

基于 UML 的 workflow 管理系统模型的研究

Research of an UML - based WfMS

时培芳 张永胜 (山东师范大学信息管理学院 济南 250014)

摘要:本文首先简要介绍了工作流和统一建模语言 UML,对工作流管理系统的模型进行了分析和研究,并在此基础上给出了基于 UML 的工作流管理系统的模型。

关键词:工作流 workflow 管理系统 UML

1 概述

工作流的概念起源于 20 世纪 70 年代的办公自动化和作业批处理领域,按照 WfMC (Workflow Management Coalition, 国际工作流管理联盟) 的定义,工作流就是部分或整个经营过程在计算机支持下的全自动或半自动化。根据一系列过程规则,文档、信息或任务能够在不同的执行者之间进行传递与执行。它的目的在于将工作分解成定义良好的任务、角色,按照一定的规则和过程来执行这些任务,并对它们进行监控,达到提高办事效率、降低生产成本、提高企业生产经营管理水平和企业竞争力。

UML (Unified Modeling Language) 统一建模语言是由世界著名的面向对象技术专家 Grady Booch, Jim Rumbaugh 和 Ivar Jacobson 联合开发的,它是在已有的三大面向对象方法学的基础上构建的,并吸取了其它面向对象方法和近 30 年软件工程的成果。1997 年

UML 被 OMG (Object Management Group) 批准作为面向对象建模语言的标准。它是一种通用的可视化建模语言,用于对软件进行描述、可视化处理、构造和建立软件系统制品的文档。UML 提供了 9 种图来描述建模过程的各个方面,这 9 种图是类图、对象图、包图、顺序图、协作图、状态图、活动图、构件图和配置图。

2 工作流管理系统参考模型

工作流管理系统的概念起源于 20 世纪 80 年代初,工作流管理联盟对工作流管理系统的定义为:工作流管理系统是一个软件系统,它负责工作流的定义和管理,并按照在计算机中预先定义好的工作流逻辑推进过程实例的执行。不同的工作流管理系统具有不同的应用范围和实施方式。1994 年 11 月,工作流管理联盟 (WfMC) 发布了工作流管理系统参考模型,该模型定义了一个基本的工作流管理系统所需要的过程定义工具、工作流执行服务、其他的工作流运行服务、客户应用程序、被调用应用程序和管理及监控工具六个基本功能构件,并且制定了功能构件之间的接口标准。

工作流管理系统参考模型如图 1 所示。

下面对模型的几个重要部件做进一步的分析与研究。

(1) 过程定义工具。过程 (process): 指一个业务流程,该流程可能包括计算机实现的部分和手工实现的部分。通过过程定义工具,用户对实际业务过程进行分析、建模,并生成业务过程可被计算机处理的形式化描述。过程定义工具

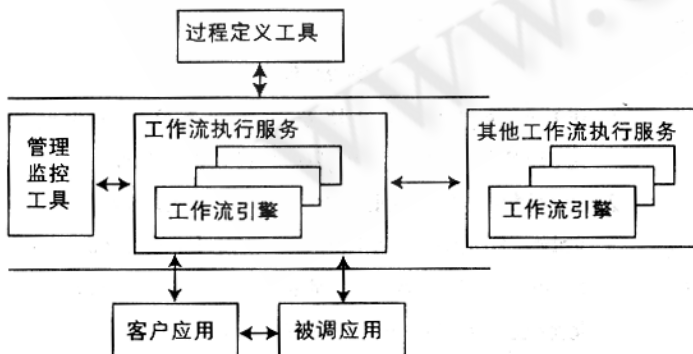


图 1 工作流管理系统模型

产生过程定义数据,供 workflow 执行服务实例化 workflow 时使用。过程定义的方法是 workflow 管理系统的一项非常重要的内容。一个好的定义方法不但应该有能力解释所有系统涉及的业务流程,而且应该易于理解与使用。

(2) workflow 执行服务。workflow 执行服务的主要功能是:通过解释过程定义数据,完成 workflow 实例的创建,控制实例流程的执行;生成有关的工作项通知用户进行处理;调用相关程序进行 workflow 数据的处理。workflow 执行服务通常由一个或多个 workflow 引擎支撑,多引擎的处理情况是每个引擎处理过程中的若干个活动或一个活动由多个引擎协调处理。在网络环境中,这些 workflow 引擎可能集中在同一个物理位置上工作,形成集中式的系统,或者分布在不同的物理位置上,形成非集中式的系统。此外,不同的 workflow 引擎还必须具备相互通信的能力,这些引擎可能属于一个非集中式的 workflow 系统,也可能属于两个或多个 workflow 系统。

