第11卷 第26期 2011年9月
1671—1815(2011)26-6370-07

科 学 技 术 与 工 程
Science Technology and Engineering

Vol. 11 No. 26 Sep. 2011
ⓒ 2011 Sci. Tech. Engrg.

基于 Web 服务的异构数据融合系统

毛 晋 张新家
(西北工业大学自动化学院,西安 710129)

摘 要 提出包含应用层、接口层、集成层、适配层、数据层五层结构数据融合系统。采用 XML 技术解决信息形式异构的转换。采用 Web 服务技术实现适配器设计完成局部模式和数据提取以及转换;采用归并算法完成各数据源数据的融合排序,同时采用线程池技术提高系统整体性能。在 tomcat 和 axis2 架构下部署测试,达到了预期效果。

关键词 数据融合 Web 服务 线程池 XML
中图法分类号 TP393.02; **文献标志码**:A

随着互联网的发展,工程应用中经常遇到需要对数据源分布在不同地点且存在异构的数据进行融合的情况。在信息化的过程中,各科研机构和企业的各部门不可避免的建立了大量的与自身业务相关的信息系统,而这些系统通常是由不同的公司开发且往往是基于不同的平台或不同的后台数据库,造成了关联信息复杂很难实现信息共享。数据融合是实现信息共享的一种技术,以此为出发点设计并实现了一种基于 Web 服务的分布异构数据融合系统,屏蔽了异构数据源的平台数据结构等的异构性,实现了无缝拼接为用户提供了统一透明的访问。

1 系统框架

目前主要的异构数据源集成的体系结构有三种:联邦数据库、数据仓库和中间件。数据仓库优点是既可以数据集成又可以用作对决策支持的查询同时效率高,缺点是数据更新不及时以及数据重复存储。联邦数据库优点是易于实现,缺点是扩展性差且工程量大。中间件法其优点是查询结果更新快并且拥有很好的扩展性,缺点是效率相比其余要低。可见数据仓库适用于对效率要求高,数据源变动不频繁的情况;联邦数据库由于需要编写大量的接口且只支持数据库数据源的集成目前适用较少;中间件法适用于数据源变动频繁且不能预知用户查询要求的情况。

对以上三种方法的比较考虑到在实际应用中数据源的形式多样且对扩宽性的要求越来越高,提出了一种采用基于 Web 服务的面向服务的体系结构的以 XML 为公共模式基于 Web 服务技术对数据源进行访问的融合方式。所谓面向服务的体系结构[1]即系统主要有三种角色构成,服务提供者、服务请求者和注册中心。服务提供者由包装成 Web 服务的数据源和适配器构成,服务请求者由融合中心的构成,注册中心由融合中心的服务管理构成。

融合的最终目的是要为屏蔽底层数据层的实现细节,建立一个数据融合中心,并通过 Web 服务将分布在异地的异构数据库通过数据交换中心连接起来,融合中心需要维护一个注册中心保存各数据源的模式并记录各局部模式与公共模式的映射。

系统拓扑结构如图 1 所示,融合系统在结构上主要由异构数据源、融合服务总线、服务接口三部分构成。该结构在服务总线为服务接口屏蔽了异构数据源的位置、接口等细节,提供了同时采集多个数据源的集成界面。

