

J2EE 工作流的交互性研究与实现

苏 阳, 朱 明

(东华大学 计算机科学与技术学院, 上海 200051)

摘要:传统的工作流管理系统无法解决分布异构环境中数据传递与表示的问题,难以适应我国信息化建设中对于原有系统进行流程整合的需求。研究了 Web Services、XForms、EJB 以及 JBPM 开源工作流项目等技术与标准,并结合使用 Façade 和 Delegate 模式提高工作流的可扩展性,在广泛的研究与技术实践的基础上提出了一种交互接口设计模型。在该模型的应用中证实了传统工作流必须结合 Web Services 等技术,对交互性进行改造,才能满足我国企事业单位流程整合与管理的需求。

关键词:工作流;交互性;XForms;Web Services;J2EE

中图分类号:TP311.5

文献标识码:A

文章编号:1673-629X(2008)06-0086-04

Research and Realization on Interactivity of Workflow Based on J2EE

SU Yang, ZHU Ming

(School of Computer Science & Technology, Donghua University, Shanghai 200051, China)

Abstract: The traditional workflow management system can't solve the problem of transfer and expression of data in distributed and different environment, it's difficult to satisfy the requirement of integration of business process in old systems. The technology or standard of Web Services, XForms, EJB and JBPM are researched in this paper, and the design model of façade and delegate are used together to improve the extensibility of workflow management system, a framework of interactive interface integrated with Web Services and XForms is put forward in this paper. The application of the framework approve that for satisfy the requirement of flow integration and management in our country, traditional workflow should rebuild interactivity that use Web Services, etc.

Key words: workflow; interactivity; XForms; Web Services; J2EE

0 引言

当前我国企业和政府的信息化建设加速发展,原有业务系统中流程的概念已经无处不在,对于流程管理以及整合原有系统的需求日益突出。因而对于工作流相关技术的研究与应用也越来越广泛。然而大量原有信息系统往往用不同的技术构建并运行在不同地域的不同服务器中,这决定了传统工作流管理系统必须有效解决异构分布环境中数据传递与表示的问题。

文中针对基于 J2EE 技术构建工作流管理系统时所遇到的在分布异构的复杂环境中交互性无法达到需求的问题,研究了如 Web Services 和 XForms 等当下主流和未来将成为发展趋势的多种技术与标准,并在此基础上提出有借鉴价值的工作流交互模型。

1 交互接口概念模型与 Web Services

工作流的运行通常是整个组织内的一系列协作组成的。因此,工作流系统必须考虑到各种外部交互问题。工作流管理联盟(WfMC)于1995年提出了工作流管理系统的体系结构模型——工作流参考模型(Workflow Reference Model),该体系结构中定义了以下5种基本接口^[1]:

(1)流程定义接口:流程的部署和流程的定义,流程定义的运行时修改,流程定义版本控制等。

(2)工作流客户程序接口:用于工作流客户端应用访问工作流引擎和工作列表。

(3)外部应用程序调用:业务系统与工作流系统交互的主要接口之一。

(4)外部工作流协作:涉及独立的工作流系统之间的案例交换。在本系统中,暂不考虑(一般工作流引擎,例如 JBPM 在这方面的标准兼容性影响较大)。

(5)工作流管理与监控接口:用于系统管理应用访问工作流执行服务。

以上5种接口的划分是逻辑上的,工作流引擎处

收稿日期:2007-09-19

作者简介:苏 阳(1980-),男,硕士研究生,主要研究领域为工作流管理系统、并行计算机系统、J2EE;朱 明,副教授,主要研究领域为并行计算机系统、人工智能等。

于核心位置,它对外提供这些接口的实现,在这里称其为 WAPI(Workflow APIs),当拥有接口 2 或 3 中的一个, workflow 引擎就可以访问外部资源,如果系统中有多个引擎就需要实现接口 4 使它们可以协同工作^[2]。在本模型中将这些接口的 WAPI 用 Façade 模式封装在一个 Façade 类中作为门面提供给外部系统。该交互接口的适应性较弱,如果 workflow 及接口以 java 技术实现而业务系统在不同的 java 虚拟机中或是用其他技术构建的 C/S 系统则无法直接调用。而 Web Services 不断增长的可用性为解决这种系统集成与互操作问题提供了新的方式^[3]。

Web Services 是 SOA(面向服务体系结构)的一种具体的实现方式,SOA 的核心就是将多个系统连接和整合起来,将系统作为服务对外部开放,与外部进行联系,实现系统与系统(服务与服务)的相互通讯^[4]。Web Services 的主要优势在于异构平台的互通和普遍的通信能力。在 workflow 系统中可以把实际应用按单位划分成 service, service 实际上是一个流程,而其中的一个流程单元有可能要和另外的 service 产生关系。

