文章编号:1001 - 9081(2005)11 - 2692 - 03

基于关系结构的流程控制模型

宋 志[1],刘 剑[2],张 艳[3]
(1. 中国科学院 研究生院,北京 100039;2. 中国电子设备系统工程公司,北京 100089;
3. 北京市泰克公路科学技术研究所,北京 100083)
(hebzy255@sohu.ocm)

摘 要:深入讨论了工作流流程的起点模型、流程激活模型、流程运转模型以及终点模型。在传统的关系数据库的基础上,提出了一个适用于关键业务开发的基于关系结构的流程控制模型,说明了该模型目前的应用情况。

关键词:关系;工作流引擎;任务;流程控制模型
中图分类号: TP311.52 **文献标识码**:A

Relation-based flow control model

SONG Zhi[1], LIU Jian[2], ZHANG Yan[3]
(1. *Graduate School, Chinese Academy of Sciences, Beijing* 100039, *China*;
2. *Electronic Equipment and Systems Engineering Corporation of China, Beijing* 100089, *China*;
3. *Tech Highway Science & Technology Research Institute, Beijing* 100083, *China*)

Abstract: The model of start node, the model of activation, the model of flow's operation and the model of stop node in workflow technology were thoroughly discussed. Based on traditional relationship database system, a relationship-based flow control model was proposed. The application of this relation-based flow control model was explained.

Key words: relationship; workflow engine; task; flow control model

0 引言

目前,针对部门或企业的计算机应用愈加复杂,已经不仅仅停留在文档处理、信息发布、电子白板等简单的业务层面上,而是要求将信息技术扩展到关键业务之中。关键业务是部门或企业的业务核心内容,其业务过程往往由许多业务活动组成,业务逻辑以及业务规则较为复杂,业务执行依赖于众多业务活动之间的交互和众多业务人员的协作,而且因为关键业务涉及部门日常必须完成的核心业务工作,往往存在着海量数据。因此关键业务的信息化是目前针对部门或企业的信息系统开发的核心环节之一。

工作流是多个参与者之间按照某种预定义的规则传递文档、信息或任务的过程自动进行,从而实现某个预期的业务目标[1]。工作流技术本身所具有的协调本质决定了其在关键业务信息化过程中所扮演的重要角色。工作流管理联盟(WFMC)给出了工作流系统的通用框架——工作流参考模型[2]。在工作流参考模型中,工作流引擎为工作流管理系统在定义时提供支持在运行时提供解释和服务,是工作流管理系统的核心。工作流引擎主要分为机构模型、信息模型和控制模型[3]。

关键业务的执行是若干业务活动的集合,这些业务活动按照一定的规则前后连接在一起,相互协作,最终实现关键业务的推进。业务推进的关键点在于业务过程的活动和路径转移。也就是说,与关键业务推进密切相关的关键技术是工作流引擎的信息模型(本文称之为流程控制模型)设计及实现。本文针对关键业务信息化的需求,以某电子政务项目为背景,详细讨论了工作流引擎的流程控制模型,最后基于关系数据结构的形式设计了一个轻量级工作流引擎的流程控制模型。

1 流程

流程控制模型定义工作流引擎所用到的各种控制数据。通过流程控制模型可以方便地描述关键业务的业务规则和活动依赖关系。关键业务的推进是由任务或者活动构成的。在描述业务中,一个任务表示的是流程所需要完成的某一项工作,这项工作可能是一次操作即可完成,也可以是几次操作的组合。活动是 WFMC 的标准模型元素,描述的是工作流的一个活动。在 XPDL 中,活动是描述流程运转的最小单元。流程定义通常用活动的集合表述,但活动混淆了“状态”和“动作之间的差异”,在流程中状态(或者说等待状态)代表了一种对外部参与者的依赖,在流程运行时这意味着流程引擎必须等待,直到外部参与者通知工作流管理系统指定的状态完成了。动作是在流程运行过程中,工作流系统为响应指定时间运行的一段逻辑程序。当流程运行过程中指定事件发生时,工作流系统启动并执行这些动作。因此为避免混淆概念,在本文对流程的分析描述中采用任务概念而不是 WFMC 提出的活动概念。

