

基于 Web Service 的异构 workflow 管理系统的集成和互操作研究

李红信 范玉顺

清华大学自动化系 (北京 100084)

E-mail: hxli@cims.tsinghua.edu.cn

摘要 本文首先分析了分布、异构 workflow 管理系统的互操作对各种交互模式的需求, 针对现有的 workflow 管理系统的分布式应用解决方案的不足之处, 提出了一种基于 Web Service 的支持分布、异构 workflow 管理系统之间集成和互操作的实现方法。Web Service 因其开放、标准化而受到业界广泛重视, 已成为新一代基于 Internet 建立分布式应用的规范。本文从异构 workflow 管理系统之间的信息集成、workflow 引擎之间调用语义的实现、workflow 执行服务的描述、发布和绑定等方面, 论述了 Web Service 对异构 workflow 管理系统之间集成和互操作的支持。

关键词 Web Service workflow 管理 集成

Web Service-based Integration and Interoperation of Heterogeneous Workflow Management System

Li Hongxin Fan Yushun

(Department of Automation, Tsinghua University, Beijing 100084)

Abstract: In this paper, the interaction mode required by the interoperation between distributed and heterogeneous workflow management system are analyzed first. Then a Web Service-based solution for supporting the integration and interoperation of heterogeneous workflow management system is proposed. Web Service is widely accepted for its openness and standardization. It has become the new paradigm for Internet-based distributed application. Various aspects in Web Service-based integration and interoperation of heterogeneous workflow management system, such as the information integration between heterogeneous workflow management systems, the implementation of the call semantics between workflow engines and the description, publishing and binding of workflow enactment service, are discussed in this paper.

Keywords: Web Service, workflow management, integration

1 引言

企业通过各种业务过程向外界提供产品和服务, 从而实现其目标。因此, 企业业务过程的效率成为企业赢得基于时间的竞争 (Time-based Competition) 的关键。workflow 技术是实现企业业务过程自动化的使能技术。利用 workflow 管理系统支持企业之间业务过程的衔接和集成显得日益重要。

workflow 管理系统的应用背景决定了其用户工作在不同的软、硬件平台之上, 分布在不同的地理位置。为了利用 workflow 管理系统支持企业之间业务过程的协作和集成, 消除企业之间的业务过程因为信息载体或信息表示方式的不同而引起的停顿, 企业之间的 workflow 管理系统需要集成和互操作。这要求 workflow 管理系统所提供的工作流执行服务能够屏蔽底层应用系统的异构和分布特性, 从而给其它 workflow 管理系统提供单一的、透明的 workflow 执行服务的界面。因此, 分布、异构

workflow 管理系统之间的互操作必须利用分布式计算环境 (Distributed Computing Environment) 实现。分布式计算的基本原理是客户机/服务器模型。

本文首先分析了分布、异构 workflow 管理系统的互操作对各种交互模式的需求, 针对现有的 workflow 管理系统的分布式应用解决方案的不足之处, 提出了一种基于 Web Service 的支持分布、异构 workflow 管理系统之间集成和互操作的实现方法。Web Service 因其开放、标准化而受到业界广泛重视, 已成为新一代基于 Internet 建立分布式应用的规范。本文从异构 workflow 管理系统之间的信息集成、workflow 引擎之间调用语义的实现、workflow 执行服务的描述、发布和绑定等方面, 论述了 Web Service 对异构 workflow 管理系统之间集成和互操作的支持。

2 workflow 管理系统之间的互操作模式

企业之间业务过程集成的需求要求各自的工作流管理系统支持灵活的“请求-响应”模式。某些请求需要 workflow 管理系统做出快速响应, 例如: 一个 workflow 管理系统的工作流引擎向另一个 workflow 管理系统发出的创建工作流实例的请求, 要求后者尽快返回执行结果, 因此, 这两个 workflow 管理系统的工作流引擎之间需要采用同步交互方式。而对某些 workflow 引擎发出的请求, 则无需返回执行结果, 例如: 对某些操作执行完毕后写入日志文件的请求, 可以采用这种方式进行处理。然而, 在某些要求确保可靠执行的情形下, 即使无需返回结果, 也要求服务器返回执行完毕的确认信息。