Web Services 以简单对象访问协议(SOAP)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。而这些协议与标准均建立在可扩展标识语言(XML)和超文本传输协议(HTTP)等 Internet 通用协议的基础上,这使得应用 Web Services 相关实现技术(如 JBPM BPEL Extension)包装 workflow 管理系统各个接口的 Façade,使其在分布异构环境下交互成为可能。交互接口的概念模型如图 1 所示。

操作性交给 workflow 系统内部。

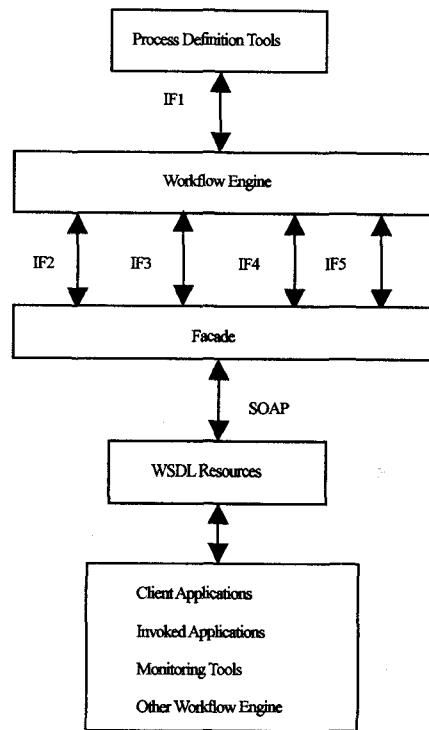
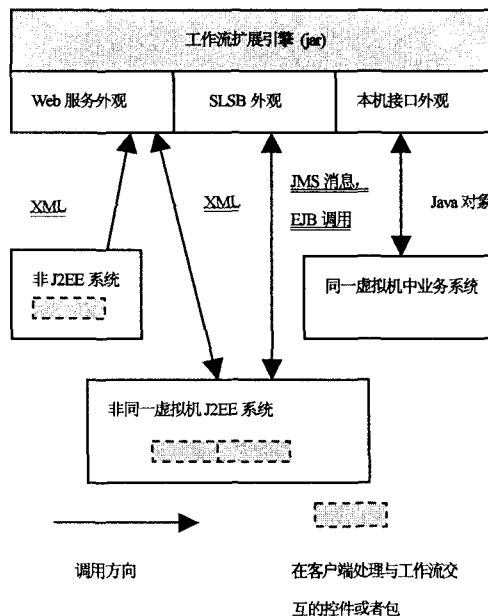


图 1 交互接口概念模型



注:SLSB 为无状态 Session Bean

图 2 工作流系统与业务系统之间的交互

经过上述分析,处理各种应用系统之间的交互,得到的技术架构有以下两点建议:

1) Web 服务在不同的系统之间传递 XML 格式的数据。在异构的应用环境下,有的 workflow 系统可能要处理到不同的性质、结构的应用系统之间的交互,因此采用 Web 服务的方式处理各种系统,是一种优先的选择。

2 交互接口的多种设计方式

企业原有的业务系统中,往往既有以 Delphi 等技术为基础的 C/S 系统,也有 J2EE 应用等 B/S 系统。在技术异构的环境下,workflow 系统都可能被应用。

同时,原有的各种 J2EE 系统也经常因为使用的中间件不同而决定了 workflow 系统不能依赖于任何具体中间件,而是需要普遍支持主要的中间件产品,包括 IBM Websphere, BEA WebLogic, Sybase EAServer 等。

上面提出的模型并非唯一的选择,在有的情况下会因为技术复杂度高而不够经济。在不同应用环境中接口提供方式可以不同,应当根据实际情况选择最适合的方式,而作为一个成熟的 workflow 产品也应当考虑提供多种方式实现的接口。图 2 中给出了 3 种可供选择的方式,无论哪一种接口,都应该在客户端应用中提供本地调用方式的接口服务。把与 workflow 服务器的互

2)应用 XForms,其优点在于设计的 XForms 表单是基于 XML 的,能够在 J2EE 和非 J2EE 系统上不需修改得到解释;同时,XForms 使用 HTTP 协议提交 XML 格式数据,XForms 能够直接解析来自 Web 服务的结果并展示。XForms 在工作流客户端(即业务系统)中负责数据展示和与工作流的数据交互,将最大限度隔离工作流与基于工作流的业务系统的耦合。

3 数据展现与 XForms

在大型应用系统中,往往存在多种同构或者异构系统之间的数据交互。作为横跨整个企业或者机构组织的工作流系统,必然需要提供面对来自各种应用系统的数据交换能力^[5]。

流程中数据分为流程相关数据和应用数据。各种应用数据需要在工作流和各个系统之间传递、展现。考虑到业务系统的各式各样,数据传递要符合下面的需求:

1) 支持可能参与的各种技术平台(J2EE, Java Swing, Delphi 等),即传递中的数据与平台无关。

2) 在流程变化时,尽可能减少业务系统的变动,即应该将数据交互的工作在工作流系统内部去完成。

3) 在流程变化时,流程中应用数据的展现不应该导致应用程序的修改。

应对这样的需求,使用 XML 来交换数据,是一个优先的选择。在这种情况下,服务器端通常使用 Web 服务或者是采用其他方式提供 XML 格式的数据。而在显示层面,工作流系统应提供能在以上几种技术平台上展现数据的接口,并且负责数据的展现。使用 XForms 来作为用户接口,是一个不错的选择。

XForms 是 W3C 下一代 Web 表单的标准。通过将传统的 XHTML 表单分解为三部分:XForms 模型(XForms Model)、实例数据(Instance Data)、用户接口(User Interface),它分离表示与内容,支持重用,提供强类型,减少与服务器往返通信,同时提供了设备独立性,减少脚本编程。XForms 不是一种独立的文档类型。它只能集成到其他标记语言中,例如 XHTML 或者其他合适的宿主语言。XForms 与 HTML 表单相比,有下面几个显著的区别:

(1)基于 XML。

(2)XForms 不再从标记语言控件中收集,而是直接传递 XML;XForms 表单的初始化数据和提交的数据都是 XML。

(3)XForms 不仅可以应用在 XHTML 中,而且可以应用于其他合适的标记语言中。

由于 XForms 的实例数据可以链接至任意返回

XML 格式数据的 URL,XForms 还具有了如下的功能:XForms 实例数据可以是远程计算机上的 XML;XForms 可以将数据发送和接收 Web 服务。

在支持 XML 方面,与 JSF 和 Freemake 等技术相比,XForms 对 XML 具有十分强大的支持。XForms 本身基于 XML,使用 XML 作为主要的数据传递载体。XForms 直接支持 XML 格式的数据,可以直接解析来自 Web 服务或者返回 XML 的 URL 地址的 XML 数据,可以将 XML 格式的数据提交到任意 URL 地址。

同时,XForms 支持强类型的数据交换,可以对交换数据进行较为严格的控制。

基于 XForms 的用户接口系统,可以与各种应用系统进行数据交互,可以支持多种数据来源,进行统一展示。

4 一种交互接口设计模型

综合以上的研究与分析,得出了一种交互接口的设计模型,如图 3 所示。

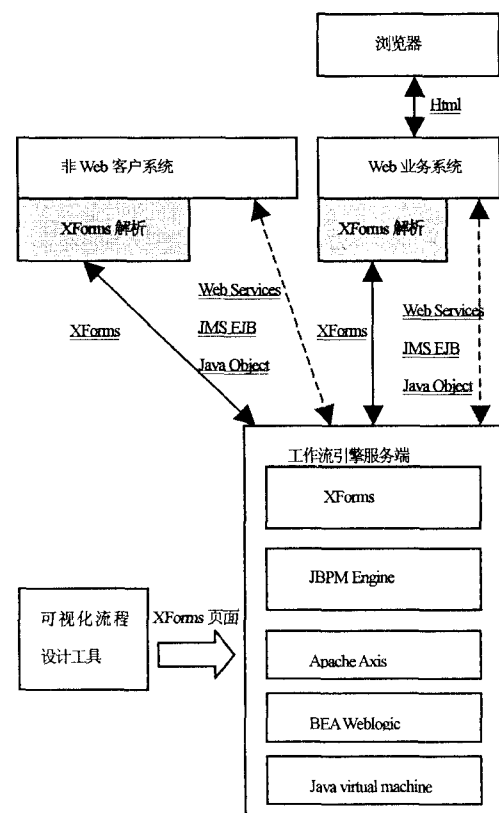


图 3 一种交互接口设计模型

JBoss JBPM 是一个优秀的开源工作流引擎,Apache Axis 总体上是一个 SOAP 引擎,它还可以作为 Web 服务器(如图中的 BEA Weblogic)或 Servlet 引擎(如 Tomcat)的一个可插入服务,并且可以扩展地支持 WSDL。业务系统与工作流之间使用 XForms 或者其他数据交互方式。需要指出的是,当使用 EJB 或者

Web Services 时, workflow 系统应该有专门在客户端处理交互工作的 API 包(或者控件), 业务系统直接使用该 API 包(或者控件), 而不需要执行处理 Web 服务或者 EJB 调用等^[6]。

