

动态即时的协同工作平台设计与实现^{*}

徐光侠^{1,2} 祝伟华³ 杨丹³

(重庆邮电大学软件学院 重庆 400065)¹ (重庆大学计算机学院 重庆 400030)²

(重庆大学软件学院 重庆 400030)³

摘要 针对传统的信息交流模式是静态的工作流方式,提出了一种动态即时的协同工作信息交流平台。文章以办公系统为背景,将 Java 技术、XML 技术、工作流技术及消息传递中间件技术有机结合,实现了用户在协同办公平台内的在线信息交流。实际工程应用效果表明,该系统的设计是成功的。

关键词 动态协同,工作流驱动,中间件技术

Design and Realization of Coordination Work Platform Based on Dynamic and Instantaneous

XU Guang-Xia^{1,2} ZHU Wei-Hua³ YANG Dan²

(School of Software, Chongqing University of Posts and Telecommunications, Chongqing 400065)¹

(College of Computer, Chongqing University, Chongqing 400044)² (School of Software Engineering, Chongqing University, Chongqing 400030)³

Abstract Aiming at the mode of traditional information communication being a static workflow manner, the paper presents a kind of information communication platform that can be worked in dynamic and instantaneous coordination. Based on office automation system, making an organic combination among the techniques of Java, XML, Workflow and Middleware of message transfer, the information communication on line is realized among users in the office platform of synergetic work. The application effect of practical engineering shows that the design is successful.

Keywords Dynamic coordination, Workflow drive, Middleware technique

1 引言

信息社会对信息交流方式的要求越来越高,传统的固定工作模式已不适应即时获取信息满足协同工作的需要。本文以办公系统为背景,将 Java 技术、XML 技术、工作流技术及消息传递中间件技术有机结合,构建了一个动态即时协同工作的信息平台。该平台基于 XML 协同动态工作流驱动和可视化的工作流建模工具(X-WorkFlow),满足了用户可视化流程自定义的要求;基于 XML 的协同消息服务驱动,实现了用户间在协同办公平台内的在线信息交流,满足了人们对信息即时协同的工作需要。

以下就协同办公平台的设计与功能实现问题进行讨论。

2 工作流驱动及 X-WorkFlow 的设计

2.1 引擎的结构

工作流引擎是工作流驱动的核心部分,工作流引擎主要包含下列功能:解释流程定义;生成过程实例;控制流程实例的创建、激活、挂起和终止等,实现流程活动导航;引导流程活动的进行,包括顺序和并行操作期限调度工作流相关数据解释等;确定需要用户处理的工作项,且提供用户交互的接口;维护工作流的控制数据、工作流的相关数据以及与用户或应用程序传递工作流的相关数据;调用外部应用程序和连接工作流相关数据的接口等。

工作流引擎^[1]的过程流转系统可以表示为一个三元组 $\langle D, I, M \rangle$,其中 D 表示过程定义, I 表示过程实例, M 表示过

程流转控制系统。过程实例是过程流转的主体,它是在控制系统的控制下由过程定义模型复制并添加相应的状态信息生成的。控制系统控制多个实例,监控和修改实例的状态,如图 1 所示。

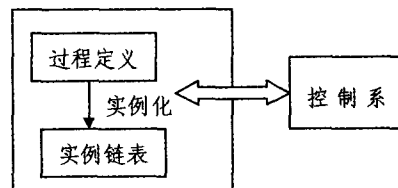


图 1 过程定义、过程实例与过程流转控制系统的关系

为了驱动工作流程,可以采用基于邮件或基于数据库的方式。采用前者,可以充分发挥电子邮件系统在广域网环境下的数据分发功能,但整个系统将运行于一种松散耦合的模式下。而采用后者,所有的数据都保存在某种类型的 DBMS 中,过程的执行实际上就是对这些数据的查询与处理。由于工作流是在协作的基础上驱动,集中式的数据库管理将影响系统的扩展性。

作为一种尝试,本系统采用基于 XML 的构造,数据的交换、存储都是基于 XML 的,过程的执行是对 XML 数据的查询和处理。

2.2 基于 XML 的工作流过程模型 DT-XPDL

1994 年 WfMC 推出了一个最小元模型的文本描述语言,即 Workflow Process Definition Language(WPDL)。WPDL 的核

^{*}基金项目:重庆市自然科学基金(CSTC,2005BA2002)。徐光侠 硕士研究生,讲师,主要研究方向:软件工程,企业信息化等;祝伟华 博士研究生,副教授,主要研究方向:软件工程、图形图像等;杨丹 博士,教授,博士生导师,主要研究方向:企业信息化及企业应用集成,软件过程技术及管理。