客户/服务器之间的“请求-响应”模式可用调用语义来描述, 与返回结果有关的调用语义定义了是否返回调用结果, 或是否要求确认请求的执行情况。与通信方式有关的调用语义定义了调用是采用同步通信还是异步通信。采用同步通信时, 客户程序在发出请求后被挂起, 直到请求被服务器处理完毕才能进行后续的处理; 而采用异步通信的客户程序在发出请求后无需等待, 即可进行其它工作。若需要服务器返回执行结果, 则在进行其它工作同时期待着接到请求结果。

设 C 表示与通信方式有关的调用语义的集合, R 表示与返回结果有关的调用语义的集合,

则 C 与 R 的直积集 $C \times R$ 表示同时使用上述两类调用语义的调用语义集合。根据上面的分析:

$$C = \{Synchronous, Asynchronous\}$$

$$R = \{Result, No-result, Ack\}$$

其中 Ack 表示要求返回确认信息。由于在服务器既不返回执行结果也不返回确认信息的情形下, 客户程序的等待将造成死锁, 因此, 应禁止 ($Synchronous, No-result$) 调用语义, 从而表示上述两类调用语义的集合应为:

$$C \times R \setminus \{(Synchronous, No-result)\}.$$

3 workflow 管理系统之间分布式计算的实现方法

WfMC 的 workflow 管理参考模型只给出了构成 workflow 管理系统的各模块之间以及异构的 workflow 管理系统之间的交互接口, 但没有考虑分布式计算的实现问题。

传统的分布式计算采用远程过程调用 (RPC) 实现, 客户/服务器之间采用 RPC 通信的主要问题是: 开发人员除了解决业务问题外, 还必须处理底层网络的一些问题, 并负责定位的服务器, 从而降低了开发业务问题解决方案这一问题的重要性。另外, 传统的远程过程调用本质上是同步的。早期的分布式计算还采用网络应用编程接口 (API) 实现, 网络 API 支持同步通信和异步通信, 但它没有提供编码数据设备, 以使多个平台上有不同数据表示的应用程序能够理解这些数据, 这常常要求创建和使用额外的软件, 处理诸如不同平台上数据表示的差异等。

对分布式应用的这些要求导致了大量的通信中间件商品或标准的产生, 使参与分布式计算的客户/服务器的通信建立在通信中间件 (Communication Middleware) 之上, 使用通信中间件时, 应用程序开发者只需处理某种类型的单个 API, 而程序之间通信的其它细节则由通信中间件实现, 即开发者无需关心较低层细节的复杂性, 从而将精力放在解决业务问题和优化业务逻辑方面, 提高了应用开发的效率。通信中间件分为面向消息的中间件 (MOM)、对象管理组织的对象管理组织 (OMG) 的对象请求代理和 Sun Microsystem 的 J2EE 等。MOM 使客户/服务器之

间以消息为信息载体,以容纳消息的队列为通信中介,例如 IBM 的消息队列产品 MQSeries 和 Microsoft 的 MSMQ 为代表。然而,消息队列本质上是异步通信,即它支持的调用语义是:

{(Asyn, result), (Asyn, No-result), (Asyn, Ack)}

对象请求代理在客户程序和服务器程序之间插入对象代理层,较好地解决了服务的透明性问题,其体系结构以 CORBA 为代表。CORBA 支持同步通信,并以延迟同步 (Differed Synchronous) 通信的方式实现异步通信。

Java 使得应用程序编写一次就可应用于任何支持 Java 虚拟机的平台,然而必须确保客户程序和服务器程序两端都运行着 Java。

使用通信中间件存在的问题是:要解决分布式计算中的异构问题和服务的透明问题,必须采用某个特定的开发商提供的通信中间件。而在经济全球化的背景下进行电子商务,期望企业之间采用同一通信中间件进行通信是不现实的。