1.1 流程起点模型

起点是一种特殊的任务节点,在这个节点上可以执行一定的操作,也可以仅仅是一些数据状态的改变。起点的运行最终会导致一个流程实例的产生,即激活一个流程。

起点模型分为单起点、多起点两类,如图 1 所示。

单起点模型是最常用的工作流激活方式,起点激活工作

收稿日期:2005 - 05 - 25;**修订日期**:2005 - 08 - 16

作者简介:宋志(1977 -),男,重庆潼南人,助理工程师,硕士研究生,主要研究方向:计算机应用、信息安全; 刘剑(1978 -),男,吉林吉林人,硕士,主要研究方向:计算机信息安全; 张艳(1978 -),女,河北石家庄人,工程师,硕士,主要研究方向:光通信.

流后,流程沿任务 A→任务 B 方向顺序执行。

多起点模型主要有三种方式。多起点方式一:起点 A 和 B 都可以激活流程的运行,而且激活后流程都共同指向任务 A,对于任务 B 而言,不关心流程是如何激活的,只与从任务 A 传来的正确的流程数据相关。多起点方式二:起点 A 激活工作流后,流程沿任务 A→任务 B→任务 C 方向流转;如果从起点 B 激活工作流,任务 A 则被跳过。这种起点方式可以看成两个子流程的选择性汇总。多起点方式三:虽然存在多个起点,但是基本按照一个统一的流程方向运行。这种多起点方式也可以理解为多个流程之间的信息交互,流程 A 发送消息数据,激活流程 B 的运行。但未必一定从流程 B 的默认起点激活,可以从流程 B 的中途某个任务激活。

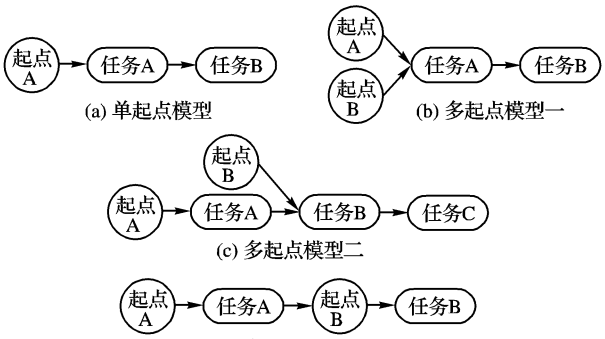

图 1　起点模型

1.2　流程激活模型

实例化工作流,需要有一系列条件来激活,或者说需要一系列条件来激活流程起点。一个流程被激活后,会从起点开始沿着预定流程路线,有序或无序地往下进行。流程的激活方式主要有以下几种。

1.2.1　人工激活

绝大多数流程的激活,都是以人为的信息数据输入实现的。比如在公文流转过程中,工作人员呈批件的填写,激活了呈批件处理流程的开始。

1.2.2　按时间激活

在特定的时间,由于特定的情况,符合特定的条件,激活某个特定的流程(或任务)。这种激活方式在现实生活中很少单独出现。多数情况下,因为在某一个流程中,限定时间内,因某项任务未达到预期状态,而激活另外的任务或处理流程。这种激活方式是受外来因素影响的,而大多与一些流程任务一起出现。仍以呈批件为例,当呈批件流程运转到某个节点若干天以后仍然没有下一步动作执行,系统自动发送催办通知,这就是一种按时激活工作流的方式。

1.2.3　外部消息激活

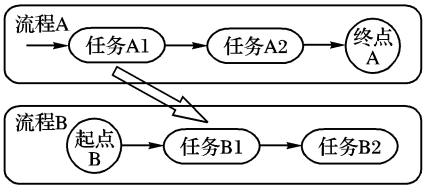

图 2　外部消息激活工作流模型

多流程信息交互及流程嵌套实现时普遍采用外部消息激活的方式。如图 2 所示,来自流程 A 的消息可以是流程 A 中某个任务所发,也可以是流程结束时所发;流程 B 被激活点可以是起点 B,也可以是流程 B 的中间起点。