2011 年 6 月 1 日收到
第一作者简介:毛 晋(1987—),山东泰安人,硕士,研究方向:模式识别与智能系统。

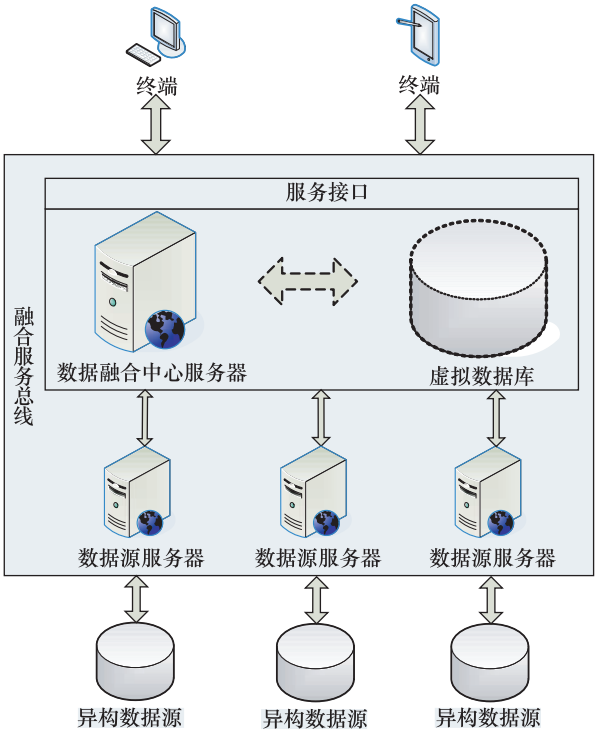


图 1　融合系统拓扑结构图

2　数据融合系统的总体设计

2.1　数据关系模式

数据模式的异构是数据融合过程中一个关键问题,系统的数据模式分为四层,源数据层、局部模式层、公共模式层和用户模式层。

源数据层:在不同的异构数据源中数据是以各种形式保存,关系模型、对象模型以及半结构化的文本模型等。

局部模式层:源数据层的数据要需要以通用的数据类型描述在融合中心进行映射配置。

公共模式层:在融合事务建立之初,在管理员的参与生成的公共模式下表示数据,主要面向用户显示融合后虚库的数据。

用户模式层:根据用户的不同权限,显示其所能访问的服务以及获得的数据,为不同用户显示不同数据。

在这四层数据模式中屏蔽异构,需要完成三次转换。源数据层到局部模式层,局部模式层到公共模式层,公共模式层到用户模式层。源数据到局部模式转换使用 Web 服务操作数据源提取数据实现;局部模式到公共模式由适配器中通过映射关系由模式转换模块实现;公共模式到用户模式由公共模式根据用户权限生成子模式在融合服务总线实现。

2.2　系统结构设计

在系统拓扑结构的基础上,采用分层设计的思想,提出了自上而下包含应用层、接口层、集成层、适配层、数据层五层结构的架构模型,其中接口层、集中层和适配层共同组成了融合服务总线。

融合服务总线定义:由适配器和接口实现的一组基础架构,支持异构环境中服务的交互,提供服务的互操作行。应用层的各种应用和数据层的异构数据源挂接到总线上通过总线实现透明寻址的交互访问。分层结构如图 2 所示。

