

面向行业数据交换中间件 EasySwitch 的系统设计与实现^{*}

殷庆 刘卫宁

(重庆大学计算机科学与工程学院 重庆400044)

摘要 企业信息化的深入开展和电子商务的推进,各企业应用系统间数据交换的需求日益增加,但是各应用系统间由于操作平台和数据格式等差异性给交换带来许多困难。针对这种情况,本文实现了一个基于 SOAP/XML-RPC 机制的数据交换中间件系统(EasySwitch),它具有可重构的数据交换核心和面向行业应用的特点,为企业应用系统提供了一个实用化的、基于数据交换的应用协作平台。该系统定位清晰,服务对象明确,实用性强,完整地解决了行业内各企业应用系统间的异构数据交流问题。本文在分析了数据交换的基本原理与思想后,着重阐述了 EasySwitch 的系统架构、应用集成功能设计和各部分核心实现技术。

关键词 数据交换, SOAP, XML-RPC, 应用集成, Web 服务

The Design and Implementation of a Industry-Oriented Data Exchange Middleware System EasySwitch

YIN Qing LIU Wei-Ning

(College of Computer Science and Engineering, Chongqing University, Chongqing 400044)

Abstract To solve the problem of data exchanging and application integration among heterogeneous enterprise application systems this paper introduces a industry-oriented data exchange middleware system (EasySwitch) based on SOAP and XML-RPC. This system has characteristic of both reconstructed data exchange core and industry-oriented application which constructs a data exchange and interoperation platform for enterprise application systems. The basic data exchange theory is analyzed in this paper followed by system architecture of EasySwitch, designing of application integration function and key realization techniques of each component.

Keywords Data exchange, SOAP, XML-RPC, Application integration, Web service

1 引言

随着企业信息化的不断深入,各个企业基本上建立了适应本企业业务需求的信息系统。这些系统都是建立在私有数据格式和处理流程基础之上的。在当今电子商务的交互模式下,两个或多个应用系统需要进行业务数据的交流,并根据不同的业务数据触发内部的处理流程,协同完成商务活动。

这种多个异构系统基于数据交换的应用协作所遇到的困难主要体现在:第一,从应用层面上讲,系统所能认识处理的数据是私有的,格式是固定的,它们对其它系统的应用数据是不能直接认识和处理的。第二,系统的基础运行平台的异构性,包括操作系统不同,开发语言不同,所支持的对象访问机制和通信协议不同等,造成了系统之间不能直接互访。第三,当多个应用系统有交叉数据交流需求时,每一个交流实体会面对 $N-1$ 种集成关系。如果采用分散的点对点集成模式,其开发和维护都是很困难的。

本文根据行业内交流数据的相似性和相关性基本理论,通过对数据模式的完全解析构造出可重构的数据交换核心,并在此基础上开发了应用集成框架,从而完整实现一个基于 SOAP/XML-RPC 机制的数据交换中间件系统(EasySwitch)。该系统具有可重构的数据交换核心和面向行业应用的特点,集中解决了企业应用系统进行数据交流时存在的数据格式差异性,为应用系统协作提供了一个实用的、快速的数据交换平台。文章在分析了数据交换的基本原理

与思想后,着重阐述了 EasySwitch 的系统架构、应用集成功能设计和各部分核心实现技术。

2 关键技术特征

对于进行数据交换的企业应用系统来说,虽然它们的私有数据格式存在差异,但它们都是对同一行业内的业务数据的描述,因此具有数据格式的相似性和内容的相关性。这种不同企业间数据的相似性与相关性为实施数据格式与内容的转换提供了可能性,这是我们在构建 EasySwitch 系统时的一个基本的理论依据。行业数据标准是同一行业内部对有关业务数据描述达成的共识,它的设立使各企业私有数据之间有了一个共同的参照系,这样使得原来各企业数据分散的点对点的对应关系简化为各企业私有数据与同一个行业数据标准之间的对应。这一点正是 EasySwitch 数据交换核心的设计思想,也是 EasySwitch 面向行业应用的特点所在。