心部分是工作流过程定义。工作流过程定义包括一个或多个工作流过程活动,WfMC 的主要目的之一是支持异构 workflow 管理系统与产品之间的互操作性。2001 年 WfMC 发布了一个语言 XPDL^[2] (XML Process Definition Language, XPDL)。

参考 WfMC 推出的工作流过程定义语言(WPDL)/XML 过程定义语言(XPDL),结合办公中公文流转业务特点,提出采用面向公文流转系统的基于 XML 的过程定义语言 DT-XPDL (Document Transfer based XML Process Definition Language),以实现工作流过程引擎。

2.3 动态工作流引擎的实现

(1) 过程实例的状态

过程模型^[3]的过程实例有 4 个状态:初始、运转、暂时、结束。4 个状态在工作流控制系统控制下相互转移,如图 2 所示。

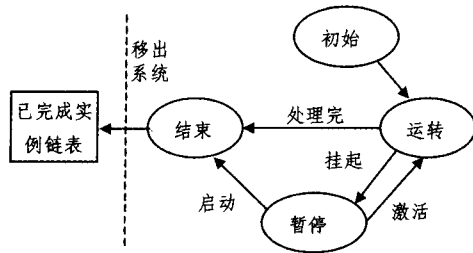


图 2 过程实例状态转换图

由图 2 可见,对完成的过程实例,将被控制系统移出 workflow 运转系统,这样可保证系统中完成的实例不再占用系统资源,避免了对活动实例处理效率的影响。

(2) 实例的管理:轮询队列

引擎通过轮询控制状态。对于过程实例引擎用 XML 的元素对其进行描述,在 DOM 树的构造中,这些实例处于树的最底层,相当于叶子结点。引擎通过元素名对它们逐次访问,当用户登录进入系统时,引擎会将待处理的元素信息返回给用户,如果没有该用户要处理的信息,引擎不采取任何行动。当用户进入系统后,如有待处理的实例,该实例将进入新的待处理队列,这时由用户发送处理请求,直到实例处理或用户推出系统,否则实例将挂在待处理队列中等待处理。

(3) 主动式的推

为了实现主动式的交互协调,实例运转时,可通知客户处理事件,由客户的行为驱动下一步的运转方向和状态,在运转中改变工作流程,从而实现“动态工作流”,所以必须有某种机制引擎。在需要时主动通知客户进行下一步处理,由于协作是基于 Web,并使用了 XML 技术,为了在浏览器中解析 XML,就必须利用 Applet, Servlet 和 JSP 相结合的方法。服务器的 Servlet 和浏览器端的 Applet 通过 XML 在 Web 上交换信息,当服务器有消息要通知客户时,Servlet 向 Applet 发送 XML 格式的信息,Applet 通过用浏览器执行 JavaScript 解析 XML 获得所需信息,并以窗口消息的方式通知客户,如图 3 所示。

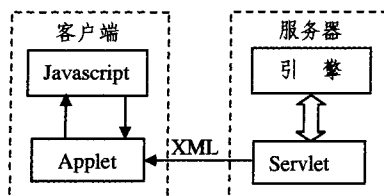


图 3 主动式交互的机制

客户与服务器的通信都是自定义的 XML 格式信息,利用 XML 作通信格式,不仅易于扩充,在当前基于 Web 的实现中,还具有易于跨越网络代理和防火墙的优点。

(4) 动态工作流引擎实现

动态工作流引擎是整个 workflow 驱动中的核心的部分,负责读取由可视化流程设计器生成的 XML 流程定义文件,并对它进行解析、运行。

当用户需要创建流程实例时,流程引擎首先会创建该流程对应的 XML 描述文件的副本,作为整个流程运行中的参考,流程引擎的 XML 解析器将进行工作,负责把设计期的流程和对象转化为运行期的流程实例和对象。具体来说就是把基于 XML 存储的信息,转化为基于数据库存储的信息,在这个过程中,将使用到数据持久化技术,以实现数据库层的隔离。流程实例创建后,它将在流程引擎的控制框架下被动运行,流程引擎负责流程的调度、接收用户端的输入,根据用户的输入和流程规则进行流程分支的选择,按照条件进行流程的跳转、暂停、恢复、中断。