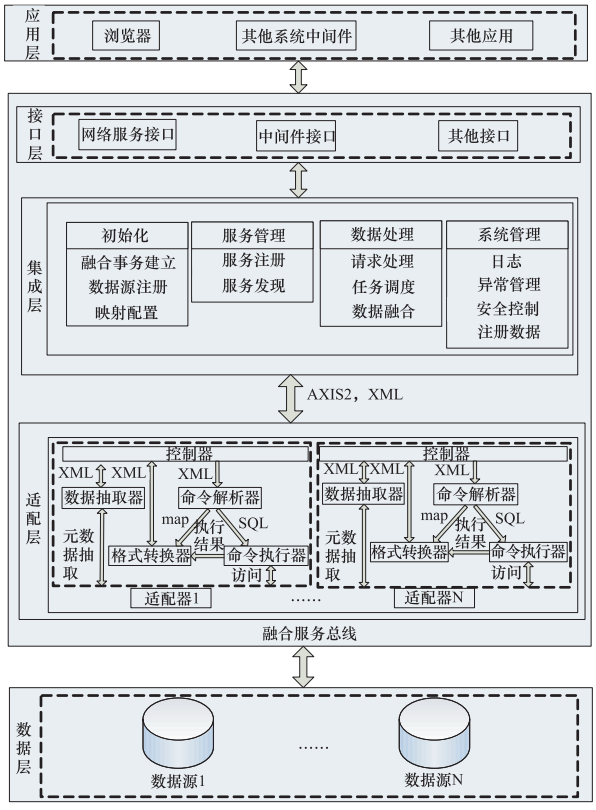


图 2　融合系统架构图

2.2.1　应用层

应用层是面向用户的设计,用于信息的显示和获得。系统从应用层获得用户的输入数据,并将处

理的后的数据显示给用户。用户可以是使用浏览器访问中心服务器的人,也可以是更上层数据处理系统即系统可以是另一个系统的信息处理单元。用户在应用层通过公共模式浏览数据和发送全局命令。

2.2.2 接口层

接口定义了外部世界可以看到的服务功能,并提供了访问功能的方式。依据用户的不同类型,需要不同的协议,对不同的应用请求给不同的格式信息,同时对不同的应用请求给予不同级别的公共数据模式。对于浏览器的访问请求则提供普通的网络服务接口即可。

2.2.3 集成层

集成层是系统的核心层,负责处理用户请求调用服务获得数据进行融合处理返回数据以及融合事务建立以及数据源的注册。

(1) 初始化模块

初始化主要为系统提供公共模式、局部模式以及建立映射关系,在这一过程需要数据库管理员的参与,主要包括事务建立、数据源注册和映射配置。

事务建立的主要实现公共模式生成。公共模式以一种统一的方式描述数据的逻辑结构和特征,用于对用户显示数据和获取用户命令。在业务需求下管理员在融合中心创建融合事务,将所需数据描述为几个表,对数据的描述的提取转化为 XML 格式保存在注册管理,该描述即可视为该次融合事务的公共模式。

数据源注册主要实现局部模式生成。局部模式为各个数据源对自身数据的描述方式。局部模式在数据源注册过程自动生成。数据源的注册包括两个方面,一个是数据源的基本信息,一个是元数据的注册。数据源注册需要提供:数据库的类型、数据库名称、数据库用户名、密码以及 IP 地址,以完成异构数据源的注册。通过数据源的注册,将数据库部署在网络上,实现了访问数据库的位置透明性和本地化。某个数据源注册到本系统之后,系统可以调用适配层的 Web 服务得到数据源的元数据按照固定的 XML 格式保存在集成层的注册管理。该 XML 描述的局部数据源的数据形式即事务局部模式。

建立映射关系消除公共模式与局部模式间的异构性。描述公共模式和局部模式的 XML 文件都保存在集成层的注册中心,数据库管理员在融合中心通过配置工具读取 XML 文件完成映射配置生成映射表。根据映射关系后全局命令同时对应的转换为局部命令并与映射关系同时保存在注册管理。

(2) 数据处理模块

数据处理模块主要实现对数据的获取和处理,是融合系统中最核心的模块,主要包括请求处理组件、任务调度组件和数据融合组件。

请求处理组件完成系统对用户通过接口访问服务器请求受理,主要完成用户请求数据的解析提取请求融合事务代码,根据代码级别和用户权限进行校验,通过后进行融合事务请求处理。

集成层通过对底层的服务调用获得各数据源的数据。使用生成的局部命令作为参数调用适配层的服务,考虑到在一次融合事务要想多点发送请求并处理返回数据,采用多线程并行执行来实现会有更高的效率。但是为每个请求创建一个新线程的开销很大;为每个请求创建新线程的服务器在创建和销毁线程上花费的时间和消耗的系统资源要比花在处理实际的用户请求的时间和资源更多[2]。

线程池是针对这种情形的一种有效解决方案,建立多个可重用的线程,把创新线程和取消进程的开销分摊到了多个任务上,可以有效消除了线程创建所带来的延迟,立即为请求服务,使应用程序响应更快。可以定义线程池的数目来强制其它任何新到的请求一直等待,直到获得一个线程来处理为止,避免了资源不足崩溃的情况出现。具体实现流程图如下所示:

数据融合模块完成将各数据源返回的多个有序结果整合为一个有序结果。数据融合的关键问题有结果格式验证、结果合并、结果排序。对于调用服务返回的结果首先通过 XML Schema 完成校验,合并和排序通过归并算法一并完成。校验通过后的数据首先由字符串载入 Document 对象以便快

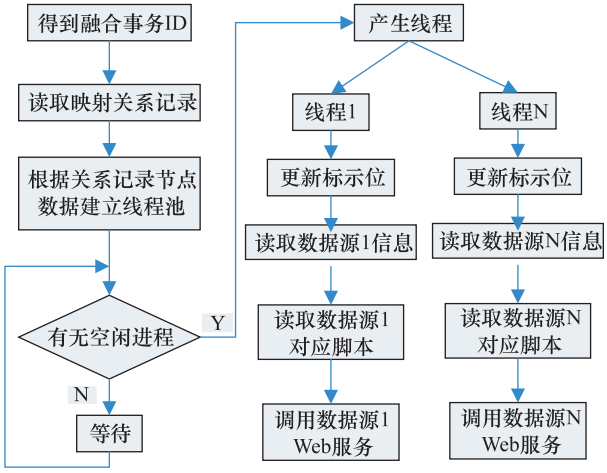


图 3　线程池服务并行调用 Web 服务流程图

速解析出数据。

将调用服务返回的结果合并排序,最终将所有的结果融合成 XML 文件。将两个或多个有序文件组成一个单一的有序文件符合归并排序的适用情况[3]。对每一个融合事务建立一个对象数组保存结点返回的数据,每一个数据源建立一个对象数组保存该节点返回的数据,对返回数据的每一行建立一个对象数组保存解析出的具体数据。对融合事务对象数组存放的各节点对象数组做根据其所保存的具体数据某一项做两两归并,该事务所有的数据源对象数组最终归并为一个对象数组。归并完成后将数据从对象数组中读取转为 XML 文件。

(3) 服务管理模块

由于系统采用基于 web 服务的面向服务的体系结构,所以担任注册中心角色的服务管理模块功能特别重要,在这个模块主要包括服务注册和服务发现两个功能。

服务注册是由两个过程来完成,服务描述和注册。各个数据源将融合业务所需求的服务进行正确的描述,使用 WSDL 正确的描述出 web 服务的业务信息、服务信息和技术信息,为调用者提供服务的功能与完成方式、服务的调用方法和服务的定位与绑定信息。

服务发现是由服务的请求者访问注册中心的搜索服务,得到满足搜索条件的服务的集合,在其中选择最合适 Web 服务。在这过程中,服务调用者通过服务注册时所提供的描述信息,根据业务信息进行选择,得到服务的调用方法和绑定信息,进行调用。

(4) 系统管理模块

注册数据管理将初始化过程中生成的公共模式、局部模式、数据源连接信息以及映射关系保存在融化中心同时向系统中的使用这些数据的组件提供接口。根据系统组件的需要提供所需的数据,降低组件间的耦合性。

日志和异常处理为系统正常运行提供基本的服务,对集成层调用适配层 Web 服务异常的捕获,保证系统不会因为数据源的异常或配置层的错误而导致系统崩溃。

2.2.4　适配层

适配层是异构数据融合系统的重要组成部分,是直接访问抽取异构数据的层次,由分布在各个数据源的适配器组成。适配器包括有两方面的功能:初始化阶段获取数据源的元数据生成局部模式;运行阶段对数据源进行操作并对获取数据完成格式转换。

根据功能需求,将适配器的结构设计由控制器、数据抽取器、命令解析器、格式转换器和命令执行器构成的形式。所有模块都设计成为服务,根据流程的需要在这些服务中相互调用,即提供的安全性和稳定性,又能提供共灵活的访问方式。将这五种服务和数据源共同封装成一个 Web 服务,注册到集成层的服务管理模块,在系统运行的时候提供服务。