2.1 数据交换原理

EasySwitch 的数据交换是基于行业数据标准,以及各企业私有数据格式与之预定义的映射关系来完成的。图1说明了交换原理,阐述如下。

1) XML 作为一种高度结构化、层次化、扩展性的语言很适合用来描述不同应用、不同领域的的数据。企业应用系统把 XML 作为基础数据处理格式为数据交换创造了条件。各个企业的应用系统采用了不同的 XML 标签集,这些标签集虽然

^{*} 国家十五电子商务与现代物流攻关项目(2001BA205A12)。重庆市科技攻关项目。殷庆 硕士研究生,主要研究方向为分布式计算。刘卫宁 副教授,硕士生导师,主要研究方向为电子商务,智能交通,计算机网络。

具有不同的层次结构和名称,但它们都是对行业内业务数据的描述,因此具有结构和内容的相似性。

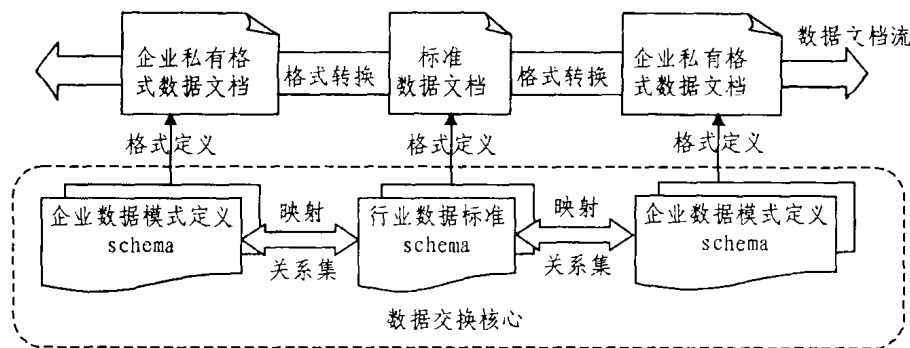


图1 数据交换原理图

2)数据文档的结构用数据模式(Schema)来定义,schema定义了相应数据文档的层次结构,以及每一数据节点的类型、名称等信息。通过对 Schema 的解析形成节点集,保存了相应数据文档的一切特征信息,可以实现对数据文档的完全控制。

3)EasySwitch 平台的接入用户在完成初始化任务时建立其私有数据模式同行业数据标准之间的节点级映射关系,包括节点编码的对应关系和节点数值运算函数关系。这些映射关系集也是用 XML 来描述的,可以随时进行局部的修改和调整。

4)根据已建立的各企业私有数据模式与行业数据标准之间的映射关系,EasySwitch 平台的接入企业之间可实现自由数据交换,形成连续的数据文档流(关于数据模式的节点解析,映射关系的函数表达以及数据文档基于模式的节点路径分析、存储和恢复技术将另文介绍)。

EasySwitch 的数据交换核心是由行业数据标准模式、各接入企业的私有数据模式以及相应的映射关系集共同构成。由于对 schema 实施了节点级的分解,映射关系也是基于模式节点来定义的,因此行业数据标准是可定制的,整个数据交换核心也表现出可重构的特性。这样在系统初始化时通过定义特定的行业数据标准,可使 EasySwitch 适用于不同的应用环

境和行业,表现出极强的可移植性。

2.2 应用集成框架

EasySwitch 的可重构的数据交换核心提供了面向行业的数据交换机制,在此基础上还必须提供应用集成的框架以实现对上层企业应用系统的支持,两者共同实现基于数据交换的行业应用协作平台。具体来说 EasySwitch 提供了模式和映射管理服务帮助平台接入用户完成接入初始化任务以及数据交换核心的创建和存储工作;提供了数据封装和通信服务完成各种系统消息的创建、发送和接收工作;提供了接入代理机制,完全屏蔽了 EasySwitch 内部复杂结构,为用户提供一个单一的、通用的、可定制接口层。