流程引擎分为 4 个包:engine 包是流程引擎的核心包,负责读取流程配置信息、创建和销毁流程实例,并定义流程对象基类、流程监听接口等流程核心部分;designtime 包是设计期对象包,复制解析设计期的 xml 文件,并分解为相应的对象,以方便持久化到数据库;runtime 包是运行期对象包,它负责对运行期对象进行管理,并实现流程的重要功能,如动态解析流程参与者、动态路由选择、流程事件监听、用户脚本支持等;organization 包是组织结构包,定义了流程运行所需的参与者类型,如:员工、角色、部门、小组等。该包以接口的形式存在,以适应 workflow 系统能跨系统运行,相关应用系统只需要实现这些接口,该 workflow 系统就可以运行。以下对其中关键的对象原型设计^[4]作些讨论。

2.4 动态流程设计器(X-WorkFlow)的实现

为了让使用者能更好地使用协同工作平台,并根据自身的需要对流程进行持续的改进,有必要让使用者,而不是开发者对流程进行重新设计。基于该需要,让 workflow 系统具有完全图形化、可视化^[5]的操作界面是必要的。

可视化流程设计至少有如下要求:1)由于协同办公平台是基于 Web 的实现,为了同整个平台保持兼容,用户更易于使用,因此可视化流程设计也必须是采用纯 Web 技术进行实现;2)支持所见即所得的设计,采用拖拉的方式对流程进行设计和修改,隐藏技术细节,让用户能更专注于自己关心的流程;3)能生成标准化的 XML 流程定义文件,在系统的可视化流程设计器中,采用纯 Web 技术进行开发,而不采用任何类似于 ActiveX 插件的本地化技术,这样能更好地兼容各种浏览器和操作系统平台;4)在支持所见即所得方面,设计器可采用 Html+JavaScript+XML 的技术进行开发,以完全实现流程元素的动态创建、拖曳、粘接、自动验证等功能。

可视化流程设计的结果是一个标准的 XML 文档,它是对所设计流程的 XPDL 描述。该 XML 文件将作为流程引擎创建并运行流程的入口,参与流程运行的全过程,由动态 workflow 引擎进行解释并执行。设计器 JavaScript 类组成由于篇幅有限,这里不作具体详述。

根据以上分析和设计,系统实现了一个基于 XML 的动态工作流的可视化设计器^[6] (X-WorkFlow, XWF)。用户可按需要生成流程,无需编程。

2.5 流程监控的实现

流程跟踪、监控是流程引擎的外围模块,主要是给用户一个操作接口,流程参与者通过这个接口,可以实时地、可视化地对流程的流转情况进行跟踪,并能对处理情况进行查询,流程管理者还可以在流程图上对流程进行管理,包括对参与者的管理、对流程流向的控制以及改变流程状态等。功能有:经办流程跟踪查询;经办流程统计;流程超期提醒;所有流程统计;流程监控等。

2.6 可视化流程页面编辑器的实现

可视化流程页面编辑器是可视化流程设计器的一个辅助设计工具,它的作用是让用户可视化地设计在流程中需要提供使用的相关表单界面。其结果将作为流程设计器的相关属性,在运行时由控制框架调用,并展现给用户。可视化流程页面编辑器其实是一个基于浏览器的可视化 HTML 编辑器,其输出结果是一段 HTML 代码。其功能为:文件上传功能;上传文件类型限定功能;图文混排功能;所见即所得功能;表格功能;表单及常用控件支持;Word 文件转换功能等。

3 协同消息服务中间件 SCMS 的实现

3.1 消息传递机制

任一成员通过系统信息发送,只需直接调用 JMS 规范提供的接口函数。SCMS 系统负责将该信息按照通信协议所定义的消息格式进行打包,然后将打包后的消息放入消息发送队列,并释放 1 个信号。该信号触发发送线程,使其从消息发送队列中读出消息,通过消息通道代理利用传输协议和物理连接将消息发送到远程系统。SCMS 是根据消息的地址绑定接收对象,通过通道代理调用接收对象的接口函数,将消息放入接收对象的消息接收队列。在发送方完成消息发送的同时,接收方也完成消息接收。接收方只需直接从自身消息接收队列中读出消息、解包,并根据消息的类型进行相应的处理。消息传送有三种形式:一对一、一对多、多对多。在 JMS 规范中,一对一采用点对点模式,一对多和多对多采用发布/订阅模式。SCMS 的工作模式参照了 JMS 中的点对点模式和发布/订阅模式。

点对点模式:点对点传输消息是建立在消息队列基础上的,每个客户端对应一个消息队列,客户端发送消息到对方的消息队列中,从自己的消息队列读取消息。队列将发送给它们的消息一直保留,直到消息被阅读或消息到期。