(1) 控制器

控制器组件控制对外部请求的响应,对于上层发起的请求进行判断,并对合理的请求进行处理并返回数据,是适配器接收请求和返回数据的门户。其中主要包含有校验函数模块和对其他服务调用的模块。

(2) 数据抽取器

数据抽取器实现了系统初始化阶段对数据源元数据的抽取,完成源数据层到局部模式层的转换。对于数据库在服务中使用 JDBC[4] 访问元数据

的接口,将获得结果集提取出数据转换为规定的XML的结构后返回服务调用方即控制器的服务调用模块。

(3) 命令解析器

命令解析器功能是接收集中层使用多线程发送的局部命令,将局部命令中的执行语句和映射关系提取处理,保存在适配器中,等待命令执行器服务和格式转换器服务在运行时使用。

(4) 命令执行器

命令执行器接收命令解析器处理后产生的标准SQL语句并执行操作并将结果传递格式转换器的组件。为了达到可以访问多种关系数据库的目的,在执行器的设计中使用了JDBC技术。使用JDBC连接数据库后,执行SQL语句,并将返回的结果集处理,提取其中数据并依据定义的局部模式的XML文件结构转换为XML形式传递给格式转换器。

(5) 格式转换器

格式转换器主要实现了局部模式到公共模式转换的映射,并实现了融合结果的初次排序。由于在命令解析器处理后映射关系的基本形式为:公共模式的每一个项都由局部模式的多项式表示。从而依据映射关系中的公共模式结构建立起返回结果的xml结构,即针对全局模式的每一项建立一个空结点,然后根据映射表中记录的该项对应的表达式选取局部模式的项的值做运算后更新到XML结构中。

实现局部模式到公共模式的转换后生成了一个以全局模式描述的局部数据信息的XML文档。为了在集成层中实现了更高的效率实现全局融合排序,需要对该文档进行初次排序。在适配器中对结果的初次排序采用了XSLT[5]中的<xsl:sort>元素来实现[6],设置select属性为排序,转换后生成的新的XML文档即根据排序项排序后的XML文档。将该XML文件经过控制器返回集成层。

2.2.5 *数据层*

数据层是系统中真正数据存放位置,由分布在不同地点采用不同平台不同时期建立的异构数据源组成。通过适配层中提供的适配器接入数据融合总线为融合提供所需要的数据。

3 数据融合系统执行流程

数据融合系统的执行流程分为两个过程:注册过程和运行过程。注册过程为运行过程提供所需要的基础数据包括用户类型、数据源连接信息、公共模式、全局模式、映射关系配置等;运行过程则实现接收用户请求、抽取数据并根据映射关系完成数据转换。运行流畅如下:

(1) 管理员根据实际的业务需求建立融合事务,生成全局模式保存在融合中心数据库。注册数据源,在融合中心的数据库保存每个数据源节点的连接登录信息,抽取数据源的数据元数据生成的各数据源的局部数据模式。

(2) 各数据源数据库管理员在融合中心完成全局模式与局部模式的映射关系,并由映射关系完成全局命令到局部实命令的映射。

(3) 用户注册登录,查看其权限内的某融合事务发送请求。

(4) 系统接收用户请求后向连接在数据融合总线的参与融合事务的数据源节点发送命令。

(5) 数据源通过适配器接收融合中心发送的命令,访问数据源执行命令并将得到的结果根据映射关系进行转换并排序,失败则提示异常并记录。

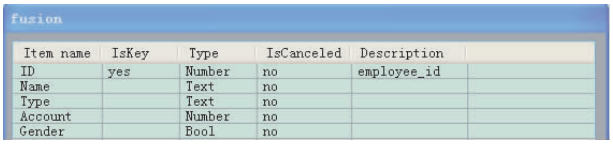
(6) 融合系统接收到从数据源返回的转换好的数据进行校验,进行融合排序,生成XML格式的结果返回给用户。

