授,研究方向为数据库与 WEB 技术。

遵循 SOA 标准。但是 Web Services 的特性十分适合用来实现 SOA 架构。Web Services, 它有比较粗的粒度, 这样较粗的粒度能够构成 SOA 中服务的粒度。Web Services 之间能够交换带结构的文档, 而且这些被交换的文档能够包含完整的异构数据信息, 还可以同时附带其他关于数据的数据, 比如元数据 (Metadata)。与此同时, SOA 也没有确切地定义服务应该如何进行交互, 它通常情况下只是仅仅定义了服务之间应当如何进行相互理解。在需要进行交互的服务之间, 如何进行传递消息时, Web Services 拥有具体的指导原则, 它通常是通过 HTTP 协议传递的 SOAP 消息来实现 SOA 模型。所以, 从本质上讲, Web Services 可以实现 SOA, 而且只是 SOA 实现方式其中的一种。

Web Services 平台主要采用如下四个技术:

XML^[5], 它是 Web Services 平台中表示数据的基本格式, 用来解决数据表示的问题。W3C 制定的 XML Schema 定义了一套标准的数据类型, 并给出了用来扩展这套数据类型的开发语言, Web Services 平台就是用 XML Schema 作为数据类型的系统。

UDDI^[6], 它是通用描述、发现和集成规范。服务提供者可以通过 UDDI 协议注册其要发布的服务, 服务请求者依照 UDDI 规范向 UDDI 注册表搜索相应的 Web Services, 然后得到相应的 WSDL 文档进行服务调用。

WSDL^[7], 它是一种基于 XML、用于描述 Web Services 及其操作、参数以及返回值的语言。它详细描述了服务的完整信息, 包括服务的功能、位置、访问指令以及发送和接收的信息结构, 通过它可以获得 Web Services 调用的所有信息。

SOAP^[8], 它提供了标准的 RPC 方法来调用 Web Services, 并定义了 SOAP 消息的格式, 以及如何通过 HTTP 协议来使用 SOAP。

2 关系数据库集成方法

目前有不少关于异构数据源集成的方法, 但是最常用的也就只有三种: 数据仓库、联邦数据库和 Mediator/Wrapper (中间件)。

2.1 数据仓库

数据仓库^[9] (Data Warehouse), 它是一个面向主题的、集成的、相对稳定的、反映历史变化的数据集合, 通常情况下主要用于支持管理决策。

它的优点是可以用于数据集成, 还可以用于决策的支持, 并且它处理简单, 还可以为用户提供简单而且快速的频繁数据的访问。然后, 它的缺点就是数据没有及时更新、数据的实时性不是很好、数据仓库中保存的通常情况下都不是实时的数据, 而且它的数据一般

都是重复存储的。因此, 一般来说, 它比较适合一些简单的系统。

2.2 联邦数据库

联邦数据库系统^[10] (Federated Database System, FDBS), 它主要是由参与联邦的半自治的数据库系统共同组成的。它要实现数据库系统相互之间的数据共享, 它一般会独立于其他数据库以及联邦的每个数据库的操作, 它们彼此之间一般是通过添加要访问的对方数据库的接口来实现相互之间的数据共享和传递。

联邦数据库要实现互操作时, 通常是先将各数据库和其他数据库模式进行一一映射。因此, 联邦数据库之间如果要想实现全部的模式映射, 则必需建立 $N * (N - 1)$ 个模式映射规则。这样的话, 其缺点就是当数据库太多时, 映射规则的任务将会变得异常繁重。因此, 联邦数据库一般也只适合数据库个数较少的情况。

2.3 Mediator/Wrapper (中间件)

Mediator/Wrapper 方法^[11]也叫中间件方式, 它是一种独立的系统软件以及服务程序, 分布式应用软件将借助于它在不同的技术之间进行资源共享。

中间件方式是当前最常用的一种数据集成方法。它通常提供一个全局模式以及对各数据源的局部模式, 从而可以通过这些模式集成来自不同数据源的数据。它主要是通过由一个 Mediator 和 Wrapper 组件, 把查询的数据转换成一个统一的规范格式, 最后再提供给用户使用。