3 EasySwitch 总体结构

系统总体结构如图2所示,系统逻辑上由中心交换系统(MC)和客户端代理(UA)两部分构成。MC 是系统核心,包括格式转换服务、数据模式管理、成员管理等核心服务组件。它们是以封装了处理逻辑的 Servlet 形式存在,部署于应用服务器中;UA 是中间件系统在客户端的一个延伸,主要为客户应用提供接入服务,完成解析和封装数据消息以及和 MC 通信等任务。

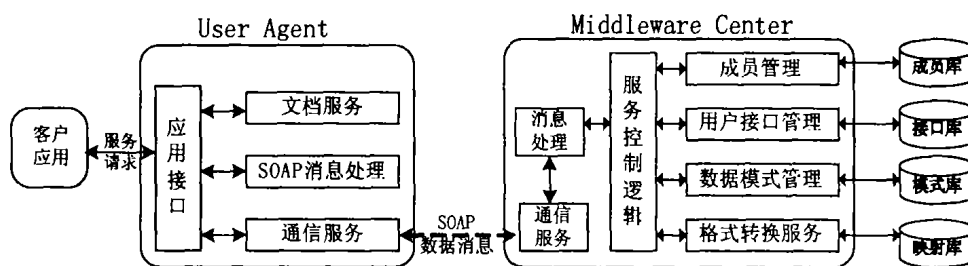


图2 系统总体模块结构图

现以数据交换为例说明系统的工作过程。客户应用系统发出数据交换服务请求消息到接口层,UA 解析此消息获取其交换数据部分。UA 创建系统标准 SOAP 数据交换消息,并添加身份验证标识和服务类型信息到消息头,通过通信服务将此 SOAP 消息发送至 MC。中心系统接到数据消息后进行如图3的处理过程。

1)数据交换服务器(DataExchangeSvr)调用接收服务器(DataReceiverSvr)完成数据消息的接收并对消息进行合法性检验,包括消息发送者身份验证、数据模式 schema 验证、schema 对应的映射关系验证、消息接收者验证。

2)若此次数据消息完全合法,则发送 MSG_OK 类型系统消息回 UA,继续流程;否则发送 MSG_ERR 系统消息回 UA,拒绝服务。

3)DataExchangeSvr 获取消息类型码,调用数据转换服务器(DataTransformSvr)执行格式转换服务。DataTransformSvr 根据源数据消息包含的 schema 和接受者信息,从映射库中读取相应规则完成数据转换,并创建目的数据消息队列返回给 DataExchangeSvr。

4)DataExchangeSvr 接着调用数据发送服务器(DataSendSvr)执行数据发送任务。DataSendSvr 根据每个目

的消息中包含的接收者 ID,从接口库中读取接收者 UA 信息,执行数据发送任务,如果发送失败则将本次数据消息入库保存,等待以后用户接收。

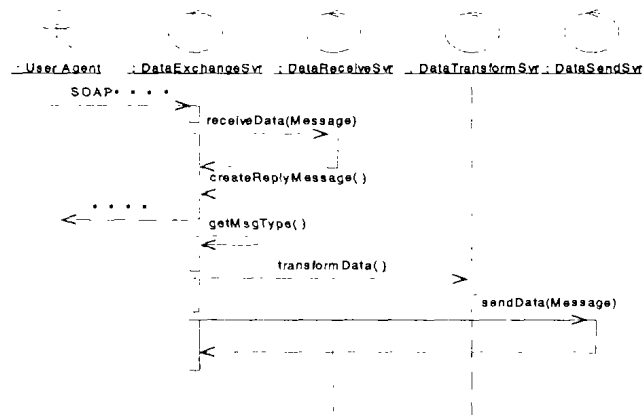


图3 MC 对数据交换消息处理流程

4 通信基础架构

由于 EasySwitch 是一个基于 Internet 的分布式系统,在数据交换核心之上构建应用集成框架时,一个重要的问题就是系统内各部分的数据通信机制的设计。

