

文章编号:1001-9081(2005)02-0478-03

面向 EAI 的高可用性松耦合的扩展 JCA 架构

雷爱平,尹建伟,陈刚,董金祥

(浙江大学 计算机科学与技术学院,浙江 杭州 310027)

(Leiap@sohu.com)

摘要:企业应用集成(EAI)不可避免的要整合各种异构的数据资源,同时还需要集成业务过程,而 JCA 则是一种较好的基于 J2EE 的解决方案。针对现有 JCA 规范还不成熟、难于使用的问题,主要从企业应用集成的角度对该架构作了一定的扩展,以满足 EAI 过程中的高可用性和松耦合要求。

关键词:JCA;EAI;高可用性;松耦合

中图分类号:TP391 **文献标识码:**A

EAI-oriented and high-availability and loose-coupling XJCA

LEI Ai-ping, YIN Jian-wei, CHEN Gang, DONG Jin-xiang

(Department of Computer Science and Technology, Zhejiang University, Hangzhou Zhejiang 310027, China)

Abstract: EAI (Enterprise Application Integration) must integrate all kinds of heterogeneous data resources and different business process. JCA is one J2EE-based solution which can solve these problems quite well. To avoid the limitation of immature specifications and solve different applying problems in current JCA, this paper expanded JCA mainly from the point of view of EAI, so that it could meet high-availability and loose-coupling in EAI.

Key words: JCA; EAI; high-availability; loose-coupling

0 引言

在计算机时代,随着企业规模和业务的发展以及信息化支持的加深,许多企业都存在着大量的软硬件系统。如何将这些异构的新旧系统集成在一起,是这些企业所面临的重大问题。国内外众多学者对 EAI (Enterprise Application Integration)进行了研究^[1-4]。随着 Java 技术的出现,也有部分学者研究了基于 Java 及相关技术上 EAI 的实现^[5-7]。于 2001 年 8 月 22 日发布的 J2EE 连接器体系 (Java Connector Architecture, JCA) 定义了一个连接 J2EE 平台和异构 EIS (Enterprise Information System) 之间的标准架构,解决了集成中“最后一公里”问题,对于基于 Java 技术的 EAI 系统来说,这是一个里程碑。

作为 JCA 的制定者,Java 标准制定组织 Java Community Process 是 JCA 的主要研究组织。在 JCA 的应用方面,PeopleSoft、Siebel、SAP 等其他许多厂商正在努力为他们企业软件包提交标准 JCA 适配器;Accenture、CSC、EDS、KPMG 等很多领先的系统集成商也正在针对 JCA 这种新框架开发基于标准的集成应用;当然,主要的 J2EE 产品销售商——BEA、Sun、IBM、HP、Oracle、Compaq、NEC 等,都在极力支持以 Java 技术为基础的集成方式^[8]。对于 EAI 集成商来说,研究和应用基于 J2EE 的集成(特别的,包括 JCA)是非常有意义的事情。本文将从应用集成的角度,对 JCA 的规范和应用进

行研究,并对 JCA 进行扩展。

1 JCA 及其优缺点

JCA 定义了一套 EIS 和应用服务器之间的系统级协议 (Contract):连接管理协议、事务管理协议、安全协议、事务流入协议、消息流入协议、生命周期管理协议、工作管理协议。同时, JCA 也为 EIS 的访问定义了一个通用客户接口 (CCI)^[9]。

JCA 的优点很明显。它为 EIS 厂商提供了一种按照开放的产业标准定义 EIS 接口的途径。通过使用公共的可调用接口以及继承 JCA 提供的 QoS 机制,程序员能够在不牺牲性能和系统完整性的前提下,简化 EIS 的集成工作。每一种 EIS 都只要一个 JCA 的实现。因为这个实现使遵循 J2EE 连接器规范的,所以它在所有 J2EE 服务器之间是可移植的。