3 基于 Web Services 关系数据库集成框架

3.1 关系数据库与 XML 之间的映射

关系数据库中的数据主要以相互联系的二维表格存储, 记录单元的值, 具有原子性, 而且记录没有顺序, 数据库结构固定, 数据一般是通过检索相关的多个表而获得。XML 文档中的数据主要以层次化的结构文档组织, 节点能保存元素和属性两种值而且它的元素可以进行嵌套, 并且元素是有顺序的, 文档本身的结构具有自描述性, 也可以直接检索^[12]。XML 与关系数据库的转换如图 1 所示。

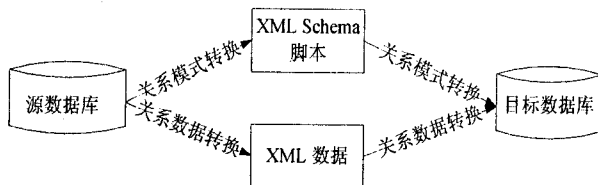


图 1 关系数据库与 XML 的转换

针对关系数据库和 XML 文档的对比区别, 目前主要有两种 XML 与数据库之间的相互映射方法^[13], 分别为: 基于模板驱动的映射方法以及基于模型驱动

的映射方法。其中,基于模板驱动的映射方法,它一般不需要事先定义好 XML 文档和其它数据之间的映射关系。但是它会在相应的 XML 文档中嵌入带参数的 SQL 命令,然后让数据传输中间件来执行对底层数据库的查询,并将获得的结果生成最终的 XML 文档。和基于模板驱动映射方法不同的基于模型驱动映射方法,它通常在把数据从数据库中取出并传送到相应的 XML 文档,或者在将数据从 XML 文档写回到数据库这个过程中,它一般不会仅仅依赖内嵌 SQL 命令,通常是真正的建立数据库模式和 XML 文档结构之间的映射关系。

JAXB^[14](Java Architecture for XML Binding),当前它是业界的一个标准,同时也是可以根据 XML Schema 生成 Java 类的一项技术。它提供了一种将 XML 实例文档反向生成 Java 对象树的方法,与此同时它还可以将 Java 对象树的内容重新写到 XML 实例文档中。JAXB 提供了将 XML 模式绑定到 Java 表示上的一种快速而简便的方法,从而可以使 Java 开发者在 Java 应用程序中能够方便地结合 XML 数据和处理函数来实现相应的数据转换。如图 2 所示,首先利用 xjc 命令行程序编译 DTD 或 Schema 文件生成与 XML 数据结构相对应的 JavaBeans 类,再通过 JAXB API 中的 marshal 与 unmarshal 操作完成 JavaBeans 对象与 XML 数据之间的互换。这样就可以实现 XML 与数据库的映射了。本文设计的框架是使用 JAXB 将对象与 XML 进行数据绑定的。

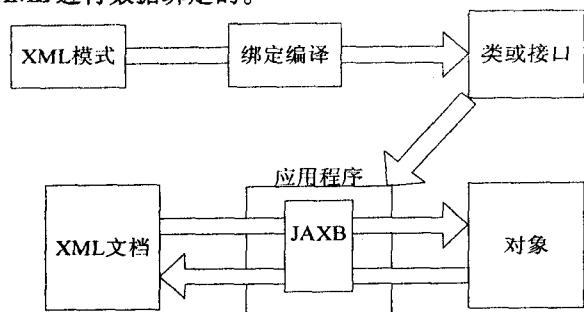


图 2 JAXB 应用框架

3.2 基于 Web Services 关系数据库集成框架设计

为了实现多个关系数据库的数据集成到中央数据库的目的,需要从各个数据源按照一定的抽取规则抽取数据,再根据相应的规则同步到中央数据库中,设计框架如图 3 所示。

该框架主要包括数据源配置管理、数据定时同步的设计、数据同步服务的设计以及数据库连接池的设计等核心内容。它采用 J2EE 平台,运用 JDBC 技术、Web Service 技术等实现的,结合了异构数据源集成的最常用的三种方法(数据仓库、联邦数据库和 Mediator/Wrapper),运用了 XFire、JAXB 等技术实现 Web Service,并根据数据库连接池原理设计了数据库连接池,实现取得实时数据和保存历史数据到中央数据库的功能。