4.1 SOAP 数据消息设计和实现

简单对象访问协议(SOAP)基于 XML 格式,现已成为标准的结构化数据封装协议。EasySwitch 采用 SOAP 来封装系

统的交换数据和其它系统控制信息。〈Header〉部分装载系统控制信息,如表1所示。

表1

标签名	含义
〈identifier〉	用户的标识符
〈identity-code〉	用户的验证码
〈message-type〉	消息类型值,在类 MessageType 中编码。
〈schema-id〉	Body 部分的数据文档所使用的模式的 Id。此 Id 值全局唯一。
〈transaction-id〉	本次数据交换的事务号。
〈reference-id〉	本次数据交换的参考号。
〈sender〉	消息发送者的 Id。
〈receiver〉	消息接受者的 Id。

〈Body〉部分填入实际欲交换的 XML 数据。同时 AttachmentPart 部分可以携带非 XML 数据比如其它形式的文本、图形数据等,这样就极大地扩展了系统可处理的数据类型。它们是系统识别和处理消息的基础,而且通过增减标签可以很方便地扩充系统的功能。

项目中我们采用了 Java 中的 JAXM 作为 SOAP 实现标准,基于 Sun 公司的 JWS DP1.1 软件包来开发。MC 是根据消息的不同类型调用不同的服务组件进行处理,EasySwitch 包括的消息类型如表2所示,这些值在类 middlew-are. common. MessageType 中进行编码,供其它类使用。

表2

消息类型名称	描述	功能说明
MSG-SUBSCHEMA	提交模式消息	消息体包含数据模式文档。用户提交数据模式,以及对模式管理时使用本消息。
MSG-REQSCHEMA	请求模式消息	本消息包含对某一数据模式的请求信息。当用户管理自己的数据模式及建立映射时使用。
MSG-SUBMAPPER	提交映射消息	消息体包含某一映射关系文档。当用户建立映射关系,以及管理映射关系时使用。
MSG-REQMAPPER	请求映射消息	本消息包含对某一映射关系的请求信息。当用户管理自己的映射关系时使用本类型消息。
MSG-SENDDATA	发送数据消息	消息体包含欲发送的数据文档。用户进行数据交换时使用本类型消息。
MSG-RECEIVEDATA	接收数据消息	当用户主动向系统接收发给本方的数据时使用本类型消息。
MSG-OK	成功接收消息	应答消息,说明成功接收前一消息。
MSG-ERR	错误消息	应答消息,说明前一消息含有错误。

4.2 消息处理响应模式

MC 包含了较多的服务模块,针对不同的消息类型采用不同的消息处理模式,设计如下:

1)对于用户接入初始化时,在建立本企业的数据 schema 和建立行业数据标准映射关系中,UA 提交的 MSG-SUBSCHEMA、MSG-REQSCHEMA、MSG-SUBMAPPER、MSG-REQMAPPER 消息,MC 是采用同步处理的方式。即收到消息后立即处理,将处理结果返回给 UA,在此期间用户必须等待 MC 完成服务请求。

2)在数据交换过程中,交换各方是处于一种松耦合状态,不强求交换各方同时在线,因此 MC 对于数据交换消息 MSG-SENDDATA、MSG-RECEIVEDATA 采用异步处理模式。对通过合法性验证的数据消息,立即返回应答消息 MSG-OK 给 UA 后,才进行后续处理。UA 不必等待整个交换过程完成即可接受用户的其它服务请求。这样可以最大限度地降低系统的耦合度,减少发生死锁的几率。

3)对于某些有同步需求的数据交换过程,系统通过 〈transaction-id〉和〈reference-id〉来实现。比如企业 A 向企业 B 发送了一次 transaction-id 为 0001 的合同数据,企业 B 在处

