基于多层 B/S 模式的数据交换原型系统研究*)

田 东 徐 玲 潘大四
(重庆大学计算机学院 重庆 400030)

摘 要 以XML作为数据交换的中间体,以SOAP协议的消息机制为传输手段,以符合EJB2.0的CMP模型的实体Beans类化关系数据表,提出了一种基于多层B/S模式、能无阻碍穿透企业防火墙、跨平台的安全数据交换原型系统,给出了原型的详细设计及实现过程,并应用于实际系统中。实践证明,这种以前端JavaBean作为Web服务的功能实现,生成Web服务的系统,极大地方便了数据请求端进行XML数据文档解析,能有效地将数据添加到本地数据库中,构建满足用户所需的、结构清晰的XML文档,达到了真正意义上的跨平台操作。

关键词 XML,数据交换,映射文档,J2EE

The Design and Implementation of Data Exchange Prototype System Based on Multi-tiers B/S Model

TIANG Dong XU Ling PAN Da-Si
(College of Computer, Chonqing University, Chongqing 400044)

Abstract A exchange data architecture is put forward based on the multi-tiers B/S model. It uses critical technology such as XML as medium, Web service based on the soap proxy for transport the message, with CMP entity beans to wrapper the database table with object oriented, which penetrated the firewall unblocked. Take this JavaBean as the implement for the Web service provider and generate Web service. It added the data to local database and generates a clear and user needed XML document , which enabled the system to be independent of platform realistically.

Keywords XML, Data exchange, Mapping template, J2EE

1 引言

随着互联网的迅猛发展,信息资源为工作、学习和生活提供了极大的方便。然而,企业的信息系统和以往相比已发生了很大的变化,一个大型系统有很多分支系统,各分支系统可能采用不同平台、不同的数据库管理系统,异质异构的数据越来越多,这对数据一致性、信息共享和决策支持都带来了阻碍。随着数据库平台、操作系统、应用程序开发语言的多样化,异质异构数据的增多,数据集成和数据交换的需求的增加。传统平台内分散数据集成和数据交换模式的缺点逐渐暴露出来,给企业的数据共享带来了严峻的挑战[1]:

·现阶段的数据交换方式大多为基于C/S设计模式,以这种方式通过因特网连接异地的数据库必须要求对方开放防火墙端口进行数据交换。因此从本质上存在安全问题。

·现阶段的数据交换往往是针对某个具体的数据库系统。随着数据库系统不断增多或后端数据库结构发生变化,应用程序无法再进行扩充。

·大多数基于Web的数据交换系统采用了自行定义交换格式,无法满足日益增多的结化数据,如XML格式的数据。

·无法通过Web进行在线数据交换。跨地域传送数据的方式依靠email、FTP系统、

Socket Code、VPN,这些方式也同样存在开放端口的安全问题。

目前数据交换学术界大多利用XML模式文档来描述数据库模式,然后通过XML模式文档作为中介在异地数据库中生成新的数据库模式,或者通过XML模式文档作为两个不同数据库模式的中间模式进行数据交换[2],另外有利用将关系实体模型构造成层次模型的方法[3],但这些理论较难实用化。

本文在深入研究目前国内外数据交换技术的基础上,有效地结合XML、Web Service以及J2EE体系架构和符合EJB2.0规范的CMP实体Bean的优越性,给出了一种新的基于B/S模式的多层结构以及分层组件的开发技术的网络数据交换体系架构,对多层体系架构使用UML进行了建模分析;分析了Facade设计模式在设计EJB组件体系结构时的优越性,以及符合EJB2.0的CMP实体Beans类化关系数据表的优越性。对分层组件结构和各子模块给出详细设计等,从而为企业之间的异构数据交换提供了一种实用的解决方案。

2 多层B/S模式数据交换原型系统的设计

2.1 体系结构方案

现代企业对数据共享和集成有着迫切需求,目前企业信息系统大多已经运行在Web模式下,而现有数据交换应用却主要在C/S模式下开发,无法安全地实现跨地域传输数据,即使可以跨地域也存在开放防火墙端口的安全隐患,另外基于C/S模式的部署和维护相对比较困难等等,不能很好地满足企业需求。相对于传统的C/S模式,B/S模式具有网络和后端系统对用户透明的特点,给调试、日常维护和升级都带来极大的便利,只要专注于服务器一端就可以了,同时给远程管