本框架的核心是集成服务的设计,首先需要设计中央数据库与子数据源之间的映射关系,采用基于对象的映射方法实现关系数据库的数据集成。整体思想是使用数据库连接池查询并获得需要的数据记录信息,把这些信息转换成对应的 Java 对象,并用 XFire 写成服务,公开它的接口,提供客户端使用。网络中传输是数据记录的 XML 文档形式,这样需要首先使用 JAXB 把相应的 Java 对象绑定成相应的 XML 文档,再通过 SOAP 进行传输。

最后,简单介绍本集成框架的一些优点:

(1)它采用的是基于模型驱动的映射方法,所以它可以转换来自不同数据源的数据,通常情况下主要包括结构化数据(关系数据库)以及半结构化数据(XML 数据)。

(2)它采用 XML Schema 模式而不是 DTD 模式来进行数据描述,因此它将应当具有更好的数据表现能力。

(3)它采用数据库连接池技术,可以大大减少数据库连接时间以提高数据库的利用效率,还可以对数据源进行管理。

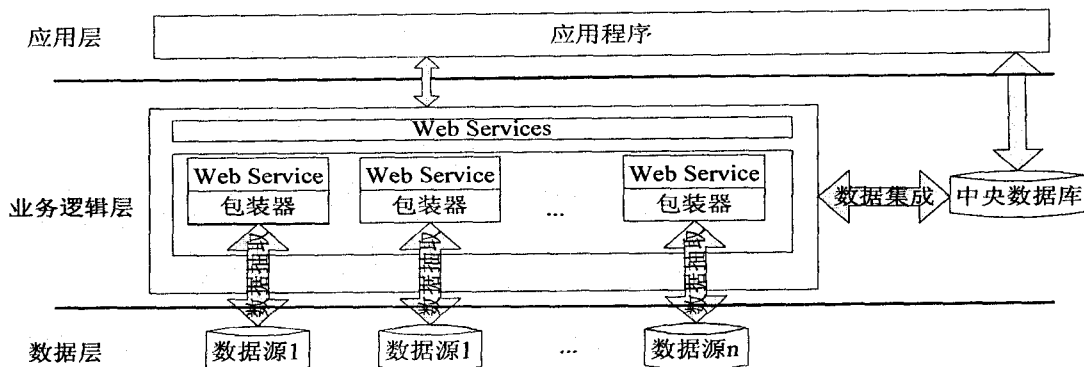


图 3 基于 Web Services 关系数据库集成设计框架

(4)它采用 Web Services 来实现数据同步和数据集成的,这样就可以将应用层与业务逻辑层有效的分开,降低系统的耦合性。

(5)它实现的定时同步功能,可以通过设定时间实现数据定时同步和集成功能。

(6)它解决了数据库之间一些常用的数据冲突问题,实现了不同数据库之间的信息集成。

(7)它是用 J2EE 技术实现的,因此,它可以进行快速开发 Web Services 服务,具有很高的开发效率和较高的组件复用率。同时,它基本上可以满足当前市场上大多数企业的实际应用需求。

3.3 框架在实际项目中的应用

我们将前面设计的数据库集成框架应用到某电力公司在线检测综合系统的数据集成项目中,实现对该公司的六个子系统的数据进行实时同步和集成,这六个子系统分别为:变压器油色谱在线监测系统;SF6 气体密度在线监测系统;开关柜温湿度监测系统;电气设备远红外在线成像分析系统;电能质量在线监测系统;直流屏网络在线监测系统。从而,在公司层面上建立综合的、标准的、可维护的、可再次利用的、可历史追溯分析的变电设备在线检测数据库以及统一监测的使用平台,在一定程度上取消了在线检测“信息孤岛”,同时也提高了在线检测技术应用水平、推进在线监测数据