(3) workflow 客户应用。workflow 客户应用的主要功能包括两方面,第一,接受来自于 workflow 执行服务的工作项列表,把列表中的每一个工作项传送到相应的用户界面,提醒用户进行处理;第二,向参与者提供完成业务工作的手段,可以通过内嵌的客户程序,也可以通过外部的应用程序。

(4) workflow 被调应用。workflow 被调应用是完成实际业务工作的具体应用程序,它本身并不属于 workflow 管理系统,但 workflow 管理系统必须研究如何与这些应用程序进行通讯,其中包括应用程序的调用,数据的传递等。

(5) 管理和监视工具。管理和监视工具负责监控 workflow 的执行,察看和改变 workflow 的活动状态。

workflow 参考模型还定义了 workflow 应用程序结构之间的接口,这些接口允许程序在不同层次上的交互操作。所有的应用程序都包含了相互作用的应用部件,不同的产品其部件的作用能力不同,为了完成 workflow 产品之间的交互操作,对其部件接口和数据交换格式进行标准化就非常必要。根据接口,可以构建不同的交互操作情况,识别市场上不同层次产品的性能。

3 基于 UML 的 workflow 管理系统的分析

利用 UML 提供的各种图形元素,包括活动图,可以方便直观地描述用户需求、系统的静态特性和动态行为,实现系统的模块化和模型的重用。利用 UML 来描述 workflow 管理系统有两个好处,第一,UML 是软件界公认的符号标准;第二,UML 也可以用在不需要实现细节的一般场合。基于 UML 的 workflow 管理系统的模型构建主要从构建以下几个 UML 图进行:

3.1 用例图

用例通过系统与一个或多个活动者之间的一系列消息描述了与活动者的交互,活动者包括人员、其它的计算机系统和进程。用例图从外部用户的角度捕获系统的行为,它将系统的行为划分为对活动者有意义的事务,这些功能片就是用例。用例图对活动者,所感知的系统功能进行建模,目的是列举活动者和用例,显示活动者在每个用例中的参与情况。

图 2 是 workflow 执行子系统的用例图。活动者包括 WfClient(workflow 客户端)、Monitor(workflow 监控端)、DefinitionDB(workflow 定义数据库)、EnactmentDB(workflow 运行数据库)、OrganizationDB(组织机构数据库)、ApplicationDB(应用程序数据库)、WorkItemDB(工作项数据库)、ConfigFile(workflow 系统配置文件)。这里,WfClient 作为接收用户交互的界面部分,依照固定的规则,将用户的请求送给 workflow 执行子系统进行处理。Monitor 作为接收系统管理员交互的界面部分,将系统管理员对系统作出的调整,发送给 workflow 执行子系统进行处理。其余的 DefinitionDB 等活动者,负责将 workflow 执行子系统每一步的操作与状态记录到数据库中,以永久保存。用例包括 ResourceLocate(资源定位)、EngineContainer(引擎容器)、ProcessDefLoad(定义装载)、ProcessMonitor(过程监控)、Util(公用程序)。其中,EngineContainer 通过 ResourceLocate 定位所有系统所用到的资源,表 EngineContainer 用例使用 ResourceLocate 用例,用带有箭头的实线表示。EngineContainer 不直接与用户交互,活动者对 workflow 的参与都是通过 ProcessMonitor 这个 workflow 执行子系统的入口来进行的。EngineContainer 通过 ProcessDefLoad 将现有的 workflow 定义装入,这样才能运行该 workflow,EngineContainer 用例与 ResourceLocate 用例之间是使用关系。