XForms 并非唯一的选择, 还有其他技术可以应用到 JBPM 中来实现快速原型开发和与业务系统进行交互(数据交换和状态展示)。有些工作系统(如 Runa)支持 FreeMaker, HTML 表单等多种状态展示方式。而 JBPM-3.2 控制台中则引入了 JSF 作为工作流程中的状态展示方式。总的来说, 各种系统使用这些表现层技术, 主要目的在于将流程状态与用户接口(HTML, Freemaker, JSF)对应起来, 从而向终端用户展现流程状态, 同时接收输入。

用 XForms 进行 workflow 引擎服务和业务系统之间的交换, 主要有以下几个优势:

(1) 使用可视化设计工具可以帮助业务人员设计业务流程, 或者可以帮助开发人员快速开发系统原型。

(2) 使用 XForms 展现与流程相关的数据并处理数据交互, 从而将这些数据交互工作交给 workflow 系统内部处理, 而不需要业务系统干预。当流程变化时, 不会带来业务系统的变化。图 3 中, 实线代表 XForms 与 workflow 之间的交互; 而虚线是不使用 workflow 的情况。

(3) XForms 直接访问业务系统数据, 可以通过 URL 的形式, 也可以通过 Java 接口的形式。其中, URL 形式可以是 XML 文件, 也可以是返回 XML 的服务器 URL, 例如 Servlet 等。

(4) Web 服务和 EJB 访问: XForms 插件提供了对于 Web 服务的访问, 这对于在任何需要的时候在您的系统中支持 EJB 和 Web 服务等, 是有益处的。

(5) 数据验证: XForms 提供了便利的数据验证机制, 可以简化业务系统客户端和服务器端的数据验证工作。

(6) XForms 中集成了 AJAX 技术以提升用户体验, 从而在提升软件可用性的同时, 也节省了大量开发。

使用 XForms 的劣势在于:

1) 将加大 workflow 管理系统开发任务和以及本身复杂度。

2) workflow 客户端开发成本和难度很大。例如将 workflow 嵌入到 JSP 中存在较大的难度。

结论: 可以采取分步实现的方式来完成这个目标。可以在先期使用简化的 XForms 代替完整功能的 XForms。一个建议的方案: 使用 XML 传递数据, 同时, 对于应用数据建立展现模型(简化的 XForms), 利用现有网络协议传递给业务端。在客户端, 根据该展现模型展现数据。

5 结束语

传统意义上的信息系统没有把过程管理和应用软件区分开, 业务过程被隐藏在系统中难以识别。随着我国信息化建设的加速发展, 对于在各个时期构建的信息系统进行业务流程的整合与管理已经提上日程。workflow 技术为适应复杂的应用环境必然要借鉴日趋成熟的 J2EE、Web Services、XForms 等技术标准。文中在广泛的研究与技术实践的基础上提出了一种交互接口设计模型, 具有较强的实用价值。在目前的基础上, 还将对 JBPM BPEL Extension 做细致的研究与技术实践, 降低实现 Web Services 的技术复杂度, 并在 Façade 下面提供多个 delegate 来提高 workflow 的可扩展性。

参考文献:

- [1] Workflow Management Coalition. The Workflow Reference Model[S]. 1995.
- [2] Workflow Management Coalition. Workflow Management Application Programming Interface (Interface 2&3) Specification[S]. 1998.
- [3] Jang Jinyoung, Choi Yongsun. uEngine: Web service based Workflow Management System[EB/OL]. 2004-02. <http://uengine.sourceforge.net/files/uEngine-web-service-support.pdf>.
- [4] Graham S. Building Web Services with Java: Making Sense of XML, SOAP, WSDL, and UDDI[M]. 刘晓晖, 等译. 北京: 机械工业出版社, 2003.
- [5] Workflow Management Coalition. Workflow Standard Interoperability Abstract Specification[S]. 1999.
- [6] 夏冬, 白树仁, 邓惠建. 基于 J2EE 的 workflow 管理系统模型[J]. 计算机工程与科学, 2006, 28(3): 123-133.

(上接第 85 页)

- Quality of Service[M]. [s.l.]: Morgan Kaufmann, 2001.
- [3] Dysan D M. QoS & Traffic Management in IP&ATM Networks[M]. New York: McGraw-Hill Companies, Inc, 2000.
- [4] Charidar M, Naor J, Schieber B. Resource Optimization in QoS Multicast Routing of Real-Time Multimedia[J]. IEEE/

- ACM Trans on Networking, 2004, 12(2): 340-348.
- [5] Marsan M A, Bianco A, Giaccone P, et al. Packet - Mode Scheduling in Input - Queued Cell - Based Switches [J]. IEEE/ACM Trans on Networking, 2002, 10(6): 666-678.
- [6] 钱光明. 基于业务的多优先级队列区别服务方案[J]. 计算机工程与应用, 2006, 42(10): 118-120.