1.3　流程运转模型

在工作流中,一个流程是由一系列的任务按照某种预定方式组合起来的。任务不是最小的动作单元,任务由多个动作组成。基本的运转模型包括串行(Sequence)、自循环型(Self-Cycle)、并行(Parallel)、异或(XOR)、鉴别(Discriminator)、抄送、同步聚合(Synchronize)、简单聚合(Simple Merge)、多重聚合(Mutiple Merge)和鉴别聚合(Discriminator Merge)[4]。其中“抄送”是一个不规范的工作流运转模型,但是现实中应用比较广泛。如图 3 所示,存在主流程任务 A→任务 C,在一个任务 A 执行完毕后,继续执行主流程上下一个预定任务 C,但同时激活另一个任务 B(或者根本就是激活另外一个流程)的执行,但是任务 B 及其后续流程不对主流程运转造成影响。

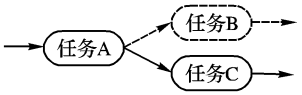

图 3　流程运转模型——抄送

另外,根据日常办公特点,还有如下几种比较特殊的流程运转模型:

退回　也可称为回退。例如:任务 A 到任务 B 为正常发送,但从任务 B 到任务 A 出现两种情况:1)正常发送;2)因为某些特殊原因被任务 B 退回,要求任务 A 重新办理。退回模型在电子政务中的公文流转或审批中应用比较广泛。

自由流　一个任务执行结束后,其后续运转不按照预定的顺序执行,而是人为地动态选择。流程的打乱会带来很多问题,数据完整性、流程可控性都会受到影响。但是在电子政务中,自由流的情况非常普遍,不能回避。

代办　流程中任务流转到员工 A,但员工 A 由于出差等原因无法正常办理,可以委托给员工 B 处理,以保证流程正常运转。

催办　执行完任务 A 到任务 B 的运转后,任务 A 设定一个催办日期(或者由执行任务 A 的角色人工催办),在催办日期到来的时候向任务 B 发送催办请求。

撤办　也可称为撤收。流程由任务 A 运转到任务 B,任务 B 虽然接受了任务 A 的请求或数据,但还没有确认执行的情况下,任务 A 有权撤办任务 B,重新执行。

1.4　流程终点模型

流程终点从节点数目和形态上划分可分为单结束点、多结束点和非标准结束点。在非标准结束点结束的流程,并不在预定义结束点结束,而是因流程中间的异常行为或人为原因强制终止。按结束点行为可分为正常终止、异常终止、激活新任务三类结束点。

2　基于关系结构的工作流引擎流程控制模型

基于关系的工作流引擎流程控制模型指的是工作流引擎中的流程控制模型(即信息模型)通过关系结构来表达;控制工作流引擎运作的各种程序逻辑(即控制模型)也是通过常规关系数据库管理系统中所提供的存储过程、包以及触发器等机制来实现;同时,事务的并发控制也通过数据库系统所提供的机制来实现。

从技术角度来说,使用关系结构来表达流程控制模型可以降低开发过程中的技术难度,减少工作量。与流程相关的各种业务活动的状态和数据可以存储在数据库系统中;与之相关的数据的完整性可以由数据库管理系统来维护;利用关

系结构可以方便地定义流程控制中的各种数据格式和数据结构;可以方便地利用数据库管理系统提供的各种 DML 语句来操纵工作流引擎所需的各种数据。从开发应用系统的角度来看,针对关键业务的应用系统通常会采用一个常规的关系数据库系统作为后台的支撑,因此,采用基于关系结构的工作流程控制很容易与应用的开发环境做到无缝的集成。

图 4 给出了基于关系结构的轻量级工作流引擎流程控制数据模型 ER 图(核心表结构)。核心是任务活动表 TASK_ACTIVITY,与之相关的表结构主要有任务过程 PROCESS、活动流转规则 ROUTING_RULE、上一步依赖规则 PRE_ACT_RULE、任务指派规则 ASSIGN_RULE、任务列表 NEXT_TASK_LIST 以及已完成的任务列表 DEAL_TASKS。