理完这份合同数据后应发送一次 reference-id 为 0001 的合同确认消息,企业 A 可根据 reference-id 来判断这是对 0001 合同的确认数据,不至于出错。

4.3 承载通信协议设计

SOAP 规范并未指定采用何种协议作为 SOAP 消息的传输协议,SOAP 可以用 HTTP、SMTP 或 TCP 流机制进行传送。当前的各种 SOAP 实现绝大多数采用 HTTP 协议作为 SOAP 的传送协议,主要是因为 HTTP 作为超文本传输协议也可以适用于传送 XML 形式的 SOAP 数据。EasySwitch 采用的 JAXM 软件包也是基于 HTTP 实现的,具有以下优点:

1)HTTP 是一个广泛使用的公共协议,它不局限于某一厂商的实现平台,这对于实现本中间件系统的跨平台性和互操作性是至关重要的。

2)大多数防火墙配置为仅允许 HTTP 的连接,这样应用系统可以在不失去安全性的前提下,实现数据交换服务。

3)采用 HTTP 协议对于服务器端的开发也是比较方便的,MC 就是运行于 Web 服务器 Tomcat 中,加入对 SOAP 消息的产生和处理功能,就成为能够处理 SOAP 消息的 SOAP Server。

5 UA 设计

UA 驻留于客户端, 主要为用户应用系统提供接入服务, 并且作为接收方还要随时接收从 MC 发送的其它用户的交换数据。UA 的设计要考虑跨平台和可移植的要求, 不能受到用户应用系统运行平台的限制。EasySwitch 采用 XML-RPC 作为接口层基本的服务调用方式, 最大限度地降低了数据中间件的 API 层的访问难度, 同时保证了平台独立性要求。

5.1 UA 和用户应用系统的交互模式

UA 屏蔽了 EasySwitch 内部的复杂结构和运行过程, 向企业应用系统展现出一个单一的服务接口。同时, UA 还能适应用户环境的变化, 比如当用户 IP 地址发生变化时, UA 能够即时发送更新信息到 MC。UA 和用户应用系统之间的交换模式如图 4 所示。

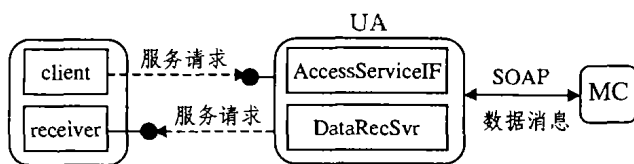


图4 UA 和用户应用系统交互模式示意图

1) AccessServiceIF 是 UA 的接入服务接口, 它为用户应用系统中的 client 方提供包括数据交换在内的所有服务调用。

2) DataRecSvr 是 UA 中的数据接收服务器, 它随时准备接收从 MC 发送的其它企业的交换数据消息。

3) receiver 是用户应用系统提供的数据接收服务接口, 用户必须在安装 UA 时定义 receiver 的位置信息。当 DataRecSvr 收到数据消息后, 从配置文件中读取 receiver 的信息, 然后通过动态调用方式生成 XML-RPC 服务请求并发送到 receiver 中, 完成一次数据交换过程。

5.2 基于 XML-RPC 服务接口的实现

UA 中的 AccessServiceIF 和用户应用系统提供的 receiver 都是基于 XML-RPC 方式定义的服务接口, 它通过 WSDL 文档来描述其服务的名称及参数类型, 调用指令和返回数据都是以标准的 SOAP 封装传送的。UA 方的 AccessServiceIF 是采用 JAX-RPC 来开发的, JAX-RPC 提供了基于 SOAP 的 RPC 调用运行环境, 其核心是实现了标准

XMLSchema 数据类型和 Java 数据类型之间的映射。AccessServiceIF 的 Java 定义如下:

```
public interface AccessServiceIF extends Remote {
    public abstract Source sendData(Source sourceData) throws RemoteException;
    public abstract Source receiveData(String memberId) throws RemoteException;
    public abstract SchemaInfo [] getSchemaInfo (String memberId) throws RemoteException;
    public abstract ErrorInfo [] getErrorInfo () throws RemoteException;
}
```

