

分布式 workflow 执行服务的设计与实现

李红信 范玉顺

清华大学自动化系 (北京 100084)

摘要 workflow 执行服务的实现方案是实施 workflow 技术最为关键的一个方面, workflow 管理系统的应用特点决定了它必须与各种分布的、异构的应用通信或与其集成。为了给用户透明的工作流执行服务而隐藏 workflow 执行服务实现的复杂性, 本文提出一种具有 3 层层次结构的工作流执行服务实现方案, 其中的分布式通信服务层由通信中间件实现, 使开发者无需关心程序间通信的复杂性, 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。

关键词 workflow 执行服务 分布式计算

Design and Implementation of Distributed Workflow Execution Service

Li Hongxin Fan Yushun

(Department of Automation, Tsinghua University, Beijing 100084)

Abstract: The implementation of workflow execution service is critical to the implementation of workflow management technology in an enterprise. Workflow management system has to be integrated with various distributed and heterogeneous applications or has to communicate with them. In order to provide a transparent workflow execution service to users and separate the complicated implementation details from them, a three-layer architecture of workflow execution service is proposed in this paper, the communication service layer is implemented with communication middleware. The designers are released from complicated inter-program communication and concentrated on optimizing the business process, therefore, the efficiency of development is enhanced.

Keywords: Workflow execution service, Distributed computing

1 引言

企业是通过相关的业务过程向外界提供产品或服务的, 所以企业业务过程的效率和有效性成为企业赢得竞争的关键。目前市场竞争的特点是基于时间的竞争 (Time-based Competition), 相对于成本、质量而言, 时间已经成为竞争的焦点。

为了提高业务过程的效率和质量, 一方面, 企业普遍采用计算机应用系统、计算机网络等信息技术支持业务过程。另一方面, 企业及研究机构从组织和管理的角度对业务过程的运作方式进行了根本性的反思, 从面向功能的管理转移到面向业务过程的管理。这两方面综合作用的结果就是企业业务过程重组。 workflow 管理技术就是在这种应用背景下

逐渐发展起来的, 它是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。

简言之, workflow 就是全部或部分企业业务过程的自动化执行, 支持业务过程自动化执行的应用系统即 workflow 管理系统。这里的“自动化”指的是业务过程中要执行的任务从一个过程步骤向后续的过程步骤的传递是自动的, 处理结果的提交或向指定数据库的存储也是自动的。 workflow 管理系统通过在过程参与者的工作项列表中突出要执行的工作流实例, 通知过程的参与者, 从而驱动着 workflow 的执行, 因此, workflow 管理系统具有能动性。图 1 为一个简单的贷款业务在 workflow 管理系统支持下的执行过程。

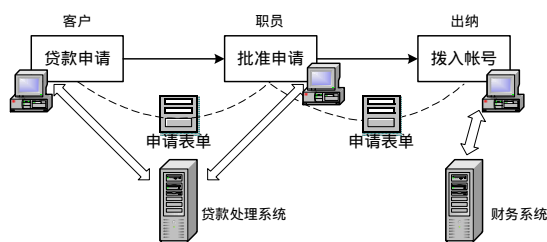


图 1. 贷款工作流的执行过程

2 工作流执行服务

工作流管理系统支持的是企业或企业之间的业务过程，其用户分布在不同的地理位置，工作在不同的软、硬件平台之上，工作流的执行需要集成或调用各种应用系统，因此，工作流管理系统运行在一个高度异构及分布式的环境中。但是，工作流管理系统的用户关心的是工作流管理系统对其请求的响应，即工作流执行服务，并希望工作流执行服务是透明的。这要求工作流管理系统屏蔽底层应用系统的异构和分布特性，给用户提供一个单一的、透明的、透明的工作流执行服务的界面。另外，由于模块化设计以及可扩展性的需要，工作流管理系统本身也由若干模块构成，它们以一定的方式协作完成工作流的执行，利用分布式计算的工作流执行服务能够满足这些要求。

一个工作流管理系统应该提供的基本的工作流执行服务如表 1 所示：

表 1 用户请求及对应的工作流执行服务

请求	工作流执行服务
start	工作流实例的创建服务
与具体业务过程步骤有关的命令，如 edit、sign ...	工作流实例的执行服务
finish、delete、pause、resume、stop	工作流实例的管理服务
inquire	工作流执行状态查询服务

