

基于 ESB 的 BPM 平台的研究与实现

符 宁^{1,2} 周兴社¹ 薛 文² 张栋毅²

(西北工业大学计算机学院 西安 710072)¹ (解放军西安政治学院 西安 710068)²

摘 要 从系统的角度研究了一种基于 ESB 的 BPM 平台 NpuESB 的设计与实现。在设计符合 JBI 规范的分布式企业服务总线技术的基础上,提出一种分层体系结构的 BPM 平台设计方案。研究了设计与实现过程中的关键技术,提出了基于 Pi 演算的 BPM 环境中的流程建模方法和分布式 ESB 的设计方案。通过引入中央服务器实现了 ESB 环境和流程的集中管理,实现了可视化的流程和 ESB 管理工具。原型系统的应用实验证实了系统的有效性和实用性。

关键词 企业服务总线,应用集成,面向服务的架构

中图分类号 TP31 **文献标识码** A

NpuESB: An BPM Environment Based on Enterprise Service Bus

FU Ning^{1,2} ZHOU Xing-she¹ XUE Wen² ZHANG Dong-yi²

(School of Computer, Northwestern Polytechnical University, Xi'an 710072, China)¹

(PLA Xi'an Institute of Politics, Xi'an 710068, China)²

Abstract The hierarchical architecture of an Enterprise Service Bus (ESB) based BPM environment, NpuESB, was proposed in this paper. NpuESB is a rapid BPM application development platform grounded on the infrastructure of JBI specification-compliant distributed ESB. By introducing ultra server and process management module, NpuESB enables centralized management of the ESB environment and BPM application workflows. Workflows in ESB environment can be modeled with the formal method based on Pi calculus proposed in this paper. Some important design and implementation considerations for the distributed ESB and management tools were also discussed. Application tests of the prototype system demonstrate NpuESB's effectiveness. Compared with traditional techniques and tools, NpuESB delivers better flexibility and scalability meanwhile stays compatible with legacy systems, thus reduces EAI application development and maintenance costs to a greater extent.

Keywords Enterprise service bus, Enterprise application integration, Service oriented architecture

1 引言

BPM(Business Process Management)指以服务计算的模式支持企业级业务过程建模、分析、优化、执行和监控的软件技术。企业服务总线(Enterprise Service Bus, ESB)是一种能够简化在企业中实施 SOA 的服务计算中间件^[1]。本文从系统的角度探讨了一种基于 ESB 技术的 BPM 平台、NpuESB 的设计和原型实现。NpuESB 以 ESB 为底层软件架构,采用分层的体系结构设计,是一个支持 BPM 应用开发、部署、管理和监控的软件环境。

2 ESB 技术与 JBI 规范

企业服务总线定义通常如下:它是由中间件技术实现的支持面向服务架构的基础软件平台,支持异构环境中服务基于消息和事件驱动模式的交互,并且具有适当的服务质量(QoS)和可管理性^[1]。ESB 通过提供一种标准的软件底层架构支持服务的运行,以标准的消息格式统一了服务之间的交

互方式,以中间件的方式提供服务容错、负载均衡、QoS 保障和可管理功能。ESB 使应用系统之间可以更容易地进行交互,为系统集成提供了更好的扩展性和灵活性。

JBI 即 Java Business Integration,是 SUN 公司提出的企业服务总线的规范^[2]。JBI 规范定义的 ESB 环境包含 3 方面的内容:标准消息路由器(Normalized Message Router, NMR)、JBI 管理部分和 JBI 组件框架。NMR 实现组件之间的消息路由。JBI 组件框架为 JBI 组件提供一个可插拔的组件容器。JBI 管理部分基于 JMX 实现 JBI 环境中系统和组件的管理。我们在 JBI 规范的基础上,对其进行扩展,实现了符合该规范的分布式企业服务总线^[3],支持负载均衡、服务失效容忍、持久消息传输,克服了单点故障的问题,并将该分布式 ESB 作为 BPM 环境的基础软件平台。