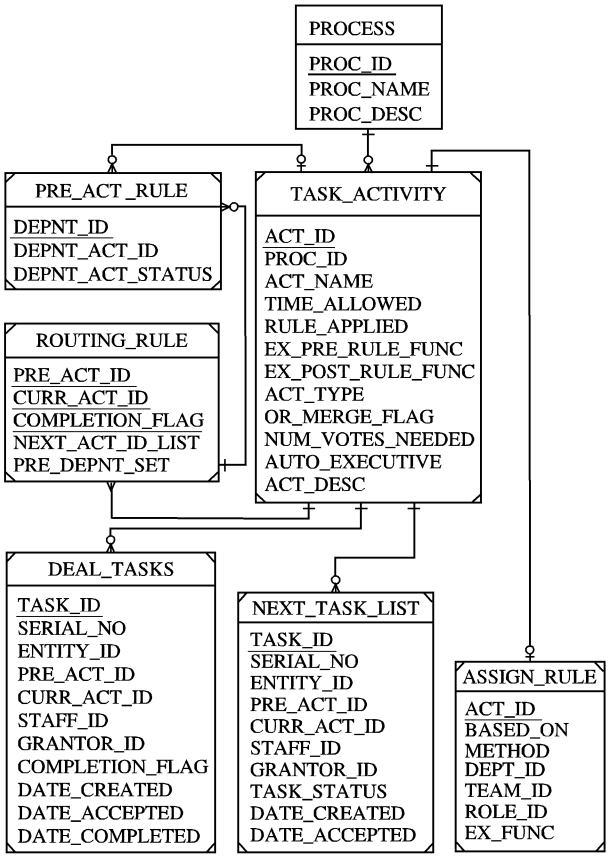

图 4 流程控制数据模型 ER 图

2.1 活动类型

每个任务过程由若干任务活动组成,任务活动通过 ACT_ID 来唯一标识,ACT_TYPE 指明相应活动的类型。同一个业务活动在工作流运行时可以有多个实例。属于同一任务过程的任务称为属于同一批次的任务。我们为流程控制部分设计的活动类型分为:INITIAL(初始化活动)、INTERACTION(常规交互活动)、AUTOMATION(常规自动活动)、AND_BRANCH(与分支活动)、AND_MERGE(与汇聚活动)、OR_MERGE(或汇聚活动)、VOTE_MERGE(投票汇聚活动)、DUMMY(哑活动)和 COMPLETION(终结活动)。

2.2 流程运转规则

在工作流引擎中,流程运转规则可以分解成活动的前依赖规则和活动的后转发规则[5,6]。前依赖规则包含顺序、与汇聚、或汇聚和投票汇聚四种规则。后转发规则包含顺序、或分支和与分支三种规则。图 4 中的 PRE_ACT_RULE 表、ROUTING_RULE 表以及 TASK_ACTIVITY 表中的 ACT_TYPE 和 RULE_APPLIED 等字段联合表示活动的前依赖规则和后转发规则。

可以用很简洁的关系结构来表达活动的前依赖和后转发规则。首先 TASK_ACTIVITY 表中的 RULE_APPLIED 字段指示相应活动应该采用何种规则判断准则,它可以有四种取值:DEFAULT,USER_DEFINED_PRE_RULE,USER_DEFINED_POST_ROUTING_RULE 和 USER_DEFINED_BOTH_RULE。DEFAULT 表示由工作流引擎自动根据 PRE_ACT_RULE 表和 ROUTING_RULE 表来进行规则检查。考虑到业务规则的多样性,本文提供了自定义方式来表达那些无法用缺省规则表示的特殊业务规则,TASK_ACTIVITY 表中的 EX_PRE_RULE_FUNC 和 EX_POST_RULE_FUNC 分别指定了前依赖和后转发规则的自定义调用接口。

活动的后转发规则主要通过表 ROUTING_RULE 表示,后转发规则可以用如下四元组来表达:

Post_Routing_Rule = (PRE_ACT_ID, CURR_ACT_ID, COMPLETION_FLAG, NEXT_ACT_ID_LIST)