*)基金项目:重庆市自然科学基金支持项目(7969)。田 东 博士研究生,主要研究方向:软件工程、集群和网格计算技术。徐 玲 硕士,主要研究方向:软件工程、数据库安全。

理节省了大量时间和费用。在系统安全性、稳定性以及实时处理等方面，B/S 模式较 C/S 结构相比有了更为可靠的保障，这使得 B/S 模式成为应用系统发展的主流模式，并且可以与企业的电于商务系统无缝连接，使企业之间通过 Internet 就能方便地对获得所需的数据。为此，在深入研究现有 B/S 模式的基础上，结合企业的现实需求，本文首先设计了一种基于 J2EE 平台和 SOAP 传输机制的多层 B/S 模式体系结构，其内部层次结构如图 1 所示。各模块的功能如下：

(1) SOAP Client：Web 服务请求者。由数据请求端调用，发送请求数据的消息；

(2) 应用服务层：其中包含所有业务逻辑，可以根据完成业务功能的不同或者为了提高伸缩性细分为四个子层，并由不同的组件构成：

1) SOAP Server 层：负责接收 SOAP Client 层传来的 SOAP 请求，对其进行解释并调用 Java Bean 层中的方法，得到执行结果后，返回给 SOAP Client，完成传输功能；

2) Java Bean 层：作为 Session EJB 的接口，由 JavaBean 调用 Session Bean，Session Bean 实现获得所需的数据的业务并传递给 JavaBean 层，然后在 JavaBean 层中按照数据请求端提供的映射文档模板，将获得的数据进行转换，形成用户所需的 XML 文档类型，并把这层作为 Web 服务的实现，生成 Web 服务；

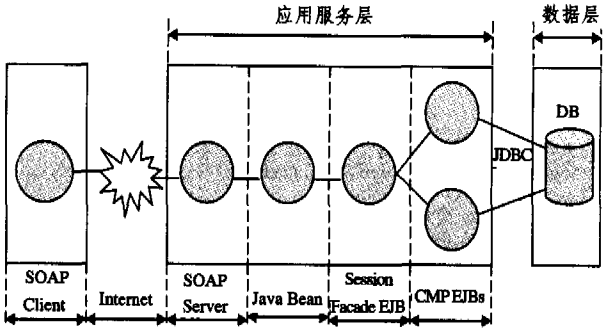

图 1　多层体系结构的层次图

3) Session Facade EJB 层：数据提取业务逻辑层，完成独立于具体数据库的数据提取业务功能。用一个 Session Bean 作为 facade 包装 Entity Bean，负责调用 Entity Bean 的方法，客户端只允许与 Session Bean 交互(本文中需要通过 JavaBean 层，作用是返回给用户 XML 文档)，从而缩短系统响应时间，减少资源利用，并隐藏了与本地接口相关联的复杂性，通过保留 Entity Bean 属性的本地高速缓存来提供客户端对 Entity Bean 的快速访问；

4) Entity CMP EJBs：CMP 模式是将 EJB 持久性管理交给 EJB Container 来完成，开发者一般只要在 EJB 对象的 Deployment Descriptor 中的 Container Managed 字段属性中指定 EJB 实例对象的持久性作用域即可，使用 CMP 实体 Beans 对数据表进行封装，生成数据表类，为不同的数据库产品提供组件集，并建立数据库引擎负责调用请求的组件。对应用程序开发者屏蔽后台数据库。当数据库升级甚至更换时，无需改变应用程序的其他部分。使得应用程序开发者不必考虑数据库系统的类型。

(3) 数据存储层：为应用层的商业逻辑提供进行分析处理的数据，通常采用关系数据库管理系统，如 Oracle、Microsoft SQL Server 和 DB2 等。