一次数据融合实例如下所示:

现在总公司M需要分公司A、分公司B的员工数据。A、B公司拥有各自建立的不同人事信息系统,则需要将该数据融合后返回用户。

(1) 总公司在数据融合中心建立所需要的数据信息表,其中列名、类型等可自定义。融合后的结果最终以该表描述的形式进行显示。

融合汇总表建立之后即可根据该表建立几个融合事务,创立事务号,并且形成事务描述语句,比



fusion

Item name	IsKey	Type	IsCanceled	Description
ID	yes	Number	no	employee_id
Name		Text	no	
Type		Text	no	
Account		Number	no	
Gender		Bool	no	

图 4　建表截图

如在此处创立事务,获取员工工号、类型和基本账户,则为事务的脚本描述为:

```
< Action AT = " Get"  ActID = "4"  SQL = " select fusion. Type, fusion. ID, fusion. Account"  TN = "  fusion  "  Epr = "-1" >
```

(2) 融合事务建立之后则可以通知 A、B 公司的数据源向融合中心注册,并提供所需要的数据,即数据源想节点返回所需要的表的元数据。

分公司注册后的信息显示如图 5 所示。

Node_ID	Name	NodeType	ODBCName	NodeIPAddr	NodeDBType	DeployODBCName	Preffix
3	branchA	2	TTSAppBuilderDBSource	20.20.8.31	SQLServer	TTSAppServerDBSource	BJ
5	branchB	2	TTSAppBuilderDBSource	127.0.0.1	Access	TTSAppServerDBSource	XA
1	数据中心	1	TTSAppBuilderDBSource	20.20.8.30	Access	TTSAppServerDBSource	CENTER

图 5　数据源注册截图

节点类型的区分表情数据融合中心与数据融合节点,记录节点的地址与数据库类型以方便连接与适配器的选择。注册同时需要提供融合所需数据的元数据。在本例中提供的数据如下:

A 点注册元数据

```
< MetaData >
< table name = " employee" >
< columns >
< item name = " name" type = " CHAR" size = "20" nullable = "0" / >
< item name = " empId"  type = " INT" size = "10" nullable = "0" / >
< item name = " gender" type = " CHAR" size = "2" nullable = "1" / >
< item name = " account" type = "  INT" size = "20" nullable = "1" / >
< item name = " type"  type = " CHAR" size = "10" nullable = "1" / >
< premaryKey > employeeId </premaryKey >
</columns >
</table >
</MetaData >
```

B 点注册元数据

```
< MetaData >
< table name = " groupinfo1" >
< columns >
< item name = " wName" type = " CHAR" size = "20" nullable = "0" / >
< item name = " wId" type = " INT" size = "10" nullable = "0" / >
< item name = " sex" type = " CHAR" size = "2" nullable = "1" / >
< item name = " wtype" type = " CHAR" size = "10" nullable = "1" / >
< item name = " account1" type = "  INT" size = "20" nullable = "1" / >
< item name = " account2" type = "  INT" size = "20" nullable = "1" / >
< premaryKey > wId </premaryKey >
</columns >
</table >
```

(3) 在得到节点元数据之后进行映射关系配置,在数据库管理员的参与下可以生产映射关系配置表。在本例中对 A 点产生的映射脚本如下:

```
< Action AT = " Get"  ActID = "2"  SQL = " select employee. type, employee. empId , employee. account "  TN = "  employee "  Epr = "  -1" >
< MapItem >
< item name = " fusion. Type"  Expr = " employee.  type  " Type = " CHAR" / >
< item name = " fusion. ID"  Expr = "  employee.  empId  "  Type = " INT" / >
< item name = " fusion. Account"  Expr = "  employee. account"  Type = " INT" / >
</MapItem >
</Action >
```