其含义是:在当前活动的 ACT_ID 为 CURR_ACT_ID 的情况下,如果当前活动的前趋活动的 ACT_ID 为 PRE_ACT_ID 并且当前活动的结束标记为 COMPLETION_FLAG 的话,工作流将流向由 NEXT_ACT_ID_LIST 所指明的后继活动。

前依赖规则需联合 PRE_ACT_RULE,ROUTING_RULE 和 TASK_ACTIVITY 共同表示,前依赖规则可以用一个三元组来表达,即:

Pre_Dependency_Rule = (PRE_ACT_ID, CURR_ACT_ID, PRE_DEPNT_SET)

PRE_DEPNT_SET 为前依赖活动集,其中的每一个元素又可以用另外一个三元组来表示:

Element_Pre_Depnt_Set = (DEPNT_ID, DEPNT_ACT_ID, DEPNT_ACT_STATUS)

Pre_Dependency_Rule 的含义是:由前趋活动 PRE_ACT_ID 流转过来的当前活动 CURR_ACT_ID 能否启动取决于前依赖活动集 PRE_DEPNT_SET 中所包含的那些活动是否已经到达各自应该到达的结束状态 DEPNT_ACT_STATUS。可以看出,只有在前依赖活动集中出现的那些前趋活动才可以联合构成对当前活动的约束关系,如果某个前依赖规则三元组中的 PRE_DEPNT_SET 为空集,则表明由此前趋活动流到当前活动的流转过程跟其他前趋活动没有任何关系,与此相应的当前活动可以立即启动。

另外,一个活动可以同时具有多个实例,即任务,这些实例可以是属于同一批次的,也可能属于不同的批次,流水号 SERIAL_NO 用来标识任务所属的批次,所有属于同一批次的任务具有相同的流水号;不同的任务之间则通过唯一的 TASK_ID 进行标识。

3 结语

本文详细地分析了工作流起点模型、激活模型、运转模型以及终点模型。在此基础上设计实现了一个基于关系结构的工作流引擎流程控制模型。基于该流程控制模型的电子政务关键业务处理系统目前已基本开发完毕,实践证明该模型相对简单实用,效果较好。但系统通用性与原型设计尚有差距,有待进一步完善。

参考文献:

[1] Workflow Management Coalition. The workflow reference mode[EB/OL]. http://www.wfmc.org, 1994.

(下转第 2697 页)

他对象的结构。

同样,Resource Allocation 是一个关于资源分配的分析模式,它由要完成的工作(Job)、工作的类别(JobCategory)、功能或技能(Facility)以及资源(Resource)等对象构成。其中资源分配模式中用 OCL 来描述关于资源的一个不变式表达式来说明不能在同一时间区间内同时分配同一资源。通过对测量模式、资源分配模式的组合使用,就可以得到针对某个特定领域的设备管理的业务模式(Equipment Management)。这里以发电及供电行业为应用背景,对电力企业的设备(Equipment)来说,应用测量模式来描述设备的运行状况,同时用资源分配模式来描述设备与生产任务的关系。在业务模式中,还可以进一步加入领域相关的对象、关系及约束等建模元素,例如设备的安装地点(Location)等。对供电企业中的某些特殊设备,例如变压器(Transformer)应用设备管理业务模式就可以得到同时涵盖运行参数测量和生产任务安排等业务功能的模型。通过对模型中的模式进行展开处理,就得到关于变压器管理的完整业务类图。