3.2 顺序图

顺序图表示随时间安排的一系列消息。每个分类角色显示为一条生命线,代表整个交互期间的角色。消息则显示为生命线之间的箭头。顺序图可以表达场景,即一项事务的特定历史。

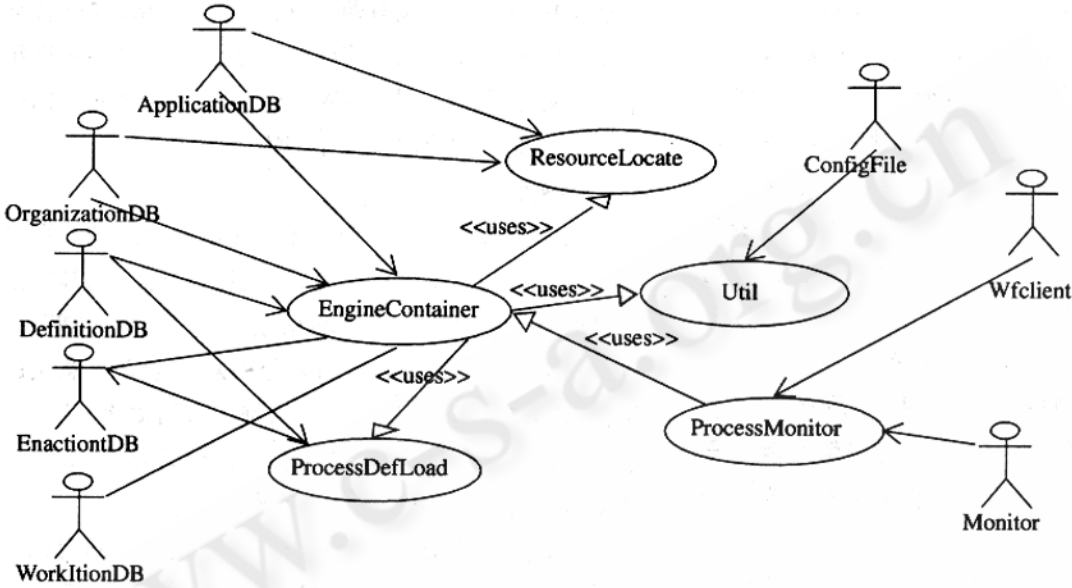


图 2 workflow 执行子系统用例图

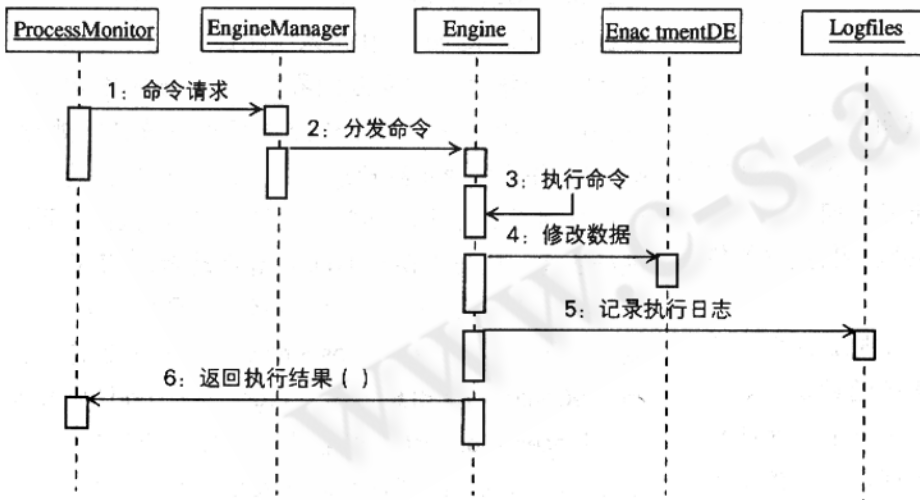


图 3 处理请求顺序图

顺序图以二维图表来显示交互。纵向是时间轴,时间自上而下。横向显示了代表协作中单个对象的分类角色。每个对象用方框表示,对象的名字在方框内部,并在名字的下方加下划线。每个分类角色表现为

垂直列-生命线。在角色存在的时间内,生命线显示为虚线;在角色的过程激活时间内,生命线显示为双线。消息显示为从一个角色生命线出发至另一个角色生命线的箭头,箭头用从上而下来的时间顺序来安排。顺序图的一个用途是显示用例的行为序列。当行为被

实现时,每个顺序图中的消息同对象的操作或状态机中迁移上的事件触发相一致。

图 3 表示处理请求用例的顺序图。图中五个方框分别表示五个对象: ProcessMonitor、EngineManager、Engine、EnactiontDB、Logfiles。这个用例是由 ProcessMonitor 接收用户操作,再将这些操作转换成固定的请求,发送给引擎执行而产生的。

当 ProcessMonitor 接收到用户在界面上所作的操作后,将这些操作转换为固定的命令请求,发送给 EngineManager。EngineManager 再根据接收到命令的类别,将命令分发给不同的 Engine。Engine 则具体执行相应的命令。Engine 执行完命令后,通知 EnactiontDB 修改相应的数据。接下来,Engine 再通知 Logfiles 将所作的操作记录下来,以供以后查询。最后,Engine 直接将结果返回给 ProcessMonitor,由 ProcessMonitor 将结果包装,显示

