

基于 Web Services 企业应用集成的设计与分析

王彦丽¹, 陈明², 陈峰², 骆力明¹

(1. 首都师范大学 信息工程学院, 北京 100037;

2. 中国石油大学 计算机科学与技术系, 北京 102200)

摘要:企业应用集成(EAI)是整合企业内外不同应用系统间信息和业务流程的解决方案。Web Services 是一种面向服务的技术架构,其突出优点是实现真正意义上的平台独立性和编程语言独立性。参考已有的企业应用集成产品,将它们原有自行设计的跨平台的数据传输、变换及解释模块和跨应用的业务流程模块用 Web Services 架构中的 SOAP, WSDL, UDDI 和 WSFL 等技术替换而形成一个新的企业应用集成模型,并研究了该模型的工作原理及所面临的挑战。

关键词:EAI; Web Services; SOAP; WSDL; UDDI; WSFL

中图分类号: TP399

文献标识码: A

文章编号: 1673-629X(2008)09-0212-04

Design and Implementation of Enterprise Application Integration Based on Web Services

WANG Yan-li¹, CHEN Ming², CHEN Feng², LUO Li-ming¹

(1. Information Engineering College, Capital Normal University, Beijing 100037, China;

2. Department of Computer Science & Technology, University of Petroleum, Beijing 102200, China)

Abstract: Enterprise application integration(EAI) is the resolution scheme integrating the information and the business flow of the different application systems outside and inside the enterprise. Web services is a service-oriented technical architecture whose main significance is really platform-independent and programming language-independent. Refers to the EAI already put into use and replaces their originally self-devised trans-platform data transmission, transform, explanation module and the trans-application platform-independent business flow module with SOAP, WSDL, UDDI and WSFL of Web Services to form a new EAI model. Moreover, studies the working principle of this model and the challenge this model has to be faced.

Key words: EAI; Web services; SOAP; WSDL; UDDI; WSFL

0 引言

企业应用集成(Enterprise Application Integration, EAI)^[1]通过协议转换与数据传输服务实现企业内部不同应用系统(如企业资源计划(ERP)、客户关系管理(CRM)、行销自动化(SFA)、供应链管理(SCM)等)之间的信息与指令的安全而有效地传输,它是支撑企业内部不同应用系统间业务流程的关键。在传统的企业应用集成时,不仅要分析各个特定的系统,而且还要为集成各个特定的系统定义一套交互平台来串接这些系统,而交互平台可能又是各厂商定制的,缺乏通用性和复用性,这会造成很大的复杂性和高昂的成本。

XML 技术的发展和 Web Services 的诞生使企业

应用集成出现了新的思路。Web Services 是由 URI 定义的软件系统,它使用 XML 定义和描述其公共接口元素(例如应用程序逻辑),以标准的 Internet 协议(如 HTTP、FTP 等)作为通信基础,其他软件系统可以通过由 Internet 协议传递的基于 XML 的消息机制来发现 Web Services 并与之交互^[2]。这使得基于 Web Services 的应用程序成为松散耦合、面向组件和跨技术实现。需要集成的各特定的应用系统可利用 Web Services 机制作为通用的交互平台:只需将要集成的功能按 Web Services 的描述方式定义并实现访问自己的接口,然后将自己注册到 Web Services 的注册中心,其他的信息系统就可以通过注册中心查找到访问自己的接口及使用方式,就可以调用自己的功能了。

Web Services 不仅使企业应用集成的成本大大降低,而且使企业的资源在 Internet 范围内被更大程度更大范围地使用。文中将 Web Services 与企业应用集

收稿日期:2007-12-15

基金项目:国家自然科学基金资助项目(60072006)

作者简介:王彦丽(1968-),女,讲师,研究方向为计算机网络、软件工程。

成结合起来,提出了一种基于 Web Services 的企业应用集成模型。

1 Web Services 体系结构

Web Services^[3]是一种面向服务的体系结构,其中定义了一组标准协议,用于接口定义、方法调用、基于 Internet 的构件注册以及各种应用的实现。整个 Web Services 的技术系列被称为“Web Services Stack”,按照堆栈的方式共存,如图 1 所示。