但 JCA 的缺点也是显然的:1) 不支持异步消息传输:只支持同步的请求/应答模型;2) 不支持双向调用:只定义了从客户端到 EIS 应用的单向调用;3) 不支持长时间运行的事务:仅支持 DTC 类型的短时间事务;4) 对 XML 的支持不够:没有提供基于 XML 的接口来实现与非 Java 的异构系统的集成;5) 不支持适配器元数据:没有提供标准的方法来查询适配器的元数据。

除了上述不足之外,在 EAI 开发过程中,JCA 也较少涉及到业务集成。所有这些都使 JCA 在 EAI 中的应用变得相当

收稿日期:2004-07-27;修订日期:2004-10-18

基金项目:国家 863 计划项目(2001AA415220;2003AA411021);国家自然科学基金资助项目(60273056);航天支撑计划资助项目(417010601)

作者简介:雷爱平(1978-),男,江西高安人,硕士研究生,主要研究方向: workflow、网络中间件、企业应用集成等;尹建伟(1974-),男,江苏徐州人,副教授,博士,主要研究方向:网络中间件、CSCW、信息集成等;陈刚(1973-),男,浙江宁波人,教授,博士,主要研究方向:CIMS、网络安全、工程数据库等;董金祥(1945-),男,浙江温州人,教授,博士生导师,主要研究方向:CIMS、操作系统、工程数据库等。

困难,开发成本和周期都随之增加;另外,也由于整个系统架构的耦合性过于紧密,系统的扩展性亟待加强。

2 扩展 JCA 架构

2.1 架构模型

定义1 事件模型(Event Model, EM)

EM 是客户端和 XJCA 及其整个 EAI 系统之间交互的中介,是应用集成事务的触发点。它是一个多元组,可表示为 $EM = (EID, CON, TID, ET, EP, \dots)$ 。其中, EID 为事件唯一标识;CON 为事件触发条件;TID 为事件所引发事务的唯一标识;ET 为事件类型;EP 为事件优先级。

定义2 适配器模型(Adapter Model, AM)

AM 是 XJCA 或 JCA 与各 EIS 系统之间的底层接口,是 EAI 系统的基础部分。所有应用集成最终都体现为数据的获取、转化和交换。AM 包括三种类型:RA(Resource Adaptor)、PA(Peripheral Adaptor)、OA(Other Adaptor)。RA 是指符合 JCA 规范的适配器接口;PA 是指 XJCA 所扩展的满足 XJCA 要求的适配器接口;OA 是指其他的适配器接口。

定义3 集成任务模型(Integration Job Model, IJM)

IJM 是企业应用集成的原子业务,实现单一的集成任务,通常涉及多个 EIS 或者同一个 EIS 的多个组成部分。IJM 是一个多元组,可表示为 $IJM = (JID, SLST, DLST, JP, \dots)$ 。其中, JID 是集成任务唯一标识;SLST 是任务源列表;DLST 是任务目的列表;JP 是任务在集成事务中的优先级。

定义4 集成事务模型(Integration Transaction Model, ITM)

ITM 是至少包括一个集成任务的集合,代表一个完整的集成业务。ITM 是一个多元组,可以表示为 $ITM = (TID, TP, JC, \dots)$ 。其中, TID 为集成事务唯一标识;TP 为该事务的优先级;JC 为事务所包含的任务集合。根据任务优先级,可以确定各个任务之间的同步和异步时序关系,满足各种集成需求。

2.2 系统框架

针对 JCA 标准的上述不足,论文分别做了相应的改进,同时结合 EAI 的特点,提出了一个新的架构——XJCA。图1所示为 XJCA 架构,其中阴影部分为扩展部分。它主要由以下几部分组成:

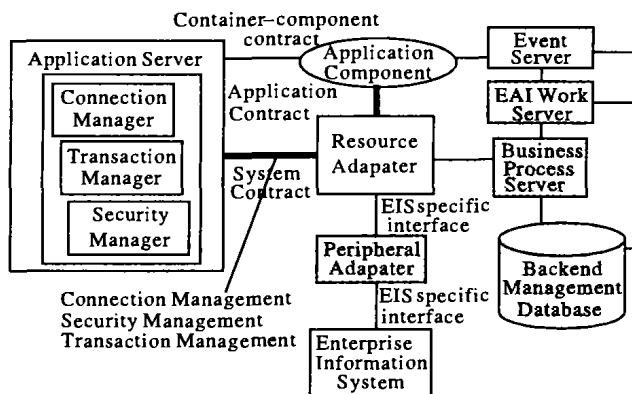


图1 扩展 JCA 架构

Event Server 即事件服务器。该服务器用于接收客户端应用集成请求、触发后台应用集成事务及其实现和返回客户端所需数据。客户端应用程序在适当的时刻(也可能根据计划自动的)发起应用集成请求,产生事件触发应用集成事务,并通知 ES。ES 返回一个内部 ID 给客户端,此后,客户端可根据此 ID 向 ES 获取该应用集成事务的执行状态及其他返回

信息。借助事件服务器, J2EE 应用能异步启动事务请求并连续地处理而不阻塞。

通信模型包括一对一同步(请求/响应)、一对一异步(消息队列)、一对多异步(发布和订阅)三种。为了实现这些通信模型,需要增加消息集成器。它负责在应用程序间发布消息,并且提供透明的消息定位服务。在 XJCA 中,我们采用的方式是在其他 Java 组件中写基于消息/事件的中间件代码,使用集成服务器(ES、EAIWS、BPS、BMD)和各种异构系统交互。

EAI Work Server 即 EAI 工作服务器。该服务器用于管理和调度所有应用集成事务,负责事务调度、事务状态管理和事务生命周期管理等。当有新的集成事件发生时, ES 将该事件相关的事务信息传递给 EAIWS。EAIWS 将该事务置入 EAI 事务队列,并根据事务属性进行调度,同时监控集成事务和任务的完成情况。

Business Process Server 即业务服务器。该服务器用于执行某个具体的应用集成任务脚本。同时也具有定义、修改、删除应用集成任务脚本的功能。作为工作流引擎的一部分, EAIWS 和 BPS 共同管理应用集成属性、调度和运行。根据这些属性,它们能够以某种事务调度算法并发调度长事务,并监视长事务的执行状态、管理长事务的生命周期。

Backend Management Database 即后台管理数据库。该数据库主要用来存储 EAI 运行过程中所需的数据。在 XJCA 中,为了降低组件间的耦合性, ES 并不直接与 EAIWS 交互, BMD 是它们的中介。ES 将新的事件请求置于 BMD 中。EAIWS 每隔一定的时间就来查询一次,以获取最新的事件列表,并由此形成有效未完成事务列表,随后再根据事务调度算法,选择可以执行的事务并发执行。EAIWS 随时记录事务执行的状态,并在事务执行完成之后将执行结果放回 BMD 中,同时修改事务状态。ES 在需要的时候从 BMD 中获取事务执行结果,并将其返回给客户端。BMD 同时也存储应用集成过程中的日志信息,以满足安全管理、开发调试等方面的需要。

Peripheral Adaptor 即外围适配器(见图2)。EAI 致力于解决分布式异构系统数据互操作的问题,能够适应各种不同的环境是非常重要的,因此我们需要各种各样的适配器。另一方面,出于可靠性、稳定性等方面的考虑,在实际应用中我们应该尽可能的采用较为成熟的技术。PA 使用到的关键技术有 JDBC^[10]、RMI、XML、EJB、JSP/Servlet^[11] 以及本地 Adaptor 引擎(Local Adaptor Engine, LAE)等。每一类 Adaptor 所提供的接口都必须能够满足用户读写所需数据的要求,并尽可能的方便其他组件的调用和使用。同时由于 Adaptor 要参与数据交换、转化、合成,它应该能够与一种中间格式相互转化。在 PA 中我们采用 XML 作为中间格式。PA 要求所有的适配器均能够转化为 XML 格式,并识别和转化 XML 格式为特定适配器格式。该功能提供了实时的动态重新格式化信息的能力,从而使得信息能够被异构环境中的多个应用所接受和读取。