为了使企业之间进行有效协作,需要开放的、标准化的、简单易用的分布式计算实现方案,Web Service 就是在这种应用背景下发展起来的支持企业之间应用集成的新技术。Web Service 支持创建开放的分布式系统的构件,并支持通过 Web 访问此构件提供的服务。Web Service 之所以受到业界

和学术界的广泛关注,就是因为它的开放性、标准化和易于使用。它建立在超文本传输协议 (HTTP) 的基础上,以简单对象访问协议/可扩展标识语言 (SOAP/XML) 作为通信协议,使用 Web 服务描述语言 (WSDL) 作为服务接口的描述方式,使用通用描述、发现和集成 (UDDI) 发布服务、查找和绑定所需的服务。

4 基于 Web Service 的工作流管理

Web Service 提供了一个位于应用层和传输层之间的抽象层,隐藏了分布式计算的许多实现细节,为分布式、异构的工作流管理系统之间的集成和互操作提供了一个基础平台,基于 Web Service 的工作流管理系统的集成方案如图 1 所示。

4.1 异构工作流管理系统之间的信息集成

随着办公自动化程度的提高,单纯的数据交换和共享已不能满足企业之间应用的集成。据统计,企业日常业务所处理的所有信息中,“典型”的数据(例如存储在关系数据库中的账户记录或地址)仅占 10%-20%,其余均为复杂的文档,文档中包含着格式化的文本、图形、视频、及语音及上下文环境等信息,而异构系统对文档(甚至对数据)的表示和存储方式不同,使得业务过程之间的无缝集成造成困难。而 XML 提供了跨平台的数据编码和组织方法,因此,异构工作流管理系统之间交互的信息可以用 XML 文档进行描述。XML 文档是文本文件,并遵循与之关联的文档类型定义 (Document Type Definition, DTD) 或 XML 结构,任何可以处理文本的计算机就可以处理 XML。XML 文档描述了一些实体,通过这些实体可以引用位于内存、硬盘或 Web 中的另外一些实体。XML 的内容区可以包含诸如文本、数据、图像等不同的域类型,适合描述异构工作流管理系统之间所交换的信息。

仅用 XML 表示异构系统之间所交换的信息还不够,还必须定义异构系统之间的访问协议。在基于 Web Service 实现分布式应用时,简单对象访问协议 (SOAP) 以 XML 形式提供了一种用于在分布环境中跨平台交换结构化和类型信息的机制。借助绑定于 HTTP 之上的 SOAP 协议,分布式应用系统传输的信息实际上是 XML 文档,从而

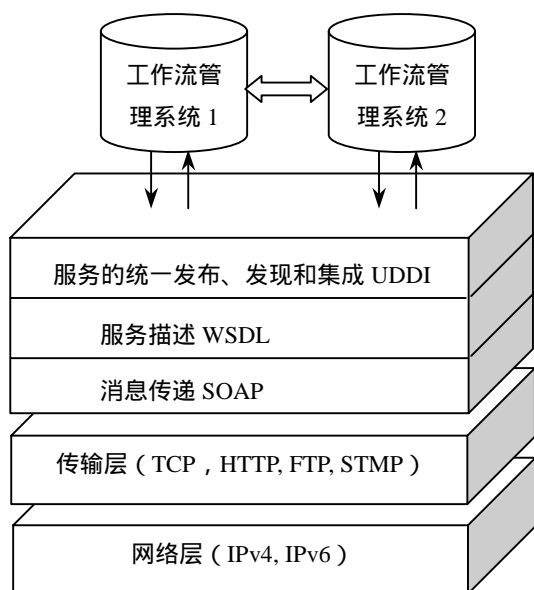


图 1 基于 Web Service 的工作流管理系统的集成和互操作

实现了分布式应用与编程语言及系统平台的无关性,使 SOAP 能够应用于从消息传递到远程过程调用 (RPC) 的各种系统。而以前的分布式应用系统中客户/服务器之间的调用方式则与复杂的分布式对象标准或中间件有密切的联系。