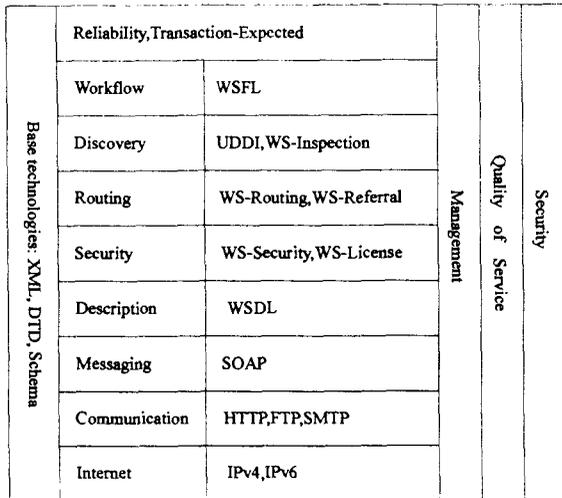


图 1 Web Services Stack

底部是已经定义好的并且广泛使用的协议标准: IP、HTTP、FTP 等。中间是目前开发的 Web Services 的相关标准协议,包括服务调用协议 SOAP、服务描述协议 WSDL 和服务注册发现协议 UDDI、WS-Inspection 以及服务 workflow 描述语言 WSFL、Web Services 的安全协议以及路由协议等。右边部分是各个协议层的公用机制,这些机制一般由外部的正交机制来完成。

SOAP(Simple Object Access Protocol)是 Web Services 通信的主要协议。SOAP 本身并不定义任何应用语义,如编程模型或特定语义实现,它通过一个模块化的包装模型和对模块中特定格式编码的数据的重编码机制来表示应用语义。它为在一个松散的、分布的环境中使用 XML 对等地交换结构化的和类型化的信息提供了一个简单且轻量级的机制。

WSDL (Web Services De-

scription Language)是 Web 服务的描述语言,它是一种基于 IDL(Interface Definition Language)技术的描述语言。WSDL 定义了一套基于 XML 的语法,将 Web 服务描述为能够进行消息交换的服务访问点的集合,从而满足了这种需求。WSDL 服务定义为分布式系统提供了可机器识别的 SDK 文档,并且可用于描述自动执行应用程序通信中所涉及的细节。

UDDI(Universal Description, Discovery and Integration)规范定义了 Web 服务的发布与发现的方法。UDDI 的核心组件是 UDDI 商业注册,它使用一个 XML 文档来描述企业及其提供的 Web 服务。

WSFL(Web Services Flow Language)使商业流程和利用了大量 Web 服务的交易生命周期能够进行无缝连接,而不必考虑其编程语言和运行环境。

WS-Security 和 WS-License 都是 SOAP 规范的一个扩展,它们组合起来一起为通过 SOAP 进行交互的 Web 服务提供了扩展的安全机制。

WS-Routing 定义了路由 SOAP 消息的机制,它通过定义一个方法来说明一个预先设计好的路由或传输路径。WS-Referral 用来配置用于转发消息的 SOAP 节点(SOAP 路由器)中关于消息路径(路由条目)的指令。

2 基于 Web Services 的企业应用集成模型

文中将 Web Services 技术应用到企业应用集成中,提出了一个基于 Web Services 的企业应用集成模型^[4,5]。

2.1 基于 Web Services 的企业应用集成模型

图 2 所示的是一个基于 Web Services 的企业应用

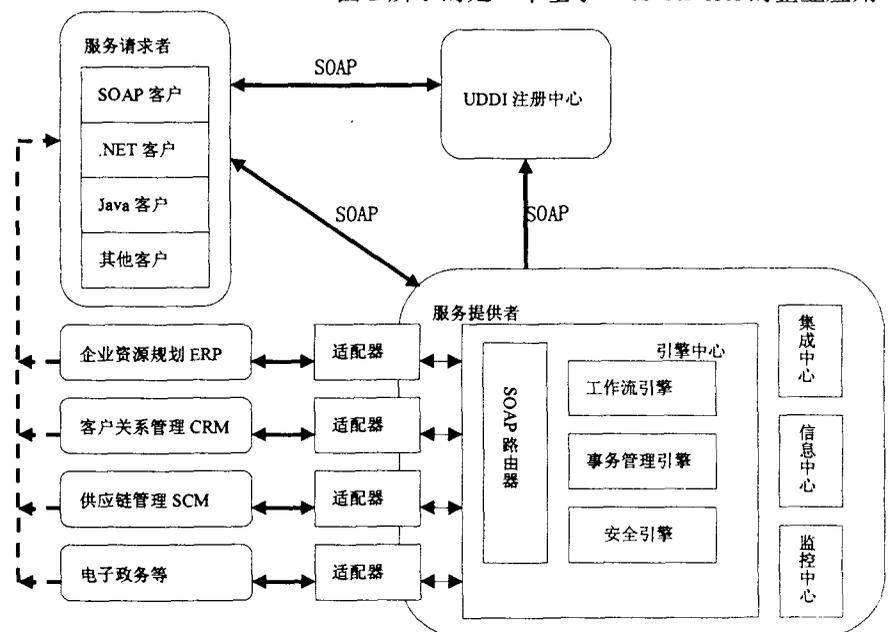


图 2 一个基于 Web Services 的企业应用集成模型

集成模型。图中 Web Services 架构的三大部分是:服务请求者、UDDI 注册中心和服务提供者。图中其他部分为当前企业中比较典型的几类系统 ERP、CRM、SCM 和电子政务等^[6,7]。