3 BPM 平台的设计思想与系统架构

在选择 ESB 作为基础软件平台的基础上,我们提出如下的 BPM 环境的主要设计目标:

到稿日期:2009-07-02 返修日期:2009-09-13 本文受国家 863 计划资助项目(2006AA012162),十一国防预研资助项目(06004089),国家发展改革委员会资助项目(20052139)资助。

符 宁(1976-),男,讲师,主要研究方向为分布计算、可信计算, E-mail: funingde@gmail.com;周兴社 男,博士生导师,主要研究方向为分布计算等。

- 实现符合 JBI 规范的分布式企业服务总线,为 BPM 环境提供基础软件平台。

- 支持基于 SOA 的 BPM 集成机制,支持 Web Services, ERP, CRM 等企业业务系统以及企业间应用系统之间业务信息的传递,实现应用系统的按需快速组合。

- 实现可视化的 BPM 应用流程设计和验证工具,使普通用户使用直观的方式定义 BPM 应用或者服务集成,并且逻辑上的定义能够自动映射到物理实现上。

- 实现集中式的 BPM 应用部署、管理和监控工具,用户能够使用这些工具实现对 BPM 应用和 ESB 环境的集中式管理。

针对以上系统设计目标,我们采用的设计思路是:以 ESB 为基础平台,通过适配器将各种外部应用连接至 ESB,实现系统的互连和互操作;将 BPM 应用中常用的基础功能(适配器和数据转换类组件)单元封装为 JBI 组件,以预置组件库的方式提供给用户;通过多个组件之间的协作完成一个 BPM 应用;对程序组件和 BPM 应用抽象建模,通过模型检验的方法验证 BPM 应用的正确性;引入中央管理服务器,对整个 ESB 环境和各个 BPM 应用进行管理和监控;为用户提供一组管理工具,包括可视化的集成应用流程管理工具、ESB 环境管理工具等,用户利用这些工具通过对中央服务器的管理,实现对 BPM 应用和 ESB 环境的管理。

NpuESB 的逻辑层次结构如图 1 所示,系统分为 4 个层次。ESB 层是整个系统的基础软件平台。ESB 层为各种应用提供标准的 JBI 接口,为程序组件提供运行环境的支持,并为 BPM 环境提供运行时支持。

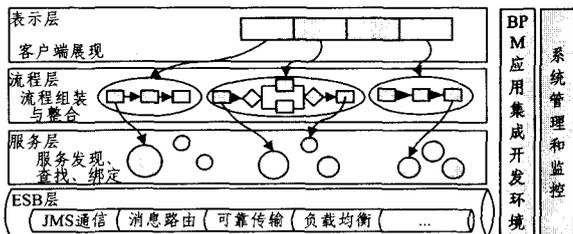


图 1 NpuESB 层次结构图

服务层的运行实体是处于运行态的 JBI 组件,每个组件完成一个基本的功能单元,以标准的接口对外呈现其服务功能。外部应用可以以适配器组件的形式连接进入 NpuESB,进而可以与其它系统进行交互。与流程层比较,本层的粒度较小,组件是构成流程的基本单元。

流程层的基本视图是应用流程,以流程为单位来管理、部署和监控应用的执行。流程是 NpuESB 的一个应用程序,由服务层的多个服务组件协作完成。流程中的具体任务单元都可以映射到服务层的功能组件。