给用户。

3.3 协作图

协作图对交互中存在意义的对象和链建模。分类角色描述了对象,关联角色描述了协作中的链。协作

图通过图形的几何排布显示交互中的角色。消息显示为附属在连接分类角色的关系直线上的箭头。消息的顺序由消息描述前的顺序号来表示。协作图的一个用途是表现操作的实现。协作显示了操作的参数和局部变量,以及更永久性的关联。当行为被实现时,消息的顺序与程序的嵌套调用结构和信号传递一致。

图 4 是对应于处理请求用例的协作图。这个用例是由 ProcessMonitor 接收用户操作,再将这些操作转换成固定的请求,发送给引擎执行而产生的。这个协作图表现了处理请求用例所涉及的五个相关对象之间相互协作的关系。

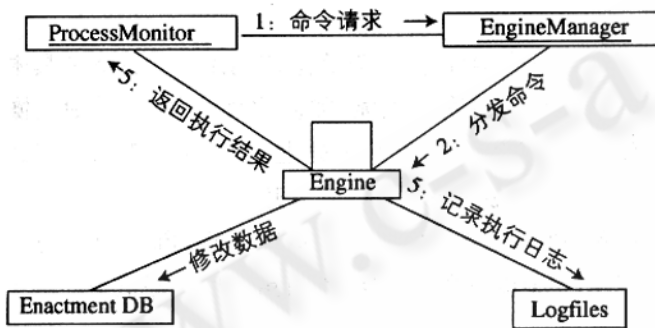


图 4 处理请求协作图

3.4 活动图

活动图描述了顺序和并发活动分组,是用于显示执行某个计算过程中的运算活动的状态机的一种特殊形式,活动图包含活动状态。活动状态代表了运算执行的状态:流程的步骤或操作的执行。活动状态表现了过程中语句的执行或工作流中活动的运行。与一般等待状态等待事件不同,活动状态等待的是运算的结束。当活动结束时,执行处理到图中的下一个活动。前一个活动结束时,活动图中的结束迁移被激发。活动状态通常没有外部事件的迁移,但它们可以由外围状态的事件而被取消。

3.5 状态图

状态指就某个特定类而言,对于发生的事件具有相同性质响应的一系列对象值。换言之,同一状态的所有对象以相同的方式响应某个事件,即对于给定的所有对象在接收到同一事件时执行相同的动作。而不同状态的对象可能对相同事件具有不同的响应,执行

不同的动作。

状态图包含由事件连接的状态。每个状态对对象生命期中的一段时间建模,该时间内对象满足一定的条件。当事件发生时,它可能导致迁移的激发,使对象改变至新状态。当迁移激发时,附属于迁移的动作可能被执行。在状态图中,状态用带圆角的长方形表示,初始状态用实心填充的圆表示,结束状态用实心填充的圆外套一个圆圈表示。

5 结束

workflow 管理系统的研究与发展对推动商业过程的自动化具有重要的意义,它对未来商业信息系统的集成和办公自动化系统的开发具有重要的指导作用,因此, workflow 系统的产品正越来越受到信息技术界及产业界的重视。但是, workflow 管理系统的参考模型并不是实际的应用模型,在实际应用当中,应该根据具体的应用环境做出灵活改变,而对同一个实际的 workflow 管理系统,用户和开发者往往采用不同的语言,导致了开发过程具有一定的难度。UML 作为一种通用的可视化建模语言,它为描述面向对象系统定义了一系列的标准符号,用 UML 来描述 workflow 管理系统,可以增强领域专家、 workflow 专家、软件设计者、用户以及其他不同背景专家之间的联系和交流,使 workflow 管理系统的开发能够顺利进行,并能更好地满足人们的需求。

参考文献

- 1 范玉顺, workflow 管理技术基础,北京 清华大学出版社,2001。
- 2 Craig Larman. UML 和模式应用,北京 机械工业出版社,2002。
- 3 孙惠民. UML 设计制作宝典,北京,中国铁道出版社,2003。
- 4 谢杰华、陈震、罗源,基于 Web 的 workflow 管理系统,计算机应用研究,2002(6):141-143。
- 5 罗海滨、范玉顺、吴澄, workflow 技术综述,软件学报,2001,11(7):899-907。
- 6 丁柯、金蓓弘、冯玉琳,事务 workflow 的建模和分析,计算机学报,2003.26(10)。