总结 为适应各企业应用系统间进行数据交换协作的需求, 解决异构系统集成困难, 本文基于行业内交流数据的相似性和相关性理论, 通过行业数据标准的设立和数据模式的节点分解构造出可重构的数据交换核心, 并在此基础上通过采用 SOAP, XML-RPC 等标准协议规范开发了应用集成框架, 从而完整地实现了一个异构系统基于数据交换的协作平台 EasySwitch。通过对数据交换核心的重构可使 EasySwitch 适用于不同领域, 比如证券系统间交易数据的交换, 电子政务系统间的公文流转等。本文所实现的 EasySwitch 系统现已初步应用于重庆市摩托车电子商务平台, 是一个典型的行业内企业数据交换的应用。需要指出的是, EasySwitch 不仅可用于行业内相关企业进行数据交换集成, 对于企业内部各个子系统进行数据协作也能很好地支持。

根据企业实际的业务流程框架及信息技术的发展状况可以把企业集成分为数据集成和服务集成。基于数据处理的商务过程, 对于自主运行的应用系统来说, 在收到某一类型的业务数据后能够触发内部处理流程, EasySwitch 对于这种基于数据交换的应用协作提供了完整的解决方案。如何扩展 EasySwitch 的应用集成框架, 实现对企业间基于功能服务的集成是下一步研究的内容。

参考文献

- 1 Gabrick K A, Weiss D B. J2EE and XML Development, Manning Publication, 2002
- 2 SOAP Version 1.2 Specification. <http://www.w3.org/TR/2003/PR-soap12-part1-20030507/>. <http://www.w3.org/TR/2003/PR-soap12-part2-20030507/>
- 3 JAXRPC-1.0 Specification, Sun Microsystems, 2002
- 4 SAAJ1.1 Specification, Sun Microsystems, 2002
- 5 Myerson J M. The Complete Book of Middleware. Auerbach Publications, 2002

(上接第 155 页)

3.4 性能影响分析

如果将 ACL 信息集中存放在文件中, 要访问 ACL, 在文件最近未被访问过的前提下, 需要至少两次磁盘访问, 而通过文件系统的扩展属性机制存放 ACL, 利用文件系统 inode 的缓冲机制, 可以减少访问磁盘的次数。且扩展属性块可以被多次引用, 提高了效率。

结束语 本文探讨了安全操作系统 SECIMOS 中基于访问控制表的自主访问控制机制的设计与实现的主要思想, 并讨论了 ACL 如何与传统 linux 的文件权限保护模式在系统中协调工作的问题。本文所提出的 DAC 机制不仅对授权主体细化到单个用户, 并且对权限和系统可操作的客体类型也加以细化, 针对不同类型的客体给出客体的权限范围, 完善了访问权限检查算法。提出了权限屏蔽值和 Default 客体的概念, 使我们的安全操作系统 SECIMOS 的 DAC 机制能够灵活

地支持实际应用中的 DAC 要求。

参考文献

- 1 中国国家标准-计算机信息系统安全保护等级划分准则 GB17859-1999
- 2 石文昌, 孙玉芳. 安全操作系统研究的发展(上). 计算机科学, 2002, (29)6
- 3 IEEE Draft P1003. 1e/2c IEEE Standard Department, 1997
- 4 Grunbacher A. Posix Access Control Lists on Linux, SuSE Labs, linux AG
- 5 红旗安全操作系统技术白皮书. <http://www.redflag-linux.com/security/document/whitepaper.pdf>
- 6 Gildfind A. Access Control lists and Extended attributes on linux, SGI
- 7 Final evaluation report. Trusted Xenix v3.0, National Computer Security center
- 8 Morris J, et al. Linux Security Modules: General Security Support for the Linux Kernel. <http://www.intercode.com.au/jamesm/lsm-usenix-html/lsm-html.html>, 2002