

ESB 中的数据交换技术^①

Data Exchange in ESB

刘迎春 兰雨晴 于乐乐 (北京航空航天大学软件工程研究所 100083)

摘要:本文描述了目前的技术焦点之一——SOA 和 ESB 技术,并集中讨论了数据交换技术在 ESB 中的重要性和实现方式。文章详细讨论了构建企业级数据交换应用时所采用的应用模式,给出了基于该应用模式用户所需要进行的工作。

关键词:SOA ESB 数据交换

1 概述

随着基于统一标准的集成技术的不断发展,面向服务的体系结构(SOA)和企业服务总线技术(ESB)已经成为目前最热门的 IT 技术之一。对于 ESB 来说,其最核心的功能,还是数据的交换。没有有效的数据交换手段,ESB 也就只是空谈。本文就将结合现有的各种软件产品,设计一个完整的企业级数据交换应用模式,开发人员可以借用此模式,采用一些现成的成品,迅速搭建数据交换系统,减轻开发的负担。



图 1 面向服务的体系结构

2 SOA、ESB 和数据交换技术

2.1 面向服务的体系结构

面向服务的体系结构(SOA, Service Oriented Architecture)是一种新的设计方法学,其目的是最大限度地重用应用程序中立型的服务以提高 IT 适应性和效

率。虽然这些概念已经存在了数十年之久,但只是在出现了基于统一标准的集成技术(如 Web 服务和 XML)之后,SOA 才开始被加速采用。SOA 的结构如图 1 所示。

2.2 企业服务总线

企业服务总线技术(ESB, Enterprise Service Bus),则是在 SOA 的基础上提出的构建基于面向服务体系结构(SOA)解决方案时企业所使用基础架构的关键部分。简而言之,ESB 提供了连接企业内部及跨企业间新的和现有软件应用程序的功能,以一组丰富的功能启用管理和监控应用程序之间的交互。

从本质上来说,ESB 就是一种可以提供可靠的、有保证的消息技术的新型中间件技术。ESB 中间件产品利用的是 Web 服务标准和与公认的可靠消息 MOM(面向消息的中间件, Message - Oriented Middleware/MOM)协议接口。ESB 产品的共有特性包括:连接异构的 MOM、利用 Web 服务描述语言接口封装 MOM 协议,以及在 MOM 传输层上传送简单对象应用协议(SOAP)传输流的能力。

ESB 的出现改变了传统的软件架构,可以提供比传统中间件产品更为廉价的解决方案,同时它还可以消除不同应用之间的技术差异,让不同的应用服务器协调运作,实现了不同服务之间的通信与整合。从功能上看,ESB 提供了事件驱动和文档导向的处理模式,以及分布式的运行管理机制,它支持基于内容的路由

^① 本论文得到了上海中标软件有限公司承担的国家 863 项目“桌面操作系统及其配套环境,编号 2002AA1Z2101,2004AA1Z2020”的资助

和过滤,具备了复杂数据的传输能力,并可以提供一系列的标准接口。

2.3 数据交换技术

数据交换技术,顾名思义,就是在不同的系统之间交换数据的技术,它是目前流行的面向服务体系结构(SOA)和企业服务总线技术(ESB)的核心技术之一。

目前数据交换技术的主流,就是使用 XML 与 Web Service 技术进行的数据交换。通过这些技术,可以实现具有相当灵活性的、通用的数据交换接口,可以有效的解决不同应用系统、不同数据源之间的数据共享与交流问题。总体来看,数据交换的基本框架都是一致的,都是采用 XML 接口(或者叫适配器)来访问各种数据源,进行 XML 标准格式到不同异构数据之间的双向映射。翻译成 XML 标准格式的数据由数据交换引擎进行处理,并通过消息传输部件进行数据的交流。