特别的,对于 COM、DCOM、COM++、DLL、Java Bean、各种文件等本地接口,我们无法直接远程访问它们。但是另一方面, EAI 系统常常需要通过这些接口来获取或存储某些资源。为了解决这个异地访问本地接口的问题,我们提出了 LAE 技术。LAE 将远程资源操作请求转化为本地请求,并根据操作完成情况返回相应信息给远程机器。作为本地资源的远程代理, LAE 需要安装在本地资源所在机器上。此后,当 PA 需要获取本地资源时, PA 首先向 LAE 发出资源操作请

求。其后,所有的操作均由 LAE 完成,本地资源获取的方法和复杂性均由 LAE 来处理。对于 PA 来说,LAE 屏蔽了本地资源的操作细节,PA 开发将专注于给 RA 和应用集成脚本提供必需和易用的接口上。对于 EAI 开发人员来说,资源的获取和存储都是透明的,因而能够将更多的注意力放到商业逻辑的实现上。

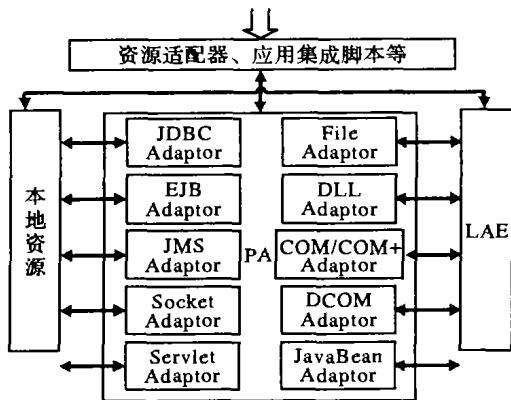


图2 外围适配器和本地 Adaptor 引擎

上述 XJCA 组件分别承担其相应的功能:PA 和 JCA 资源适配器完成数据集成的功能;ES、EAIWS、BPS、BMD 共同完成业务集成的功能;J2EE 应用服务器作为平台,支撑安全、事务、缓冲池等方面的功能。功能的分离一定程度上降低了该架构的耦合性。

3 应用实例

本文提出的这种面向 EAI 的高可用性松耦合的 XJCA 架构已成功开发出原型系统,并初步应用于某国有大型集团(如图3)。系统的开发主要使用 Java 语言,在涉及 DLL、COM/COM++ 等本地接口时使用了 VC++。该系统的应用加快了企业应用集成的开发速度,缩短了企业应用集成项目的开发周期,提高了企业多种异构系统共享资源和相互调用的能力,为集团 CIMS 中新旧异构系统间集成的顺利实施提供了平台层面的支持。下面就以该集团为例,具体描述基于 XJCA 架构的 EAI 开发和使用过程。