2.2　UML 对体系结构的建模

运用 UML 对多层体系结构的构成进行建模分析。用户通过浏览器，发送请求到一个 InfoMasterView Servlet，如果没有信息返回，则通过 JavaBean 调用 InquiryInfo，并执行 Web 服务调用，于是服务器端执行 InquiryInfo(JavaBean) Web 服务的功能实现。由无状态会话 Bean 的实现从数据库中提取数据，数据返回到 Web 服务的功能模块 InquiryInfo 再按照用户提供的转换规则，形成用户所需的 XML 文档，由 Web 服务返回给用户，用户端获得返回数据，进行解析，添加到本地数据库。序列图如 2 所示，其执行过程为：

1) InquireInfo：Servlet 或者 JSP，服务请求的引起；

2) InqireInfoProxy：Web 服务提供的 Client Proxy，由 Servlet 实现，发送数据请求消息，并对服务器端业务处理结果的返回数据进行解析处理，添加到本地数据库；

3) RouterServlet：进行消息传递；

4) InquiryInfo：JavaBean，作为 InfoFacade Session Bean 的接口，调用 Session Bean 获得数据，并将数据按用户需求转换成 XML 文档；同时作为 Web 服务的实现，生成 Web 服务；

5) InfoFacade，无状态会话 Bean，完成按用户需求从数据库提取数据的业务逻辑功能；

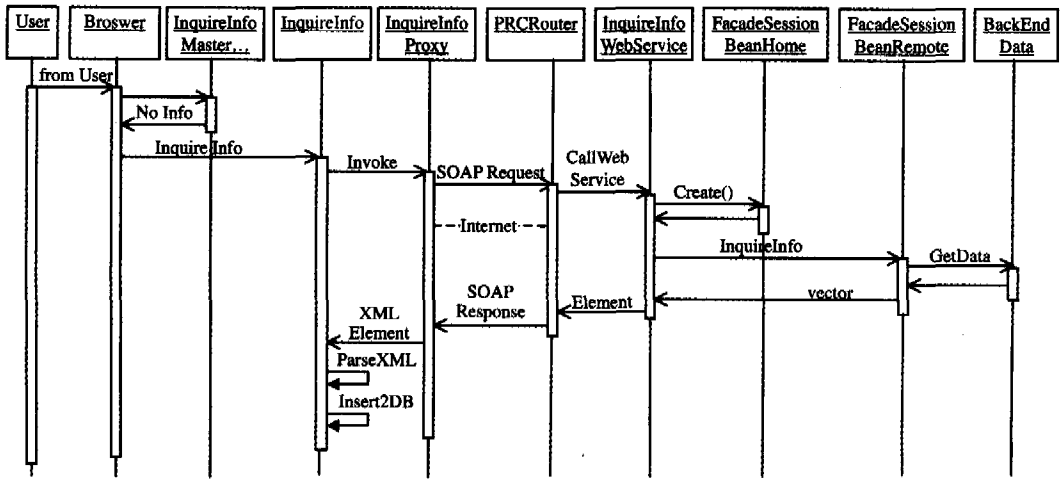

图 2　多层体系结构设计的序列图

3 基于 J2EE 平台的数据交换系统实现

本部分结合 J2EE 多层组件技术，以 Web Service 提供数据转换服务、异步 SOAP 消息机制传输结构化数据，实现了前述基于多层 B/S 模式的原型系统。其工作过程如图 3 所示，A 企业以数据库 DB2 存放着信息，B 企业以 SQL Server 数据库存放着不同的信息，A 企业与 B 企业在数据库服务器操作系统上不同、数据表结构也与 A 企业不同，即完全满足异构数据的环境。本系统的目标是要运用 XML、SOAP、Web Service 以及 EJB 等技术的结合，实现一种新的基于 XML 映射文档、通过 Web 服务将 A 企业中的信息按照 B 企业的要求提取数据到 B 企业中相应的数据表中，反之可以将 B 企业中的数据传送给 A 企业，以完成数据的跨平台、跨地域、独立于具体数据库的异构数据交换，实现数据共享的要求。