表示层是系统与用户的交互层,以可视化的形式将流程展现为直观的图形化实体。用户可以在图形界面中通过选择和操作菜单、图表等元素来开发和管理应用流程。

在系统的每个层引入管理模块,以实现应用以及整个 BPM 环境的管理。我们开发了一个集成的应用开发环境,用户可以使用该工具开发、部署和管理应用流程。

从实现的角度,我们采用了如图 2 所示的系统架构,系统

由工具层、管理层、ESB 层和遗留应用层构成。

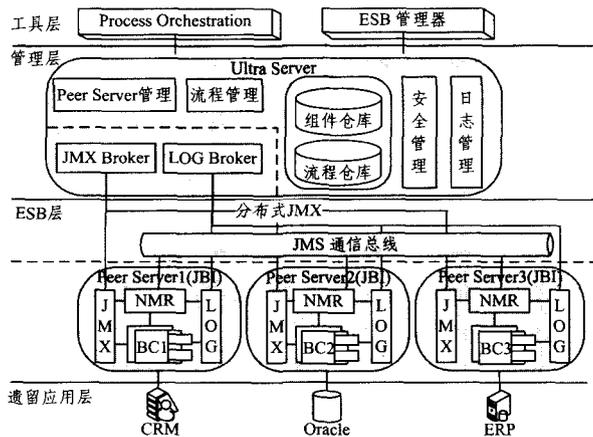


图 2 NpuESB 体系结构图

ESB 层由连接在 JMS 通信总线上多个对等的 JBI 环境共同构成。在实现上由与 JMS 服务器连接的多台计算机构成,称之为 Peer Server(PS)。

管理层对整个 ESB 环境和所有 BPM 应用进行管理。可由单独的集中式服务器实现,称之为 Ultra Server(中心服务器,简称 US)。US 是整个 NpuESB 系统的控制和管理中心。US 实现了对 BPM 环境和 BPM 应用的集中式管理。其主要功能模块包括服务管理、流程管理、PS 管理、仓库管理、日志管理和安全管理模块。

其中,服务管理模块负责系统的启动和其它功能模块的加载。流程管理模块对所有动态流程进行管理,负责流程的创建、部署、启动、暂停和卸载等管理任务。PS 管理模块管理 ESB 网络中的 PS 的状态及节点配置信息。仓库管理模块用于存储和管理静态流程和组件。JMX Broker 是各 PS 节点的 JMX 管理模块的集中远程代理。安全管理模块为系统的安全提供保障,对用户的操作权限进行安全控制。

工具层包含两个交互工具:ESB 管理器和 Process Orchestration。ESB 管理器以 Web 方式支持用户对整个 ESB 环境进行监控和管理。Process Orchestration 是一个可视化的流程设计、管理、监控工具。用户可以使用该工具以拖拽的方式设计 BPM 应用,并可通过该工具管理应用流程。

遗留应用层是企业中待集成的多种应用系统,包括 CRM、ERP、各种数据库、MIS 系统等。这些系统通过适配器连接至 ESB,并实现与其他系统的交互。

4 流程的建模与验证

基于 NpuESB 的 BPM 由 ESB 环境中的多个程序组件协作完成。如何保证这些程序组件组合的正确性,是十分重要的问题。我们基于 Pi 演算^[4]对 BPM 的流程进行形式化描述和建模,并利用形式化工具对建立的组合模型是否正确以及是否满足需求进行验证,确保流程的正确性。如图 3 所示,以一个 BPM 应用为例说明流程的建模和验证方法。CRM 应用中取得客户订单并送往 ERP 系统,ERP 系统根据订单中的内容做判断。如果用户订单满足要求则被批准,经数据转换后,通过 SMTP 服务向用户发送确认邮件。否则,被拒收的订单通过显示服务显示在业务员的屏幕上。

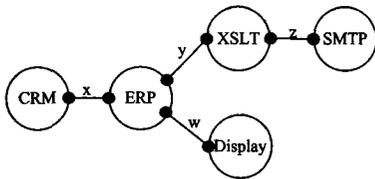


图3 流程的Pi演算流程图

首先,将ESB环境下的程序组件表示为Pi演算的进程。组件包含若干端口,连接两个组件端口的称之为通道(Channel)。组件之间经由通道发送和接收消息完成一次交互过程。示例中各消息表示为:

Purchase Order: CRM系统中的客户订单信息(Po);

Acceptance: ERP系统批准的订单信息(Acc);

Rejection: ERP系统拒绝的订单信息(Rej);

Mail: 经过 XSLT 组件转换过的订单信息(Mail)。

对该流程进行建模如下(发出的消息以尖括号<>表示):