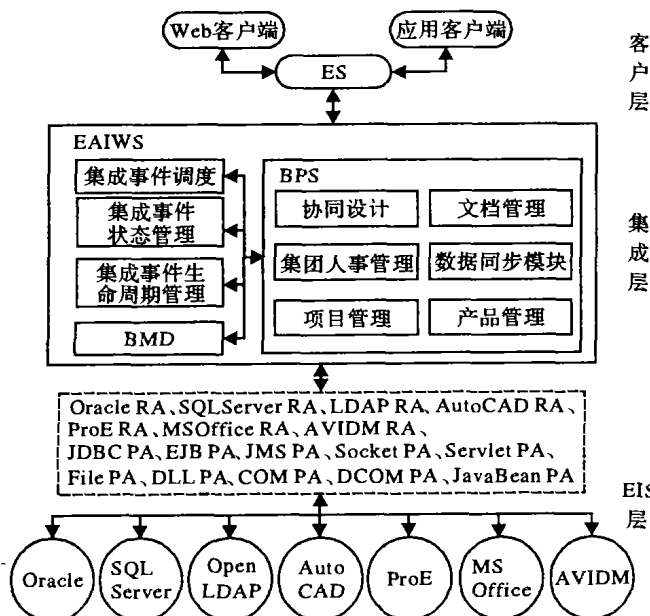


图3 XJCA 应用实例

该集团主要从事大型复杂产品和项目的开发研制,已经

使用的计算机软件主要包括 AVIDM (Aerospace Vehicle Integrated Design Manufacture, 一个协同产品研制平台)、集团财务软件以及各种 OA 软件等。由于系统庞杂,整个 EAI 涉及 Oracle 和 MS SQLServer 两种后台数据库、用于权限管理和访问控制的 OpenLDAP、处理日常文档所用的 MS Office,以及专业设计软件 AutoCAD、Pro/Engineer 等。另外,除了 AVIDM 这个新系统是基于 J2EE、使用 Java 语言开发外,其他旧系统均采用的是非 Java 语言。集团 EAI 系统需要无缝的集成上述系统,以实现在整个集团范围内的协同设计、文档管理、集团人事管理、数据同步管理、项目管理和产品管理。为此我们根据 EIS 层的各种资源开发了相应的 RA。由于有了 XJCA 中 PA 的支持,RA 的开发显得异常轻松。为了实现业务集成,我们在 BPS 上定制了具有上述功能的各种集成脚本,通过部署并结合 ECA 规则,这些脚本所表示的集成任务会在用户认为适当的时候执行。同时,由于集成的表示采用脚本的方式,当业务需求有所变化时,整个系统也能够快速的适应这种变化。

4 结语

EAI 架构模式的共同特点是解耦。本文所提出的 XJCA 的松耦合性体现在:1) 数据集成与业务集成的分离;2) 扩展了通信模型,以支持异步方式。另外,XJCA 也简化了那些支持 JCA 和 EAI 的工具和框架的开发,提供了一定程度的高可用性;1) 以 PA 的方式对外部资源和接口进行封装,以简化 JCA RA 和业务逻辑组件的开发;2) 提供了信息格式化功能和对 XML 的支持;3) 支持客户端和 EIS 应用之间的双向调用;4) 通过 EAIWS 可以较好的解决长事务所引发的问题。

参考文献:

- [1] JOHANNESSON P, et al. Design principles for process modeling in enterprise application integration [J]. Information Systems. 2001, 26(3): 165-184.
- [2] 韦银星,等. 企业应用集成技术研究 [J]. 计算机集成制造系统——CIMS. 2002, 8(8): 593-596.
- [3] ERASALA N, et al. Enterprise Application Integration in the electronic commerce world [J]. Computer Standards and Interfaces. 2003, 25(2): 69-82.
- [4] KOBAYASHI T, et al. Business process integration as a solution to the implementation of supply chain management systems [J]. 2003, 40(8): 769-780.
- [5] SINGH I, et al. Designing Enterprise Applications with the J2EE Platform [M]. Addison-Wesley Pub Co, 2000.
- [6] 姜洪溪,等. 面向业务过程的企业应用集成方法 [J]. 计算机工程与应用, 2003, 39(19): 214-216.
- [7] KANG A. Enterprise application integration using J2EE [EB/OL]. <http://www.javaworld.com/javaworld/jw-08-2002/jw-0809-eai.html>, 2002-08.
- [8] SHARMA R, et al. J2EE Connector Architecture and Enterprise Application Integration [M]. Addison-Wesley Pub Co, 2001.
- [9] Sun Microsystems, Inc. Java 2 Platform, Enterprise Edition Connector Specification [EB/OL]. <http://java.sun.com/j2ee/connector/>, 2003.
- [10] Sun Microsystems, Inc. JDBC 3.0 API Specification [EB/OL]. <http://java.sun.com/products/jdbc/>, 2003.
- [11] Sun Microsystems, Inc. Java 2 Platform, Enterprise Edition Specification [EB/OL]. <http://java.sun.com/j2ee/index.jsp/>, 2003.