3 实现工作流执行服务的分布式计算方案

工作流执行服务的请求一般是用户通过工作流管理系统提供的用户界面（例如工作流客户端或 Web 浏览器）发出的。当用户发出一个工作流执行服务请求后，工作流执行服务必须完成以下任务：

- 根据工作流模型规定的逻辑步骤执行工作流；
- 确定所有有权限执行此步骤的参与者（人或软件 Agent），并检查是否满足同步和协调规则；
- 若满足，则执行具体的服务内容；若此步骤涉及到应用程序的调用，则工作流执行服务调用相应的应用；
- 根据此步骤的执行刷新相关用户的工作列表，并更新相应的数据库；

为了提高工作流执行服务的效率，通常采用多个工作流引擎并行协作的方式完成工作流执行服务。工作流引擎的协作基于客户/服务器模型，每个工作流引擎由完成特定功能的一组服务器和客户机组成。这样，客户机与服务器之间，工作流管理系统与其所集成的各种异构的、分布的应用之间的通信方式，成为影响工作流执行服务的效率、为用户以及工作流引擎之间提供透明服务的关键因素，为此，本文提出如图 2 所示的工作流执行服务的实现方案：

图 2 所示的工作流执行服务方案为 3 层结构，每层之间相互独立，下层通过接口为上层提供服务，其实现对于上层来说是透明的。最底层为操作系统，提供 LAN、WAN、Internet 等网络通信以及进程、线程等基本服务；中间层为分布式通信服务，可以用远程过程调用（RPC）或网络编程接口（WAPI）等方法实现，也可以由通信中间件实现，如 CORBA、J2EE 及消息队列技术等实现；最上层为工作流引擎及应用服务层，此层通过接口为用户

提供 workflow 执行服务。

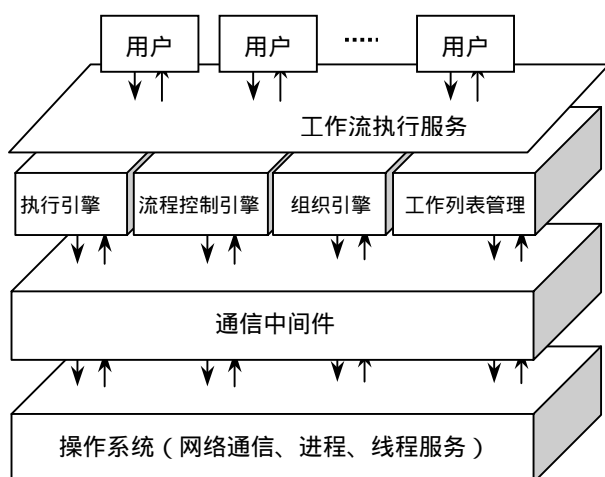


图 2 工作流执行服务实现方案

4 通信中间件

使用远程过程调用机制 (RPC) 或使用网络应用程序编程接口 (API) 提供分布式通信服务时, 开发人员除了解决业务问题外, 还必须处理基础设施的问题, 降低了开发业务问题解决方案这一问题的重要性。例如: RPC 本质上是同步的, 使用 RPC 的应用程序通常要理解网络传输的一些概念, 并负责定位满足请求的服务器; 虽然网络 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。

通信中间件是把应用程序和通信核心的细节分离的中间件。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。

5 工作流引擎的协作过程设计

工作流管理系统的用户通过工作流管理系统提供的用户界面 (例如工作流客户端或 Web 浏览器) 发出工作流执行服务申请, 然后由多个工作流引擎并行协作的方式完成工作流执行服务。并行是通过将工作流执行服务分解为若干子任务, 由特定的工作流引擎完成特定的子任务而实现的。

首先, 需要有一个工作流引擎受理用户从工作流客户端发出的申请, 以初始化相应的操作, 此工作流引擎名为**流程控制引擎**, 提供的接口为 *request()*, 调用参数可为工作流模型名、工作流实例名、操作名等, 例如:

```
request(loan, loan_to_companyA, start)
```