2.4 面向服务体系结构、企业服务总线和数据交换技术

面向服务体系结构、企业服务总线和数据交换技术三者之间的关系如图 2 所示。

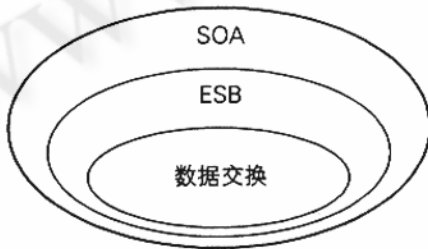


图 2 SOA, ESB 和数据交换

数据交换技术,是企业服务总线技术的最主要构成部件。数据交换产品主要侧重于数据在不同应用系统之间的交互过程,而企业服务总线技术更多的从整体上提出了未来的大型系统应采用的架构,例如面向服务的处理模式等。

3 数据交换技术的研究要点

基于上述的讨论,未来的 ESB 架构下的数据交换技术的研究要点将主要集中在以下几个方面:

3.1 XML 的存储

虽然 XML 可以采用各种不同的存储方式,但对企业级大型应用系统而言,关系型数据库是最普遍使用的存储方式。在关系数据库中物理地存储 XML 数据

有三种方法:碎片式(shredded)、结构化和非结构化。碎片式和非结构化存储虽然很有用,但还存在一定的局限性,而结构化方法可以充分利用关系数据和 XML 层次结构的能力。

碎片法把 XML 数据存成关系数据库中的列,但却丢掉了原始 XML 文档中数据之间的层次关系,如果你不需要继续保持数据的 XML 格式,那么碎片法就是非常有用的。让我们来看一个简单的例子,假设你有一个 Web 网站并允许客户在该网站上下订单,如果这些订单需要传送给多个数据库系统,那么创建一个 XML 文件并让这些数据库系统从该文件中获得数据——也就是说将 XML 文件打碎——是一种最有效和出错概率最小的方法,因为每个数据库系统获得的数据都来自网络上的同一个共享文件。

非结构化方法利用 LOB(大对象)数据类型,将整个 XML 文档当做一条单一的记录存储。多年来,数据库一直采用这种方法存储不同类型的文档,因此这种方法并不是什么新东西。虽然非结构化方法仅提供了有限的搜索能力,但它仍然是相当有用的,因为尽管你不能以它为基础进行查询,但原始数据的结构被保留下来了。非结构化 XML 存储的一大用途是保留原始文档以符合政府的规章。例如,如果一家金融机构收到了 XML 格式的原始贷款文档,它就可以从中获得每份贷款申请之间的关系并予以记录,而原始申请也保存在该记录中。

结构化方法允许你把 XML 数据存储于数据库中并保留数据的层次结构,因此,结构化存储也被称为“native XML(本地 XML)”存储,这也是每一家数据库厂商希望其产品达到的目标。保留 XML 数据层次关系带来的最大好处是可以接收 XML 文档,在关系数据库中组合或修改它,并生成新的 XML 数据。目前的主流数据库产品,都已经开始支持结构化的存储方法(例如 Oracle 9i/10g, DB2 UDB 8.1 等)。在这些产品中,用户可以直接对 XML 进行输入、创建、转换、解析、查询等工作。

3.2 XML 到其他数据源之间的转换

XML 与其他数据源之间的转换,可以分成几个方面来论述:

(1) XML 到关系型数据库的转换:关系型数据库是 XML 的最主要数据来源和目标。就目前的主流关

系型数据库来看,数据库本身已经提供了良好的 XML 接口,用户已经不必再自己开发相应的转换功能。例如,在 DB2 中,用户可以使用 SQL/XML 查询语言从关系数据直接创建 XML 文档,包括 XML 标签、属性,以及在创建文档的同时完成串联和聚合等操作。

(2) XML 到 XML 的转换: XSLT 技术很好的解决不同 XML 之间的转换问题,这一技术也已经在多种产品中得到了实现,例如 Oracle, Microsoft .NET 等等。用户可以直接使用这些产品提供的转换功能。

(3) XML 到其他异种数据源之间的转换:除关系型数据库和 XML 以外,还有其它的数据源,例如自定义格式的文件,非主流数据库系统等等。这一类数据源需要单独开发相应的到 XML 的转换功能。今后的 XML 与其他数据源之间的转换工作,将主要集中在这一类与应用相关的异种数据源上。