在使用模式来进行业务建模时,是直接使用应用分析模式,还是应用业务模式,或者两者同时应用,建模者可以根据实际情况来灵活选择。

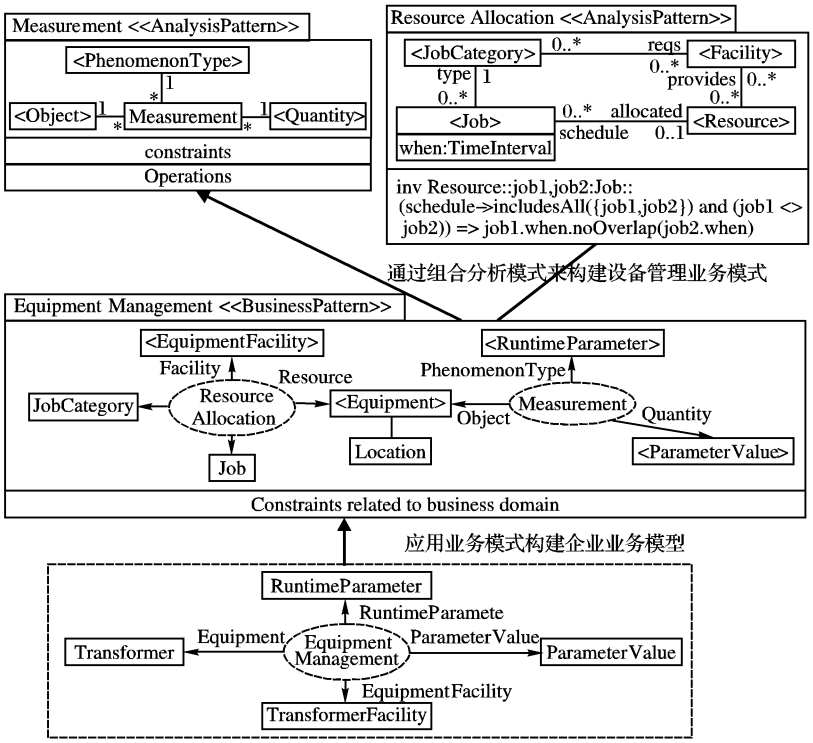

图 3　电力企业变压器设备管理业务模型的构建过程

4　结语

在业务建模过程引入模式的复用,并通过对各种模式的组合及改造,可以改善业务建模的效率和质量。需要注意的是,建模人员必须对模式所应用的环境有着充分的理解,避免模式的不当或过度使用。有时由于外部环境或系统本身的可靠性以及效率等条件的约束,针对相同的问题域可能需要在不同的业务模式中进行选择。对那些同时为不同领域的企业提供运行管理、电子商务等信息系统解决方案的软件开发商来说,在业务建模中引入不同层次的分析和业务模式不但可以在不同领域以及项目中共享以往的建模经验,同时可以简化建模的复杂程度,提高软件系统的质量和扩展性。同时,对那些应用信息系统的企业而言,由于同业务模型相比,模式表示的是更为一般的事实,业务模式或分析模式的应用将有助于企业在不同抽象程度上定义一致的专业术语集合以及业务对象之间标准的行为结构。这也是在不同信息系统之间实现基于语义的信息集成的必要前提。

参考文献:

[1] GAMMA E, HELM R, JOHNSON R, *et al*. Design Patterns: Elements of Reusable Object-Oriented Software[M]. Addison-Wesley, 1995.

[2] FOWLER M. Analysis Patterns[M]. Addison-Wesley, 1997.

[3] VERYARD R. Component - Based Business Background Material - Business Patterns[J]. CBDi Forum Journal, 2000,(6).

[4] ERIKSSON HE. PENKER M. Business Modeling with UML Business Patterns at Work[M]. John Wiley& Sons, 2000.

[5] DODANI M. Pattern Driven Solution Engineering[J]. Journal of Object Technology, 2003,2(2).

[6] Object Management Group. UML Profile for Patterns Specification [S], 2004.

(上接第 2694 页)

[2] van der AALST WMP. The application of Petri nets to workflow management [J]. Journal of Circuits, Systems, and Computers, 1998,8(1):21 - 66.

[3] 谢玉凤,杨光信,史美林. 基于条件化有向图的工作流过程优化[J]. 计算机学报,2001,24(7):729 - 735.

[4] Workflow Patterns[EB/OL]. http://is.tm.tue.nl/research/patterns/, 2005 - 04.

[5] LEUNG KRPH, CHONG JML. The Liaision Workflow Engine Architecture[A]. Proceedings of the 32nd Hawaii International Conference on System Sciences[C]. http://www.computer.org/proceedings/Hiccs2/, 1999.

[6] 张晓东,柴跃廷,任守榘. 基于业务规则的事件驱动建模方法[J]. 清华大学学报,1999,39(7):25 - 28.