标准化。

以在线检测综合系统中的电气设备远红外在线成像分析系统的数据集成的实际运行情况来分析该框架的应用效果。下面主要以植物园变电站的 29 路热像仪在 2010-01-24 12:00:00 到 2010-01-24 13:00:00 的数据同步与集成情况为参考,分析实际应用效果。在这段时间内源数据库产生的符合在线检测综合系统数据库抽取条件的数据记录如图 4 所示。

当前在线检测综合系统的对电气设备远红外在线成像分析系统的数据同步周期为 1 小时一次,12:00:00 到 13:00:00 同步后,植物园变电站的 29 路热像仪综合数据库后台记录如图 5 所示。

通过对实际数据的集成情况和综合系统的运行情况的分析,验证了该数据集成框架的有效性和实际应用价值。

4 结束语

在信息一体化的趋势要求和各行各业都需要解决“信息孤岛”的前提下,通过参阅大量的文档,对 Web Services 技术和关系数据库集成方法进行了深入的研究,设计并实现了一种关系数据库集成框架,并将它应用到实际项目中,取得了很好的效果,从而验证了该框架的实际应用价值。

ID	STATION_NAME	RXY_NAME	DEVICE_ID	DEVICE_NAME	TARGET_ID	TIME	IM_H_TM	TM_RANGE	ALARM_TMP	FLAG	HOT_POINT	TYPE	FOCUS	IMAGE_DATA	TATM	VIDEO_ID
631	植物园变电站	2路红外	29	变压器		11 2010-1-24 12:27:30	281.0791	273.74to281.0792	50	1	桩头	大立	29		289.01	2
632	植物园变电站	2路红外	29	变压器		12 2010-1-24 12:29:30	278.7176	275.4042to278.7177	50	1	桩头	大立	33		291.67	2
635	植物园变电站	2路红外	29	变压器		13 2010-1-24 12:31:30	275.3492	271.3144to275.3493	50	1	桩头	大立	28		296.07	2
636	植物园变电站	2路红外	29	变压器		14 2010-1-24 12:33:30	279.6354	273.8643to279.6355	50	1	桩头	大立	28		291.10	2
654	植物园变电站	2路红外	29	变压器		11 2010-1-24 12:43:30	297.8669	282.5262to297.8669	50	1	桩头	大立	29		303.51	2
657	植物园变电站	2路红外	29	变压器		12 2010-1-24 12:45:30	296.8929	282.2725to296.8929	50	1	桩头	大立	33		303.54	2
662	植物园变电站	2路红外	29	变压器		13 2010-1-24 12:47:30	304.2096	282.6348to304.2096	50	1	桩头	大立	28		303.17	2
664	植物园变电站	2路红外	29	变压器		14 2010-1-24 12:49:31	289.3935	286.3611to289.3936	50	1	桩头	大立	28		302.89	2