3.3 消息传递机制

严格来说,消息传递机制包括了两部分,协议和引擎。协议代表了在数据交换过程中数据传递的方式和使用的语言;引擎代表了实现数据交换的具体机制。在 XML 的传递中,目前已经出现了十多种消息传递协议,主流的 XML 传输协议如下:

(1) XML-RPC 和 SOAP。XML-RPC 提供了一个非常简单使用在 HTTP 上传递 XML 的 RPC 机制。SOAP 是基于 XML-RPC 实现的,是一个用来在非集中的分布式的环境中交换信息轻量级的协议。SOAP 是一个基于 XML 的协议,由三部分构成:一个信封,描述消息的内容和处理方式;一套编码规则,表示应用定义的数据类型实例;表示远程过程调用和响应的约定。SOAP 的主要问题如下:第一,SOAP 本身不解决高层的分布式对象问题,例如在多个用户之间传递消息时,SOAP 无法解决何时激活哪一个用户,以及何时注销哪一个用户;第二,SOAP 本身没有消息传递的语义,例如,SOAP 无法描述哪一条消息需要发送给哪一个用户;第三,SOAP 所基于的 RPC 机制可能会产生安全风险。

(2) WDDX。Web 分布式数据交换(Web Distributed Data Exchange/WDDX)提供了一个在 HTTP 之上交换复杂数据结构的机制。它的缺点是它不能被用来交换具有复杂关系的对象实例,也缺乏在多方向处理商务交易的高层语义。

(3) ebXML。由联合国贸易促进和电子商务团体

和结构化信息标准促进组织的共同倡议,电子商务 XML(Electronic Business XML/ebXML)的目标是“提供一个开放的基于 XML 的基架,使得各方以一个可互通、安全和一致的方法使用电子商务。”目前,ebXML 已获得所有重要的商业、服务业、贸易业和技术界有影响的实体的支持和积极参与。ebXML 支持在多方交易处理中必须的高层语义,包括一对一以及一对多路由模型,对多方回路文档交换的支持,以及根据消息头属性的服务质量确定等等。与其它开放的消息传递规范类似,ebXML 规范很可能保持对底层链接协议的无关。可用的协议包括 HTTP、FTP、SMTP 甚至 SOAP 等。

(4) JMS 规范。Java 消息服务(Java Message Service/JMS)规范代表了在基于标准的消息中间件领域的领先概念。JMS 包含了一个 API 以及一个提供了诸如持久、验证和事务语义的消息服务。JMS 架构的一个明显优点是其可靠性。在 JMS 中,一个发送方应用程序可以指定一个消息必须被保存在一个代理管理的数据池中直到所有订阅的接受端都收到它。在接受端,订阅可以是持久的,意味着一个客户端一旦联接到消息代理后将自动接受发送给其所订阅主题的消息,并包括当客户端断线时所有被发送的持久消息。JMS 规范并不指定在消息节点间所使用的通讯低层协议,来保证应用开发人员不用与其细节打交道。尽管 JMS 不是专门设计成来传递 XML,在实际应用中由于它对消息交换高层语义的支持使得它成为一个传递 XML 非常好的方法。

另一方面,基于 XML 的消息传递引擎也在不同的产品中得到了实现。例如,IBM 公司的 Websphere 和 BEA 公司的 Weblogic 都提供了消息传递引擎,这些引擎往往支持消息队列、发布/预订、以及 Web 服务,并且为一些关键的消息传递和交互模式提供平台。所有的消息提供者和消息使用者都不直接进行通信,而是都与引擎进行交互,再由引擎对消息进行处理。

4 在应用中实现数据交换

通过上述的讨论,我们可以看到,要在应用中实现数据交换,用户需要做的是选择合适的产品,并结合应用进行相应的开发工作。

用户的开发工作主要集中在以下的几个方面:

(1) XML 到具体应用中所使用的异种数据源之间

的转换接口。由于具体的应用可能需要访问特定格式的数据源,在这些数据源本身不支持 XML 格式的情况下,用户就需要自己进行接口的开发。

是数据交换技术在当前的研究热点和主要开发工作,文章中给出的应用模式适用于大型的企业级数据交换应用,用户可以自由选择是通过工具还是自己开发应

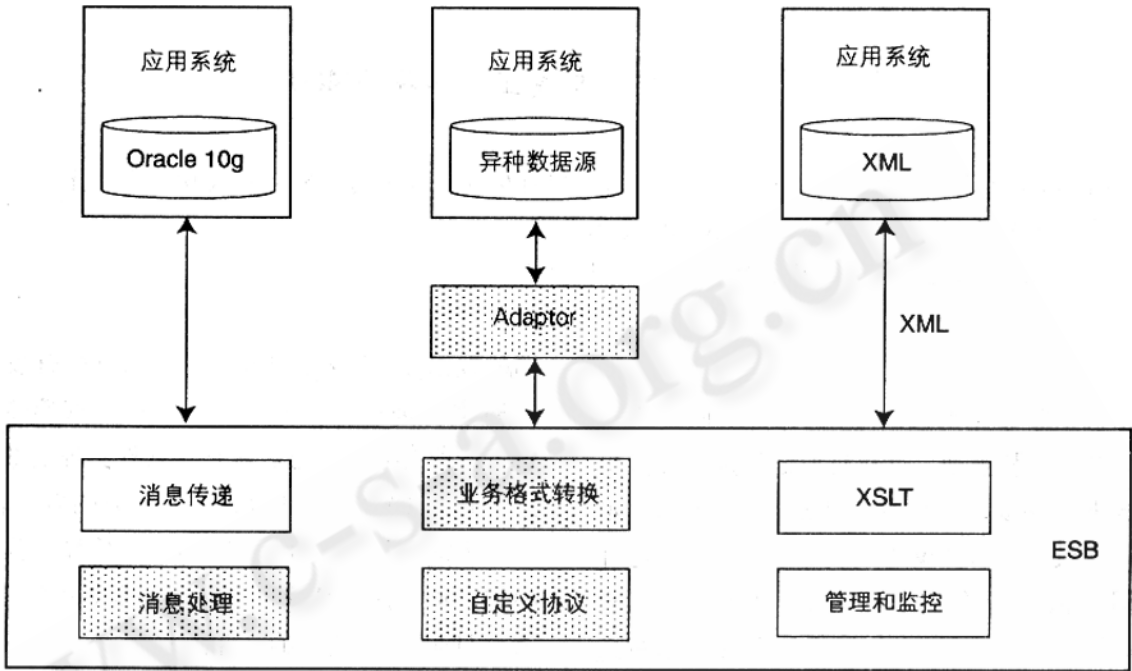


图 3 ESB 中数据交换的应用模式

(2) 消息传递协议的选择与优化。用户需要准备使用的协议,例如 SOAP 或 ebXML。用户还需要结合具体的应用来对协议进行定制,以加入自己需要的内容。

(3) 消息传递引擎中业务相关模块的开发。引擎中需要相关的模块来进行消息数据的处理,格式的转换等业务相关的工作。

(4) 进行数据交换的应用系统的相关开发。要让数据交换有意义,就需要对其进行相关功能的增强,以进行数据的交换和处理交换得来数据。

下图描述了基于 ESB 的数据交换的框架结构图。其中深色的模块代表了用户需要进行的开发工作,浅色的则代表了通过选择合适的第三方软件,用户不必自己开发的模块。

5 结论

本文主要介绍了 SOA、ESB 和数据交换技术,尤其

用的各个模块,随着研究的逐步深入,我们将对上面给出的应用模式进行不断的扩展和细化,进而帮助开发人员能够以构件化的方式迅速将数据交换应用实现。

参考文献

- 1 消息传递:传递 XML 的困惑, Gordon Van Huizen, 2000 年 7 月。
<http://www-128.ibm.com/developerworks/cn/xml/xml-messaging/index.html#N100FF>
- 2 IBM WebSphere 开发者技术期刊:WebSphere V6 消息传递资源入门。
- 3 W3C: XML Protocol Comparisons
<http://www.w3.org/2000/03/29-XML-protocol-matrix>
- 4 面向服务的架构
<http://dev2dev.bea.com.cn/technologies/SOA/index.jsp>