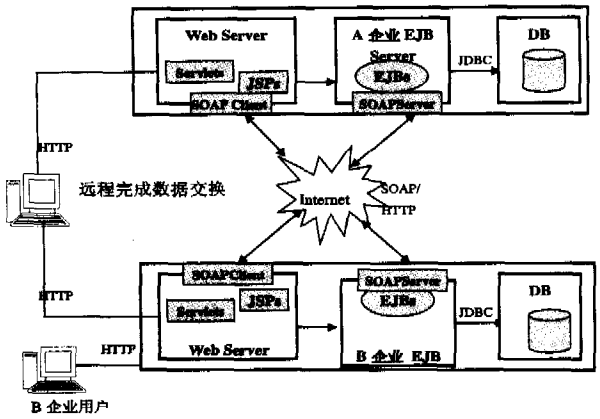

图 3 系统实现机理

4 数据实体层的实现

本层实现了关系数据到对象的转换，对外提供高性能的对象访问服务。CMP 实体 Bean：以 EJB2.0 的规范对数据库表进行封装，生成一个实体 Bean 封装一张数据表由以下类结构定义组成：

EJBNameHome：本地 home 接口，这是一个类工厂，用于创建和发现 EJB 实例。客户端使用 Home 接口生成 EJB 对象。EJB 要定义好所有的 Home 接口支持的方法。

EJBName：Remote 接口，这个类决定哪些方法可以被远程调用。在 EJB 中，客户端代码与 Remote 接口发生联系，不直接与 Enterprise Bean 发生关系。Remote 接口必须遵守 EJB 规范定义的特殊规则，所有的 Remote 接口必须继承通用接口 javax.ejb.EJBObject，扩展了 java.rmi.Remote，可以从另一个不同的 Java 虚拟机上调用。

1）EJBNameBean：类的实现，类中方法的实现逻辑。

2）EJBNameKey：键类，用于表示实体的唯一性。它标识了唯一地标识该 Bean 的每个实例的属性（或一组属性），并提供了用于创建和操作键的方法。

5 业务逻辑层的实现

在 PartFacadeBean（SessionBean）类中实现从数据库提取数据的业务逻辑功能，在 Session Bean 实现类中，EJB 需要返回可序列化的类型，因此首先需要构造可序列化的 JavaBean 类，继承 java.io.serializable 扩展接口，类中的属性为用户需要的数据，JavaBean 的类定义结构如图 4 所示，PartFacadeBean 类（如图 5 所示），类中的 InquirePart() 方法，实现按照用户提供的关键字进行查询，将获得的数据存储在 vector 向量中返回到数据请求端。

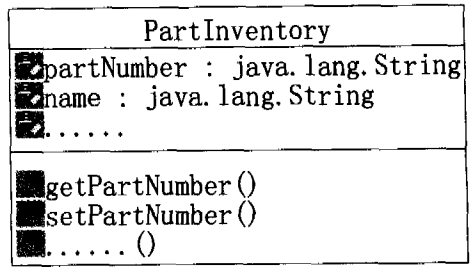

图 4 序列化 JavaBean 类图

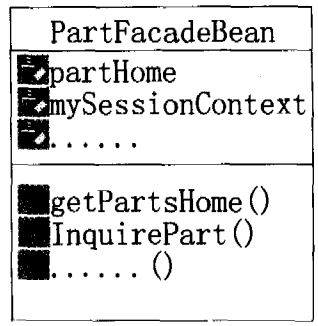

图 5 Session Bean 类图

以下是 InquirePart() 方法提取数据的实现过程简化描述：

```
    //…
    //(1)通过关键字获得实体 Bean Part 的本地接口
Part=getPartsHome(). findByPrimaryKey
   (new partKey(partNumber));
  //(2)通过 part 检索 inventory
    Iterator it = items. iterator();
    Collection items = part. getInventory();
    Iterator it = items. iterator();
    while (it. hasNext()) {
        inv=(inventory)
javax. rmi. PortableRemoteObject. narrow(
    it. next(), Wdxinventory Local. class);
  //(3)实例化 PartInventory JavaBean，以对象保存数据
    partInv = new PartInventory();
  //(4)将数据存放到 PartInventory 对象中 partInv. setPartNumber
    (partNumber);
        //…
    partInv. setLocation(inv. getLocation());
  //(5)将保存数据对象的对象添加到向量中，返回给调用者
    resultVector. addElement(partInv);
```