操作初始化后, 流程控制引擎需要协调所有有权限申请此服务的用户, 并检查此申请是否满足同步规则等, 此任务由名为**组织引擎**的工作流引擎完成, 组织引擎提供的接口为 *synchronize()*, 由流程控制引擎调用; 若返回结果为不满足, 则流程控制引擎需要更新申请者的工作列表, 并通知其申请被拒绝, 此任务由**工作列表管理引擎**实现, 其提供的操作为 *delete()*, 供流程控制引擎调用; 若返回结果为满足, 则具体的执行任务由名为**执行引擎**的工作流引擎来完成, 其提供的操作为 *execute()*, 由流程控制引擎调用。执行引擎在执行具体的服务时, 可能要调用相关的工作流应用。

执行引擎执行完毕具体的服务后, 需要将流程逻辑有关执行结果返回给流程控制引擎, 为此, 流程控制引擎提供一接口 *inform()*, 供执行引擎调用; 流程控制引擎为执行引擎完成此调用后, 需要确定下一过程步骤的执行者, 此任务也由组织引擎来完成, 提供的调用接口为 *navigate()*。流程控制引擎得到返回结果后, 将相关的工作流实例及可进行的操作添加到相应用户的工作列表中, 此任务由**工作列表管理引擎**完成, 提供的接口为 *submit()*。

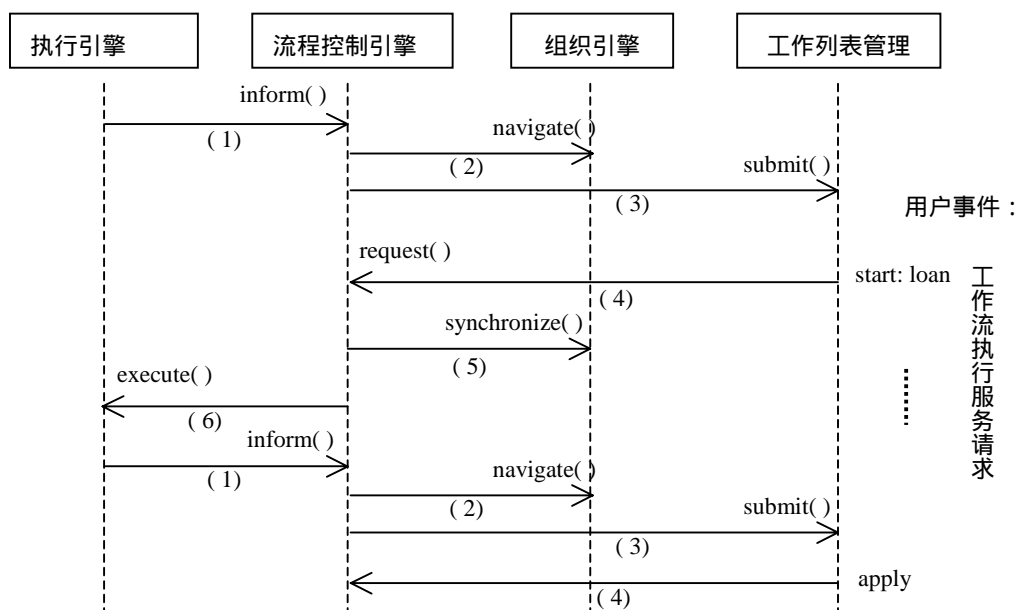


图3 workflow 执行服务的处理过程

工作流管理系统首先将每一个用户能够启动的所有工作流提交给此用户,并在其工作列表中列出所有等待其处理的工作流实例。以图1所示的贷款工作流为例,此工作流被提交到工作流管理系统的所有客户终端(包括远程登录的终端),以供需要贷款的用户启动一个贷款工作流,此过程涉及到的各工作流引擎之间请求/响应进程如图3中的(1)、(2)、(3)所示。

流程控制引擎接受用户发出的工作流执行服务请求后,引起的各工作流引擎之间请求/响应进程如图3中的(4)、(5)、(6)所示。每个工作流执行服务请求的处理过程都是4个工作流引擎重复此6个步骤的过程。

6 结论

工作流管理技术是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,这是通过在执行工作流时调用各种应用系统来实现的。因此,工作流管理系统需要与各种分布的、异构的应用通信或与其集成,为用户提供透明的工作流执行服务。本文提出一种具有3层层次结构的工作流执行服务实

现方案,其中的分布式通信服务层由通信中间件实现,使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。本文设计了4个工作流引擎并行协作完成工作流执行服务的接口及调用过程。

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.