CRM 组件: 设 $\bar{c} = \{x, Po\}$, $CRM(\bar{c}) = \bar{x}(Po)CRM(\bar{c})$

ERP 组件: 设 $\bar{e} = \{x, y, w, Po, Acc, Rej\}$, $ERP(\bar{e}) = x(msg)[msg = Po](\bar{y}(Acc)ERP(\bar{e}) + \bar{w}(Rej)(\bar{e}))$

XSLT 组件: 设 $\bar{l} = \{y, z, Acc, Mail\}$, $XSLT(\bar{l}) = y(msg)[msg = Acc]\bar{z}(Mail)XSLT(\bar{l})$

SMTP 组件: 设 $\bar{s} = \{z, Mail\}$, $SMTP(\bar{s}) = z(msg)[msg = Mail]SMTP(\bar{s})$

Display 组件: 设 $\bar{d} = \{w, Rej\}$, $Display(\bar{d}) = w(msg)[msg = Rej]Display(\bar{d})$

该BPM流程由CRM适配器、ERP组件、XSLT组件、SMTP组件和Display组件共同构成。令 $\bar{p} = \{x, y, z, w, Po, Acc, Rej, Mail\}$, 该流程可描述为 $process(\bar{p} = CRM(\bar{c}) | ERP(\bar{e}) | XSLT(\bar{l}) | Display(\bar{d}))$ 。

用户设计好一个BPM流程之后,首先被转换为本案所示的抽象模型。然后利用MWB(Mobility Workbench)工具包^[5]来推演系统的行为,同时验证模型的正确性。例如发现系统行为不完整、死锁、缺少同步等。每次用户修改流程后,都要进行这些检查,以保证该流程是按照需求设计,且能正确地实现。

5 流程管理与监控

流程是分布式ESB环境中不同物理位置的多个运行态程序组件,按照一定的组合和协作约定,共同完成一个BPM应用的执行过程。我们在US上设置流程管理模块来实现对系统中活动状态的流程进行管理。在流程管理模块内部流程以对象的形式出现,每个流程有其属性、状态并实现deploy()、start()等若干管理方法。每个流程对象负责管理属于该流程的、分布在多个PS节点上的组件集合。

当创建一个BPM应用时,US流程管理模块首先为其创建一个流程对象。该对象根据该流程的具体配置参数和部署文件,依次通知PS下载不同的程序组件,并将其实例化。当将各个组件的状态设置为stopped后,US流程对象的状态被设置为stopped。流程启动时,该对象依次启动各个PS上的程序组件,并设置自身状态为started。流程的暂停、继续等控制也是采用类似的方法实现。当流程销毁时,流程对象依次销毁不同PS的属于该流程的程序组件,并最终销毁自身。客户端工具PO通过与流程管理模块的交互来管理NpuESB

范围内的活动态流程。考虑到US可能存在失效的问题,采用不依赖于外部数据库的持久化的对象管理技术来管理流程对象。US与PS之间的管理机制采用的是JMX的远程管理机制。

6 分布式企业服务总线

为了克服JBI规范所定义的集中式ESB架构的不足,我们设计并实现了符合该规范的分布式企业服务总线,将多个PS通过JMS消息通信总线进行连接,以共同构成分布式的ESB。分布式ESG设计的核心问题是如何实现不同JBI节点上的组件之间的消息交互。为此,我们设计了分布式的消息路由器。分布式NMR由物理上独立的多个NMR通过JMS通信总线连接而成。组件要与其他节点上的组件进行交互时,不考虑其他组件的位置,只需将服务请求消息发给本地的NMR,由NMR进行跨节点的消息转发。我们在JMS服务器上为每个JBI节点的NMR分配一个消息队列。每个NMR监听自己的消息队列,当发现有发给自己的消息时进行处理,将该消息转发至目的组件。当NMR要将消息发送给其他节点的组件时,则将该消息发送给目的NMR对应的消息队列。