6 Web 服务数据转换业务逻辑的实现

采用 Web 服务作为数据传输手段，目的是利用 Web 服务的开放性、跨平台以及松散的请求与响应耦合，实现在异构环境下的数据传输。Web 服务请求者通过 Web 服务提供者从异构异质的跨地域的数据库中提取数据，并以符合服务请求者的格式形成 XML 文件返回。

创建一个 Web 服务，包括 WSDL 文档，部署描述以及 Proxy。Proxy 提供了远程过程调用的接口，各种不同开发环境都提供工具支持 WSDL 的生成和部署，如 WSAD 的 wsdl-gen.bat 或者是 Microsoft .net 的 Wsdl.exe，都支持 Web 服务的自动生成与部署。

Web Service 构建的类图 Web 服务响应端按数据请求端的映射文档模板转换数据形成 XML 文档，通过 SOAP 消息传递到 Web 服务请求端。程序运行后，以 IBM WSAD 集成

（下转第 162 页）

到目前为止，虽然还没有一种最优的本体表示语言，但是可以根据实际应用的需要选择最适合的语言。由于某一种语言不可能同时兼备表达性和推理性，只能在两者之间达到一种平衡，但是不同的应用领域对知识表示和推理的需求不同，某些语言可能比其他语言更为合适，因而，在以后的研究中，我们的一个主要任务就是找出影响创建本体的关键因素，这样才能从众多的语言中选取最合适的。

尽管目前还没有评价本体语言的统一的标准，但是我们仍然可以从上述的分析比较中总结出一些规律。

·需要交换不同应用程序之间的本体时，采用基于 Web 的语言比较适合，因为这些语言都是基于 XML 的，因而比较容易被用户理解和管理。除此之外，这些语言还有一个很大的优势，它们受到各个科研组织的强力支持，因此就有更多可利用的编辑工具。

·如果对于表达性的要求较高，则使用传统本体语言比较合适，不过，如果只是将本体看作一些类别，那么基于 Web 的语言也可以胜任。

·由于基于 XML 的语言没有提供推理引擎，因此如果要在 agent 内实现推理，它们并不适合，而一些传统语言不仅提供了推理引擎而且还可以将它们转换成其他可计算的语言。

·在电子商务领域中，本体通常是用来表示电子商务平台中所提供的产品和服务，并分门别类以便用户浏览。因而对于表达的要求并不高，而对推理的需求就要高一些，例如，自动分类对于自动地将这些产品和服务归类是非常有用的，因此，在这里使用基于描述逻辑的语言是很有帮助的。

·创建顶层本体要求很强的表达性，对于推理没有太高的要求，因此，顶层本体通常是用基于描述逻辑的语言来表示的，比如 LOOM。Cyc 工程是通用本体中一个最典型的例子，它的知识库采用 CycL 语言表示，这种语言是基于框架和一阶逻辑的。

上述指导原则对于开发者选择适当的本体语言是大有帮助的，但实际工作中情况复杂多变，应该根据实际需要，灵活地进行选择，这样才能提高开发和研究工作的效率。

目前，国际上（尤其是欧洲）在本体研究领域处于领先地位，推出了一系列的本体开发方法、开发工具和相关标准，而国内在这方面才刚刚起步，研究成果还很少。不过，总体说来，本体技术还不是十分成熟，仅仅从本体表示语言这个小小的领域来看，还有很多问题尚待解决。比如，由于本体语言在表达和推理能力中存在或多或少的缺陷，在不能改变语言本身特性的情况下，我们可以考虑从语言外部来弥补这些缺陷，例如，本体标记语言不能表示形式化公理，但是可以在这些语言基础之上建立逻辑层，这样，就能具有语言本身不具有的逻辑特征。另外，除了以往开发工作的经验和规律，找到影响本体开发的关键因素也是一个需要解决的问题。总之，本体是一个具有很大机遇和挑战性的研究领域，需要广大科学工作者的共同努力。