(1) 服务请求者。

其中包含 SOAP 客户以及流行的基于 .NET 平台或 Java 平台的客户或其它 Web 服务类型的客户。

(2) UDDI 注册中心。

其中包括企业内部使用的私有注册中心和在 Internet 上公布的公有的注册中心。这两类注册中心的使用情况要依赖于企业对其集成的方向和范围的要求来定。考虑安全和集成的成熟度要求可以先在企业内部形成私有的注册中心,随着成熟度的增加或 B2B 集成的需要可以转向公有的注册中心。可以让不同的服务根据不同的需要在不同的注册中心上注册,两类注册中心可以方便地形成映射关系。

(3) 服务提供者。

服务提供者包含四个部分:集成中心、信息中心、监控中心和引擎中心。

① 集成中心。

实现应用集成服务的配置和管理及企业应用解决方案的动态配置,包括:消息与队列功能、工作流管理、应用资源管理服务、统一界面服务等。

② 信息中心。

实现信息的安全生命周期管理和维护,包括共享信息管理、数据操作管理和共享模型管理等。

③ 监控中心。

实现对服务的运行进行管理和监控,包括服务对象管理器、动态监控及报警管理等。

④ 引擎中心。

引擎中心是服务的中心,它提供三类引擎:工作流引擎、事务管理引擎和安全引擎。

* 工作流引擎:使商业流程和利用了大量 Web 服务的交易生命周期能够进行同样的无缝集成和衔接。这种衔接包括横向的企业间的连接或企业内部的平级连接和企业内部上下级间纵向的连接。

* 事务管理引擎:确定事务协调者和参与者,由协调者控制整个事务的提交和失败后的事务回滚。

* 安全引擎:负责对服务中的传输数据进行加密和提供认证服务以实现数据在网络上的安全。

SOAP 路由器是实现客户调用 Web Services 的关键部件,以实现 SOAP 消息的传递。

(4) 适配器。

适配器是集成引擎的核心,由以下几部分组成:

① 接口:针对不同的应用系统,适配器提供不同的

接口。这是应用系统可以调用 Web Services 集成平台的惟一途径。

② 连接控制器:用于建立安全的与后端服务器的通信连接,包括用户身份鉴别、授权等。

③ 数据转换器:用于验证数据有效性、生成 WSDL,实现在 SOAP 数据格式与应用系统数据格式之间的转换。

④ 消息路由器:实现在 SOAP 路由器与适配器之间的消息传递,将 SOAP 消息过滤后路由到正确的目的地。

(5) 封装成 Web Services 组件。

在该集成框架中,应用系统既可以是已有的应用,也可以是新开发的 Web Services 应用。如果是已有的应用系统,需要首先将它封装成 Web Services 组件,方法是:生成描述该系统功能和调用方法的 WSDL 文件,然后生成服务器端基于 SOAP 的服务框架,并在此基础上开发适用于已有系统的适配器,最后将服务描述文件通过 UDDI API 发布到 UDDI 注册服务器中。因此,被封装成 Web Services 组件的已有应用系统中的服务,在通过 SOAP 查找 UDDI 注册中心或绑定企业内或外的服务时,它本身就成为了一种服务请求者,如图 2 中虚线所示。

2.2 客户调用 Web Services 的过程

(1) 客户用 WSDL 描述需要访问的服务,用 SOAP 消息向注册中心发出查询请求。

(2) 注册中心将该方法的 WSDL 描述返回客户。

(3) 客户用得到的 WSDL 描述生成 SOAP 请求消息,绑定服务提供者。

(4) SOAP 请求被作为一条 HTTP POST 请求发出,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,然后将请求消息传递到指定的 SOAP 路由器。

(5) 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。

(6) 适配器激活应用。

(7) 应用系统处理请求,并将结果返回给适配器。

(8) 适配器将得到的结果打包成 SOAP 消息,返回给 SOAP 路由器。

(9) SOAP 消息再返回到 Web 服务器。

(10) 客户最终得到包含执行结果的 SOAP 消息。

2.3 基于 Web Services 的企业应用集成的优势和面临的挑战

2.3.1 基于 Web Services 的企业应用集成的优势

与传统的集成方式相比较,基于 Web Services

的企业应用集成具有以下优势:

(1) Web Services 是由一系列标准组成的,因此 Web Services 集成各种应用的方法是标准的,具有较好的通用性和兼容性。

(2) 面向对象和 XML 等相关技术的采用,使得 Web Services 具有更好的跨平台性,可以更好地满足分布式集成的要求。

(3) Web 服务的集成结果是一种松耦合的集成模式,它通过建立涵盖服务通信(SOAP)、服务描述(WSDL)和服务发现(UDDI)等标准实现应用集成的框架。

(4) 由于和 Web 技术的结合,使得 Web Services 对 Web 应用具有很好的协调性。

(5) Web Services 能够更好地满足企业以客户为中心的协同工作。

2.3.2 基于 Web Services 的企业应用集成的不足

就目前来看,基于 Web Services 的企业应用集成方式还不能完全替代当前现有的一些传统的方式。这不仅是因为传统的一些集成方式如数据集成的某些方面 Web Services 集成方式还无法触及,还由于 Web Services 是一种新的技术,还处于不断的发展和完善之中,所以从技术角度看,尽管有些标准已经在制定过程中,但 Web Services 集成方式还存在如下不足:

(1) 由于 Web Services 目前的实施基础还不十分完善(如缺少配套的工具等),在对遗留数据信息进行包装时需要大量的编程工作,建立 Web Services 也需要投入大量的工作,所以 Web Services 的集成成本是比较高的。

(2) 目前,Web Services 是位于防火墙之内的。为了突破防火墙而与合作伙伴和客户建立联系,Web Services 集成方式必须解决相关的加密、认证、签名等技术。

(3) Web Services 还没有解决事务处理、工作流的编排与描述等问题,缺乏业务流程的管理能力。

(4) Web Services 缺乏与数据库有效的通信能力,而现在企业的大部分数据仍然存储在数据库系统之中,并且这种情况还将持续下去。

(5) 由于 Web Services 集成方式是一种松耦合的集成模式,集成结果的性能相对较低,无法满足一些系统的集成要求,例如实时系统的集成。

2.3.3 基于 Web Services 的企业应用集成与传统的企业应用集成的结合

传统的企业应用集成方式是一种专用的解决方案,形成紧耦合的集成结果,需要更多的开发和维护工作。而 Web Services 集成方式是一种标准的解决方案,形成松耦合的集成结果,但性能相对较低。传统集成方式和 Web Services 集成方式可以说各有优劣。那么如何将二者进行结合更合适呢?

事实上,Web Services 完全可以作为传统集成方式所要组织的技术方法之一。Tech Metrix Research 就提出将 Web Services 应用在其传统集成方式解决方案中的接口部分^[7]。实际上,通过在传统集成方式中引用 Web Services 的相关标准(SOAP/HTTP/WSDL 等)和技术,就可以实现前面提到的面向服务层的企业应用集成。面向服务层的企业应用集成是一种松耦合的应用集成模式,该集成模式除了具有更好的跨平台性和可扩展性外,也可以得到较高的性能。

3 结束语

文中研究和分析了企业应用集成和 Web Services,并提出了一个基于 Web Services 的企业应用集成模型。对比传统的企业应用集成和基于 Web Services 的企业应用集成,论述了基于 Web Services 的企业应用集成的优势以及所面临的挑战。

参考文献:

- [1] 韦银星,张申生.企业应用集成技术研究[J].计算机集成制造系统,2002(8):593-596.
- [2] Austin D, Ferris C, Barbir A, et al. Web Services Architecture requirements[EB/OL]. 2002-08. <http://www.w3c.org/TR/wsa-reqs>.
- [3] 魏楚元. Web Services 体系结构与实现机制探讨[J]. 航空计算技术,2003(1):101-105.
- [4] Hammer K. Web Services and Enterprise Integration[J]. EAI Journal, 2001(3):12-15.
- [5] Holland P. Building Web Services From Existing Application [J]. EAI Journal, 2002(9):45-47.
- [6] 李锐. 企业应用 Web Services 技术的集成架构模式[J]. 价值工程, 2006(8):84-88.
- [7] 孙晋文,肖建国. 企业应用集成与基于 Web Services 的构架应用[J]. 计算机工程与应用, 2003(21):205-208.

(上接第 211 页)

[9] Nikkanen K. μ CLinux as an Embedded Solution[M]. Finland: Turku Polytechnic publishing house, 2003.

[10] Choi Hyok - Sung, Yun Hee - Chul. Context Switching and

IPC Performance Comparison between μ CLinux and Linux on the ARM9 based Processor[R]. Korea: SAMSUNG Tech, 2005.