图 4 源数据库符合综合数据库条件的数据记录

ID	STATION_NAME	RXY_NAME	DEVICE_ID	DEVICE_NAME	TARGET_ID	TIME	IM_H_TM	TM_RANGE	ALARM_TMP	FLAG	HOT_POINT	TYPE	FOCUS	IMAGE_DATA	TATM	VIDEO_ID
631	植物园变电站	2路红外	29	变压器		11 2010-1-24 12:27:30	281.0792	273.74to281.0792	50	1	桩头	大立	29		289.02	2
632	植物园变电站	2路红外	29	变压器		12 2010-1-24 12:29:30	278.71765	275.4042to278.7177	50	1	桩头	大立	33		291.68	2
635	植物园变电站	2路红外	29	变压器		13 2010-1-24 12:31:30	275.34927	271.3144to275.3493	50	1	桩头	大立	28		296.08	2
636	植物园变电站	2路红外	29	变压器		14 2010-1-24 12:33:30	279.6355	273.8643to279.6355	50	1	桩头	大立	28		291.1	2
654	植物园变电站	2路红外	29	变压器		11 2010-1-24 12:43:30	297.8669	282.5262to297.8669	50	1	桩头	大立	29		303.52	2
657	植物园变电站	2路红外	29	变压器		12 2010-1-24 12:45:30	296.8929	282.2725to296.8929	50	1	桩头	大立	33		303.55	2
662	植物园变电站	2路红外	29	变压器		13 2010-1-24 12:47:30	304.2096	282.6348to304.2096	50	1	桩头	大立	28		303.18	2
664	植物园变电站	2路红外	29	变压器		14 2010-1-24 12:49:31	289.39355	286.3611to289.3936	50	1	桩头	大立	28		302.9	2

图 5 同步集成后综合数据库的后台记录

参考文献:

- [1] 李 骏. 电厂 IT 系统信息集成的研究和实现[D]. 上海: 上海交通大学软件学院, 2008.
- [2] 任俊彦. 基于 Web 服务的电子政务系统的集成研究和实现[D]. 上海: 上海交通大学软件学院, 2006.
- [3] 吕 波. 基于 WEB 服务的 EAI 研究与应用[D]. 西安: 长安大学信息工程学院, 2007.
- [4] 林怀恭, 裴瑞华, 罗辉琼, 等. 基于 SOA 架构的服务集成技术的研究[J]. 计算机技术与发展, 2009, 19(7): 141-148.
- [5] W3C. Extensible Markup Language (XML) 1.0 (Fifth Edi-

tion)[EB/OL]. 2008-11-26. <http://www.w3.org/TR/xml/>.

- [6] Clement L, Hatley A, von Riegen C, et al. UDDI Spec TC [EB/OL]. 2006-09-18. http://uddi.org/pubs/uddi_v3.htm.
- [7] W3C. Web Services Description Language (WSDL) 1.1 [EB/OL]. 2001-03-15. <http://www.w3.org/TR/wsdl>.
- [8] W3C. SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)[EB/OL]. 2004-04-27. <http://www>.

(下转封三)

从表 2 可以看出,3GPP 的服务质量(QoS)是如何从 2G 发展到 3G 最后演进到 LTE/SAE 的过程。

在 2G-GPRS 的 R97/98 版本中,语音是基于电路交换的,电路连接建立后即可保证业务的服务质量,所以语音的 QoS 保证比较简单;而 GPRS 采用的是分组交换方式,由于 GSM 无线接口技术的限制,GPRS 只能提供尽力而为的服务质量,这种业务对传输时延和传输速率是没有任何保证的^[9]。

3G-UMTS 的 R99 版本,在对 R97/R98 版本的 QoS 参数进一步细化的同时,也增加了一些新参数,比如前面提到的传输时延和传输速率,其目的是:对已有的和将要出现的所有电路和分组交换服务类型来说,网络的资源分配机制和调度算法要尽可能灵活。这一层面的灵活性和在 UTRAN 中专用传输信道定义的所有可能选项是一致的^[11]。同时 R99 版本按业务流对时延的敏感程度不同定义了 4 种 QoS 的业务类型:会话类、流类、交互类和后台类,每种类型都有相对应的 QoS 具体属性和应用举例^[12]。

发展到 R8 版本的 LTE/SAE,舍弃了电路交换,支持全 IP 网络,服务质量(QoS)的表示相比 R99 版本更加简单。由于包括更少的属性域和预定义的标签,所以减少了可能的组合数量;通过保持不同制造商对相同类型服务的网络应用的一致性,实现用户跨网络的无缝移动的体验;通过预定义的默认承载,缩短了业务建立的时延,实现用户的“永远在线”的体验;通过采用 QCI 和预定义的承载类型(GBR 或 Non-GBR),确定了不同类型的业务的 QoS 具体需求,提供给用户高速且流畅的体验^[6]。