为了解决分布式环境下的消息路由的发现和同步,将每个节点上的消息路由表区分为内部和外部两种。内部路由表保存本地的组件、接口和endpoint的对应关系。外部路由表保存了非本地组件与其JBI环境的ID的对应关系。在每个节点上设立路由表同步模块,在JMS服务器上创建用于发布路由信息变更的topic,所有的路由表同步模块监听该topic。当某个节点上的路由信息发生变化时,该节点广播这次变更信息来通知其他节点更新消息路由表。另外,我们通过超时等待的方法来发现ESB环境中的节点失效问题,通过JMS支持持久消息传输能力来保证节点失效时消息不会丢失。通过同名的冗余服务来提供系统的可用性。关于分布式ESB的更具体的实现细节可以参考文献[3]。

7 原型实现和实例验证

我们依据以上设计思想和体系结构对NpuESB做了原型实现。Process Orchestration(PO)是为用户开发的进行BPM应用的快速集成、部署和管理的可视化工具,其界面如图4所示。NMR是ESB的核心模块,其性能是决定整个系统性能的关键因素。我们采用SEDA技术,使得NMR在并发访问的负载很高的情况下保持很好的性能和稳定性;采用AOP的方法分离程序的日志、安全、跟踪、错误处理的代码;采用组合机制将其组合到实现业务逻辑的主程序代码中。在提高开发效率的同时提高了程序的可读性和可维护性。

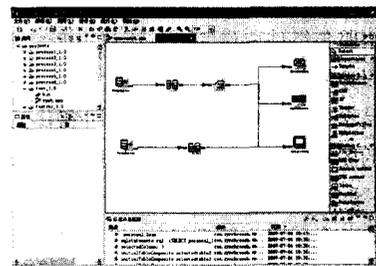


图4 Process Orchestration 界面图

在组件的实现方面按照JBI规范将其分为两类:SE和

BC。SE 类组件用于业务逻辑的控制;BC 组件则用于将外部应用系统连接进来,亦称之为适配器类组件。目前实现的 SE 类组件主要包括 XSLT 数据转换组件等。适配器主要包括主流数据库适配器等。

NpuESB 提供可视化的开发工具和丰富的预置组件,即使是非专业的用户也能够基于 NpuESB 开发多种 BPM 应用。我们在西安高新区数字园区的多个企业进行了包括数据集成、EAI 应用与 B2B 应用的多项试验。其中如图 5 所示的异构数据源的数据集成是基于 NpuESB 的一个典型应用。在这种应用场景中,多个 DBOut 适配器被配置成从不同物理位置的异构数据库系统中抽取数据,经过 XSLT 组件转换为统一的数据格式,并最终通过 DBIn 适配器写入数据中心。在原型系统实验中,非专业用户使用 PO 工具进行快速的应用开发,顺利地进行了应用的部署和管理工作。这些相关实验表明,NpuESB 达到了预期的设计目标。

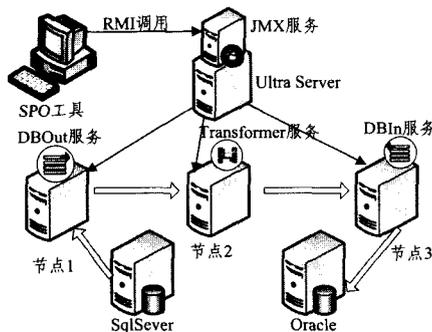


图 5 数据集成场景

与传统的 EAI 技术比较,NpuESB 具有如下的特点和优势:NpuESB 是基于服务计算的 BPM 环境,因此具有更好的灵活性和可复用性;NpuESB 遵循 JBI 标准,使得系统具有良好的兼容性和可扩展性,能够为多种技术和遗留系统提供支持;NpuESB 支持以流程编排的方式快速构建新的应用,提高了应用系统集成的开发速度;NpuESB 通过引入中央服务器实现了对 ESB 环境的集中管理,通过提供可视化的 ESB 和流程管理工具方便了用户的使用。

8 相关工作

在服务集成环境的研究方面,文献[6,7]提出了各自的服务组合环境和互操作方法,其中 Self-Serv^[6]是一个面向动态的网络环境的敏捷 Web service 组合软件环境。在流程建模