参 考 文 献

1 Corcho O, Gómez-Pérez A. A Roadmap to Ontology Specification Languages. EKAW 2000, LNAI1937, 2000. 80～96
2 顾芳，曹存根. 知识工程中的本体研究现状与存在问题. 计算机科学，2004，31(10)：1～10
3 Berners-Lee T, Hendler J, Lassila O. The Semantic Web. Scientific American, 2001
4 McGuinness D, Harmelen F. OWL Web Ontology Language Overview. Feb. 2004. http://www.w3.org/TR/owl-features
5 Smith M, Welty C, McGuinness D. OWL Web Ontology Language Guide. Oct. 2004. http://www.w3.org/TR/owl-guide
6 http://www.daml.org/2000/10/daml-ont.html, Oct, 2004
7 Bray T, Paoli J, et al. Extensible Markup Language (XML) 1.0 (Third Edition). W3C Recommendation. Feb. 2004. http://www.w3.org/TR/REC-xml.
8 Brickley D, Guha R V. Resource Description Framework (RDF) Schema Specification. W3C Proposed Recommendation. Oct. 2004. http://www.w3.org/TR/PR-rdf-schema
9 Fensel D, et al. On-To-Knowledge: Ontology-Based Tools for Knowledge Management, 2001
10 Lim S Y, Song M H, Lee S J. The Construction of Domain Ontology and Its Application to Document Retrieval. ADVIS 2004. 117～127
11 Gómez-Pérez A, Fernandez-Lopez, Corcho O. Ontology Engineering. Springer published, 2003
12 Corcho O, Gómez-Pérez A. Evaluating knowledge representation and reasoning capabilities of ontology specification languages. In: Proc. of the ECAI 2000 Workshop on Applications of Ontologies and Problem-Solving Methods, Berlin, 2000
13 Su Xiaomeng, Ilebrekke L. A Comparative Study of Ontology Languages and Tools. In: Proc. of Conf. on Advanced Information System Engineering (CAiSE' 02), Toronto, Canada, 2002, Springer, 2002
14 杜小勇，李曼，王大治. 语言 Web 与本体研究综述. 计算机应用，2004(10)：14～20

（上接第 134 页）

开发环境提供的 Web 服务客户端测试工具 UTC 进行测试，测试结果表明用户所需的 Element 类型数据已经通过 Internet 传递到本地。

Web Service 客户端负责发送请求，调用服务器端的服务，从服务器端取得按照客户端 XML 映射文档模板生成的 XML 文档，取得这个数据以后，客户端就可以根据本地的数据库模式，解析 XML 文档，将数据添加到本地数据库，从而完成数据交换。

最终在客户端得到 Web 服务响应端返回的 Element 类型的数据文档，解析该包含共享数据的文档，将数据添加到本地数据库相应的数据表中。

结论 本文在深入研究现有数据交换技术的基础上，针对其存在的不足，结合现代企业对数据交换的现实需求，提出了一种全新的、基于多层 B/S 模式的、安全易用的数据交换原型系统。在此基础上，基于 J2EE 平台下实现了该原型系统，并在实际应用中得到了验证。该系统以前端 JavaBean 包装 Session Bean，完成数据转换业务逻辑，有效地解决了难以构建满足用户所需的、结构清晰的 XML 文档的难题；以前端 JavaBean 作为 Web 服务的功能实现，生成 Web 服务，达到真正意义上跨平台，无阻碍地穿透企业防火墙；以符合 EJB2.0 的 CMP 实体 Beans 类化关系数据表，使得业务逻辑的实现与后端具体数据库管理系统相互独立，降低设计的复杂度，提高系统整体的性能及安全性；保证了从源端系统提取、转换后传回的数据信息符合本地数据库模式，极大地方便了本地数据文档的解析，具有较强的推广应用价值。

参 考 文 献

1 Williams K. Programming Professional XML Databases. Wrox press, 2000. 701～722
2 Shanmugasundaram J. Bridging relational technology and XML [Dissertation]. Dissertation Abstracts International, Volume: 62-07, Section: B: 3277
3 Hojabri B. Algorithms for generating XML documents from hierarchical views of relational databases: [Dissertation]. Masters Abstracts International, Volume: 41-04: 1107
4 李军怀，张景，等. 基于 XML 的企业异构数据集成方法研究. 计算机工程，2002，28(9)：63
5 聂培尧，魏振刚. 一种基于 XML 的数据集成系统及实现. 计算机应用，2002. 22～29