发布订阅模式:发布订阅模式(Pub/Sub)是客户将消息提交给某个主题,发布者和订阅者通常是匿名的,并能够动态发布或订阅消息。Pub/Sub 系统必须保证某个主题的所有发布者(Publisher)发布的信息准确无误地发送到该主题的所有消息订阅者(Subscriber)。主题把消息分发给当前的调用者之后就不再保留这些消息。

3.2 点对点模式的具体实现

SCMS 利用共享内存提供模拟发送队列和接收队列,提供核心和应用之间的一个消息传输通道,对发送消息和接收消息起控制和缓冲作用,如图 4 所示。

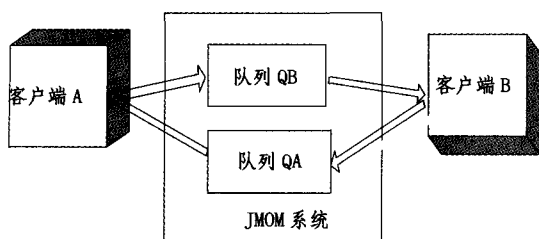


图 4 消息队列的具体实现

客户端 A 对应队列 QA,客户端 B 对应队列 QB。客户端 A 要发消息给客户端 B,则客户端 A 把消息发到客户端 B 的队列 QB 中,客户端 B 从队列 QB 中读取消息。同样,客户端 B 要发消息给客户端 A,则客户端 B 把消息发到客户端 A 的队列 QA 中,客户端 A 从队列 QA 中读取消息。

4 SCMS 基于 Web 的客户端实现

由于该协同办公平台是基于浏览器的应用,但浏览器又不直接支持 JMS 的相关服务,因此在浏览器端采用 Java Applet 技术实现了一个符合 JMS 规范的无界面消息收发小应用程序,并使用 JavaScript 脚本对其进行封装,这样 Web 程序就可以像操作普通对象的方式来操作消息对象,使用基于互联网的消息服务。浏览器端的消息服务主要由 3 个对象组成:Message、MessageServer 和 MessageClient,这些对象和 SCMS 消息服务的其他部分的关系如图 5 所示。

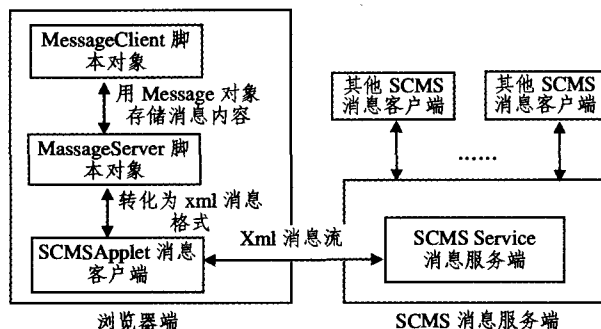


图 5 对象和 SCMS 消息服务关系

Message 对象是对 XML 消息对象的一个封装,以方便 JavaScript 对消息进行操作,同时它也是 MessageClient 和 MessageServer 之间传递的标准对象。MessageServer 对象主要是消息转发,即接收服务器端消息,并转发给已注册的本地窗口,同时接收本地待发送的消息,并发送到服务器端。MessageClient 对象负责接收本地服务器转发的消息,并提供给应用程序相关的接口进行消息的收发,该对象是浏览器端程序直接使用的对象。

结束语 上述在给出了基于动态即时的协同办公系统总体框架后,着重分析了在该系统中如何实现动态 workflow 定义、流转控制及在线交流。实现了使用 java 开发基于 XML 的动态 workflow 引擎和可视化流程器(X-WorkFlow),以及采用基于 java 的消息服务中间件(SCMS)的在线交流平台。

参考文献

- 1 窆万春,李东波,张世琪.基于对象和实例互操作行为模型的工作流研究[J].计算机学报,2001(2):197~201
- 2 周航滨,夏安邦,张长昊.基于 Web 服务的跨企业信息集成框架计算机集成制造系统——CIMS[J].计算机学报,2003(1):1~5
- 3 柴晓路,良宇奇. Web Services 技术、架构和应用[M].北京:电子工业出版社,2003.126~127
- 4 Leymann F, Altenhuber W. Managing Business Processes as an Information Resource. IBM Systems Journal, 1994,33:326~348
- 5 沈军营,黄进,严隽琪.虚拟企业中的工作流技术研究.计算机集成制造系统——CIMS[J],2000(2):7~10
- 6 Glance N S, Pagani D S, Pareschi R. Generalized Process Structure Grammars (GPSG) for flexible representations of work. In: M. Ackerman, ed. CSCW'96: Proceedings of the Conference on Computer Supported Cooperative Work. Boston, MA, 1996. 180~189