4.2 调用语法的实现

企业之间通过各自的工作流管理系统进行业务过程的集成时,不同的工作流管理系统的工作流引擎之间的互操作仍然基于客户/服务器模型,以“请求/响应”方式进行交互。这里客户或服务器角色的定位是相对的,当一个工作流引擎向其它工作流引擎发出请求时,它是客户机,而当它接受其它工作流引擎的请求时,则是服务器。

Web Service 提供 4 种客户端和服务的提供者 (即客户/服务器) 之间的交互模式:

- 请求/响应 (Request/Response)
- 要求/响应 (Solicit/Response)
- 单向 (One-way)
- 通知 (Notification)

前两种模式是同步通信方式,区别仅在于由客户端还是服务的提供者发出呼叫,“请求/响应”模式由客户端发出请求,引起特定的服务器响应此请求,其调用语义与传统的客户/服务器之间的调用语义相同,“请求/响应”模式也是基于 Web Service 的分布式应用的典型交互方式。“要求/响应”模式则由服务的提供者发出呼叫,期望得到来自客户端的响应。

“单向”与“通知”模式是异步通信方式,区别也仅在于由客户端还是服务的提供者发出呼叫,对“单向”模式,客户端以“发送后不管”的方式把消息发送给服务的提供者而不期望得到响应。对“通知”模式来说,消息的发送方向正好与此相反,并且也不期望得到客户端的响应。

Web Service 提供的这些交互模式完全可以实现 (1) 式所代表的调用语义。除 (Asynchronous, Result) 及 (Asynchronous, Ack) 调用语义外,Web Service 都有直接对应的交互模式,但 (Asynchronous, Result) 及 (Asynchronous, Ack) 调用语义可用“单向”模式与“通知”模式的组合实现。各调用语义及其实现方法总结如表 1 所

示。

表 1 异构工作流管理系统之间调用语法的 Web Service 实现

(Synchronous, Result)	请求/响应
(Synchronous, Ack)	请求/响应
(Asynchronous, Result)	单向 + 通知
(Asynchronous, Ack)	单向 + 通知
(Asynchronous, No-result)	单向

4.3 工作流执行服务的描述和发布

Web Service 要求服务的提供者负责定义服务程序接口的语义,Web 服务描述语言 (WSDL) 是用来对 Web 服务的消息逻辑结构、语法及网络逻辑特性进行描述的 XML 语言,WSDL 还负责描述消息交换格式、服务绑定以及对服务位置的引用。客户机借助 WSDL 文档理解服务提供者的接口和语法,然后向 Web 服务发出请求。

在 Internet 环境下的客户应用能够访问一个 Web 服务之前,必须能够找到该服务的位置。在 Web Service 中,通用描述、发现和集成 (UDDI) 规范用来定义 Web 服务的发现和发布方法。利用 UDDI 可以创建一个基于分布式的注册中心,它维护着一个存储 Web 服务信息的公共存储库。服务请求者通过查询注册表,期望能够找到一个适用于他们需求的服务。若找到了一个这样的服务,服务的请求者就可以绑定该服务,然后使用该服务。

当组织上独立、各自运行着工作流管理系统的企业或机构之间进行业务过程的集成和协作时,为了创建、发布和执行工作流实例,其工作流引擎之间必须通过互操作相互提供工作流执行服务。当基于 Web Service 实现异构工作流管理系统之间的互操作时,也需要将各工作流管理系统提供的工作流执行服务用 WSDL 文档描述,然后基于 UDDI 规范发布。当此工作流执行服务被其它工作流引擎发现并为之绑定后,就可以对后者提供工作流执行服务了。通常需要一个工作流管理系统提供的基本的工作流执行服务如下:

- 创建过程实例
- 设置/获取过程实例的属性
- 启动/终止过程实例的执行
- 获取过程实例的状态

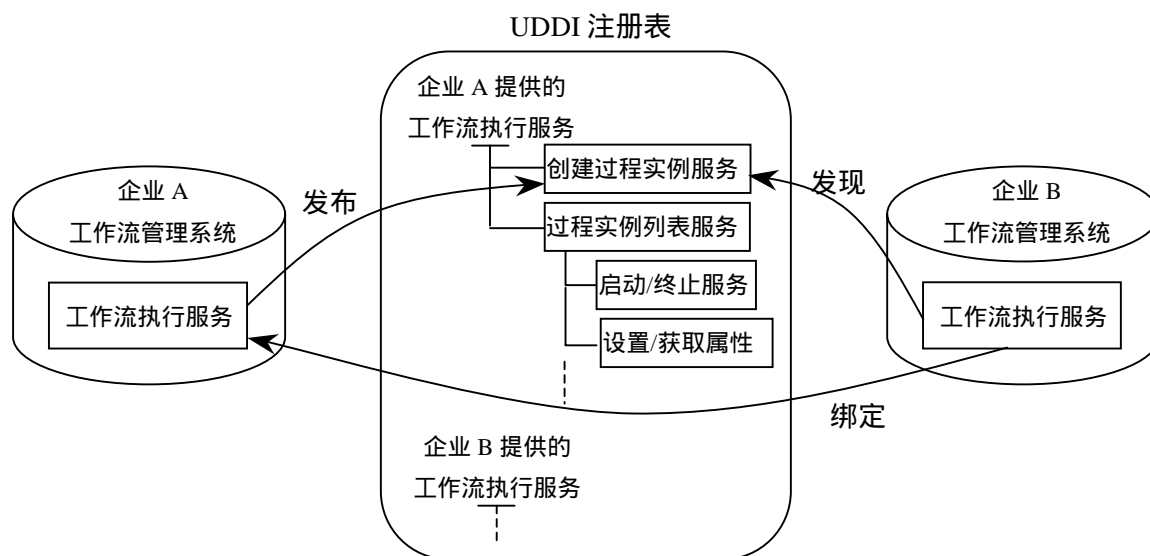


图 2 基于 Web Service 提供的工作流执行服务

这些工作流执行服务可灵活采用(1)式中相应的调用语义,并用 Web Service 实现。异构工作流管理系统基于 Web Service 相互提供工作流执行服务的过程如图 2 所示。

5 结论

Web Service 规范具有开放、标准化的特点,并获得了业界广泛支持。Web Service 使用基于文本的协议,与 CORBA、消息队列技术等通信中间件支持的分布式应用相比,XML 文档的解析将使基于 Web Service 分布式应用的性能降低。但是,随着网络和处理器速度的提高,这些影响将不再成为问题。Web Service 提供的基于 XML 文档和 SOAP 的抽象机制,为分布式、异构系统之间跨平台集成以及系统之间提供透明的服务提出了真正的解决方案。Web Service 完全可以满足异构工作流管理系统之间信息集成的需求,Web Service 支持的交互模式完全可以实现异构工作流管理系统之间互操作所需的调用语义。基于 Web Service 实现的异构工作流管理系统的集成和互操作不受特定软件供应商提供的分布式应用解决方案的影响,易于为广大的用户接受。

参考文献

- [1] 曾宪文, 时间竞争与缩短订货配送周期研究, <http://www.logistics-ec.com/zxw/1.htm>, 2000.
- [2] 范玉顺, 工作流管理技术基础—实现企业业务

过程重组、过程管理与过程自动化的核心技术,清华大学出版社,施普林格出版社,2001,4

- [3] Jablonski, S.; Bussler, C.: Workflow Management - Modeling, Concepts, Architecture and Implementation, International Thomson Computer Press, 1996.
- [4] Otte, R.; Patrick, P.; Roy, M. 著,李师贤等译校, CORBA 教程——公共对象请求代理体系结构,北京:清华大学出版社,1999.
- [5] Kenn Scribner, Mark C. Stiver 著,刘勇,宋继武译,应用 SOAP 实现.NET XML Web 服务,北京:人民邮电出版社,2002