3 结束语

LTE/SAE 系统是移动网络向全 IP 网络发展的重要一步,为了满足其需求,LTE/SAE 的 QoS 做了诸多的改进,引入了许多新的概念:比如简化了 UMTS 的承载架构,发展成 LTE/SAE 的承载业务架构,提出了默认承载、专用承载、GBR 承载和 Non-GBR 承载等

概念;在描述 QoS 参数和属性时,提出了 QCI 和 AMBR 等新概念。文章最后将 LTE/SAE 的 QoS 与 2G 和 3G 的 QoS 比较,表明 LTE/SAE 给用户带来的体验的提高。

随着 LTE 于 2009 年底在瑞典和挪威开始商用,移动通信网络进入了无线高速分组交换时代,这将给 LTE/SAE 的 QoS 管理带来有益的探索,比如 LTE 与 3GPP 的 2G 或 3G 网络切换导致的 QoS 映射问题,LTE 与 Non-3GPP 网络切换导致的 QoS 映射问题等^[6]。

参考文献:

- [1] 周兴围,赵绍刚,李岳梦,等. UMTS LTE/SAE 系统与关键技术详解[M]. 北京:人民邮电出版社,2009.
- [2] 爱立信(中国)通信有限公司. LTE-SAE 体系结构及性能剖析(上)[J]. 电信技术,2008(5):109-112.
- [3] 王晓鸣. LTE/SAE 中全新核心网的架构研究[J]. 电信工程技术与标准化,2009(4):88-91.
- [4] 黄 韬,刘韵洁,张智江,等. LTE/SAE 移动通信网络技术[M]. 北京:人民邮电出版社,2008.
- [5] 沈 嘉,索士强,全海洋,等. 3GPP 长期演进(LTE)技术原理与系统设计[M]. 北京:人民邮电出版社,2008.
- [6] 黄 韬,张智江. EPS 系统的 QoS 机制[J]. 中兴通讯技术,2008,14(6):45-49.
- [7] 3GPP TR23. 882 v8. 0. 0. System Architecture Evolution: Report on Technical Options and Conclusions[S]. 2008.
- [8] Lescuyer P, Lucidarme T. 演进分组系统(EPS):3GS UMTS 的长期演进和系统结构演进[M]. 李晓辉,崔 伟,译. 北京:机械工业出版社,2009.
- [9] 3GPP TS23. 401 v8. 0. 0. General Packet Radio Service (GPRS) Enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) Access[S]. 2007.
- [10] 3GPP TS23. 207 v8. 0. 0. End-to-end Quality of Service (QoS) Concept and Architecture[S]. 2008.
- [11] 蔺 萌,刘正军. 3G UMTS 网络中 Qos 策略分析研究[J]. 微计算机信息,2006,22(3):214-216.
- [12] 3GPP TS23. 107 v9. 0. 0. QoS Concept and Architecture[S]. 2009.

(上接第 249 页)

w3.org/TR/scap12-part1/.

- [9] 王秀娟,曹宝香. 基于面向对象原型法的 N 层数据仓库设计[J]. 计算机技术与发展,2009,19(1):117-124.
- [10] 金远平,王能斌. 联邦数据库管理系统 FDBMS 的设计和实现[J]. 计算机学报,1993,16(6):431-436.
- [11] 冉崇善,王 瑞. 电力企业 EAM 中间件系统的研究与设计[J]. 计算机技术与发展,2008,18(2):194-201.

- [12] 刘 翔,程文青,刘 威. 一种从 XML 建立关系数据库的模式映射方法[J]. 计算机技术与发展,2007,17(2):1-7.
- [13] 耿 飏,宋余庆,梁成全,等. XML 文档到关系数据库映射方法的研究[J]. 计算机应用研究,2010,27(3):951-954.
- [14] Ort E, Mehta B. Java Architecture for XML Binding (JAXB) [EB/OL]. 2003-03. <http://java.sun.com/developer/technicalArticles/WebServices/jaxb/>.