本例中对 B 点产生色映射脚本如下:

```
< Action AT = " Get"  ActID = "3"  SQL = " select groupinfo1. wtype, groupinfo1. wId, groupinfo1. account1 , groupinfo1. account2 "  TN = " groupinfo1"  Epr = "  -1" >
< MapItem >
< item name = " fusion. Name"  Expr = "  groupinfo1.  wtype  " Type = " CHAR" / >
< item name = " fusion. Name. ID"  Expr = "  employee.  wId  "  Type = " INT" / >
< item name = " fusion. Name. Account"  Expr = "  employee. account1  +  employee. account1"  Type = " INT" / >
</MapItem >
</Action >
```

脚本文件分为两个部分,Action 部分是对融合事务描述语句进行拆分得到的具体执行语句,MapItem 部分是事务建立之初的全局模式与局部模式的对应关系。

(4) 配置完成后,用户可以请求该事务。融合中心接收到用户的融合请求后,将第三步中产生的脚本作为参数分别调用部署在 A 点和 B 点的 Web 服务,得到处理后的结果。调用 Web 服务的代码如下:

```
GetSQL2Stub stub2  =  new GetSQL2Stub( );
```

```
GetSQL2Stub. GetSQL2 request2 = new GetSQL2Stub. GetSQL2( );
request2. setStrXML( xmlStr2 ); // xmlStr2 为脚本
String results = stub2. GetSQL2( request2 ). get_return( );
```

(5) 调用服务后得到分布在 A 点和 B 点的经过转换后的以 XML 表示的数据,经过并归处理后按主键排序 XML 形式的结果。得到最终的融合后的结果后进行显示或进一步处理。

```
< ResultData >
< table >
< row Source = A >
< ID >4 </ ID >
< Type > sales </Type >
< Account >4550 </Account > </row >
< row Source = B >
< ID >5/ ID >
< Type > engineer </Type >
< Account >3000 </Account >
……
</table >
</RsultData >
```

4 结论

设计了一种基于 Web 服务的方式实现数据融合,对数据源的只要求提供特定的接口,能实现数据源的"即插即用"的较好扩展性。融合系统在 Myeclipse6,JDK6 的环境下编写调试代码,部署在 Apache Tomcat6 服务器 Axis2 Web 服务引擎架构下对类型为 MS Access 和 Mysql 两种异构数据源进行测试。结果表明,该系统可以实现异构数据源的信息融合,并且能够便捷的增减数据源拥有比较突出的可扩展性和通用性。

参 考 文 献

1 Papazoglou M P. Web 服务:原理和技术. 龚 玲,张云涛,译. 北京:机械工业出版社,2009

2 Goetz B. Java 理论与实践: 线程池与工作队列. http://www.ibm.com/developerworks/cn/java/j-jtp0730, 2002

3 刘大有,虞强源,杨 博,等. 数据结构(第二版). 北京:高等教育出版社,2010

4 张晓东. Java 数据库高级教程. 北京:清华大学出版社, 2004

5 Harold E R. Java 语言与 XML 处理教程. 刘文红,赵伟明,译. 北京:电子工业出版社,2003

6 侯要红,栗宋涛. Java XML 应用程序设计. 北京:机械工业出版社,2007

Integrated System of Heterogeneous Data Based on Web Services

MAO Jin, ZHANG Xin-jia

(School of Automation , Northwestern Polytechnical University, Xi' an 710129, P. R. China)

[**Abstract**] A data integration architecture consisted of application-layer, interface layer, data integration layer, adapter layer and data layer are presented. XML technology is used to solve data heterogeneous transformation; Web services technology is used to achieve adapter design used to extract local model and data conversion. Merging algorithm is used to complete each data fusion results sorting; using thread pool technology is improved the overall system performance. The deployment tests in tomcat and axis2 framework show that this architecture could achieve prospective effect.

[**Key words**] data fusion Web services thread pool XML