方面,Self-Serv 基于状态图,而 NpuESB 采用的是理论更完备的进程代数方法。从应用目标角度看,本文是研究构建的面向 EAI 和 B2B 应用的软件平台,着重考虑以服务的方式对遗留应用系统进行集成,不局限于对 Web service 进行组合。另外,本文的工作是基于开放的 JBI 规范,这使得 NpuESB 具有更好的开放性和适应能力。基于 ESB 面向具体的应用目标构建特定的应用系统已有较多研究。本文与这些工作的区别之处是致力于构建一个通用性较强的 BPM 软件平台。

结束语 本文从系统的角度研究了一种基于 ESB 的 BPM 平台 NpuESB 的设计与实现。在设计符合 JBI 规范的分布式企业服务总线技术的基础上,提出分层体系结构的 BPM 平台设计方案。提出了基于 Pi 演算对 ESB 环境下的流程进行建模描述的方法,并通过引入中央服务器和流程管理模块的方法实现了 ESB 环境和流程的集中管理。提出了符合 JBI 规范的分布式 ESB 的设计方案。设计了可视化的流程管理工具和 ESB 管理工具。原型系统的应用试验表明,本设计方案有效。与传统的 EAI 技术和工具进行了比较,结果表明 NpuESB 具有更强的灵活性和可扩展性,能够兼容多种应用系统开发技术,有效降低 EAI 应用的开发和维护成本。

参考文献

- [1] Chappell D A. Enterprise Service Bus[M]. Sebastopol: O'Reilly Publishing, 2004
- [2] Sun Microsystems Inc. Java Business Integration (JBI) 1.0[S]. 2005
- [3] 符宁,周兴社,张海辉. 基于 JMS 的分布式 ESB 的设计与实现[J]. 计算机科学, 2007, 34(12): 118-121
- [4] Milner R. A calculus of mobile processes part I/II[J]. Journal of Information and Computation, 1992, 100(01): 1-77
- [5] Victor B, Moller F. The mobility workbench: A tool for the pi calculus[A] // Proceedings of the 6th International Conference on Computer Aided Verification [C]. California: Springer-Verlag, 1994: 428-440
- [6] Benatallah B, Dumas M, Sheng Q Z, et al. Declarative Composition and Peer-to-Peer Provisioning of Dynamic Web Services[C] // 18th International Conference on Data Engineering 2002. San Jose, CA, USA, 2002: 297-308
- [7] 胡春明, 怀进鹏, 孙海龙. 基于 Web 服务的网格体系结构及其支撑环境研究[J]. 软件学报, 2004(15): 1064-1072

(上接第 22 页)

- [34] Chen X, Neubert B, Xu Y Q, et al. Sketch-based Tree Modeling Using Markov Random Field [J]. ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia '08), 2008, 27(5)
- [35] Igarashi I, Matsuoka S, Tanaka H. Teddy: a sketching interface for 3d freeform design [J]. Proceedings of SIGGRAPH '99, 1999: 409-416
- [36] Shum H, Kang S. A review of image-based rendering techniques [C] // Proceedings of IEEE/SPIE Visual Communications and Image Processing. Perth, Australia, 2000: 2-13
- [37] 刘钢, 彭群生, 鲍虎军. 基于图像建模技术研究综述与展望[J]. 计算机辅助设计与图形学学报, 2005, 17(1): 18-27
- [38] Quan L, Tan P, Zeng G, et al. Image-based plant modeling [J].

- ACM Transactions on Graphics (Proceedings of SIGGRAPH '06), 2006, 25(3): 772-778
- [39] Tan P, Fang T, Xiao J, et al. Single image tree modeling [J]. ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia '08), 2008, 27(5)
- [40] Sakaguchi T. Botanical tree structure modeling based on real image set [C] // ACM SIGGRAPH 1998 (Tech. Sketch): 272
- [41] Comaniciu D, Meer P. Mean shift: A robust approach toward feature space analysis [J]. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2002, 24(5): 603-619
- [42] Habel R, Kusternig A, Wimmer M. Physically Guided Animation of Trees [J]. Computer Graphics Forum, Proceedings of Eurographics, 2009, 28(2)