

# OA 系统流程模型分析与元模型设计

查伟忠, 杨季文

(苏州大学计算机科学与技术学院, 苏州 215006)

**摘要:** 分析办公自动化(OA)系统中的流程模型, 参考工作流管理联盟的基于 XML 的过程定义语言, 提出 OA 系统的基本流程元模型设计, 其中包括 OA 系统的流程模型、元模型规范、处理规范等, 解决了 OA 系统中流程的标准化处理问题, 适用于建立具有行业范围的 OA 系统流程及处理规范, 有利于促进 OA 系统的标准化建设。

**关键词:** OA 系统; 元模型; 流程模型; 基于 XML 的过程定义语言

## Workflow-model Analysis and Meta-model Design of OA Application

ZHA Wei-zhong, YANG Ji-wen

(School of Computer Science and Engineering, Suzhou University, Suzhou 215006)

**【Abstract】** By analyzing the workflow model of OA application and referring the XML process definition language(XPDL) of workflow management coalition(WFMC), this paper works out a basic workflow meta-model of OA application, which is composed of the workflow model of OA application, the schema of meta-model and the schema of process management. It can resolve the key issue of OA application in standardizing the process management, and publishing the handicraft of workflow model used for trade area, which is good for improving the standardization of OA application.

**【Key words】** OA application; meta-model; workflow-model; XML process definition language(XPDL)

办公自动化(OA)系统是融信息处理、业务流程管理和知识管理于一体的应用系统, 其基础特征是具有优秀的流程处理和优化能力。通过信息的流程化管理能规范企业的管理模式, 能提高企业的管理效率; 通过优化信息的流程化处理, 企业能用现代化流程管理思想对目前的业务流程、管理模式进行重组, 进一步提高管理效率; 通过实施 OA 系统流程管理的模型化和标准化, 极大地提高了 OA 系统的可管理性和开放性。本文通过分析 OA 系统的流程特点, 依据工作流管理联盟(workflow management coalition, WFMC)定义的相关过程定义规范, 提出了 OA 系统的基本流程元模型设计和相应的处理机制。

### 1 WFMC 定义的过程管理及元模型

业务流程管理组织 (business process management initiative, BPMI)和 WFMC 于 2002 年定义了如图 1 所示的工作流过程定义规范(workflow process definition interface)。

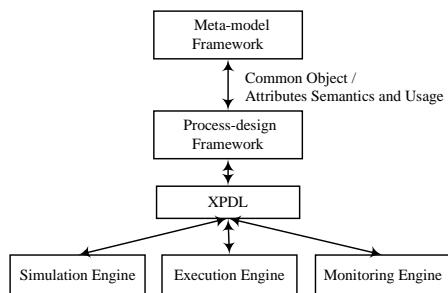


图 1 过程定义概念

在该规范中强调了过程建模和基本过程元模型定义的必要性, 并提供了一组基本的元模型和过程定义语义规范(XML schema for XML process definition language, XPDL<sup>[1-2]</sup>)。

WFMC同时给出了在过程定义中各元模型的关系及处理原则。

### 2 OA 系统流程模型分析

OA 系统的核心处理对象为“文档”, 具体实现时抽象为一个数据模型对象。一个“文档”对象具备了一个独立的生命周期(document object life cycle, DOLC)特征, OA 系统的所有功能都围绕一个“文档”的生命周期展开。一个完整的 OA 系统流程应覆盖一个“文档”的特定生命周期, 从激活(新建)、处理(流转)到终止(归档)。每个特定的生命周期阶段都应对应一个特定的状态特征, 用以描述阶段特征和定义流程的切换条件。以下通过一组定义来描述 OA 系统的 DOLC 及其基本模型。

**定义 1** 一个“文档”的生命周期模型可描述为:  $DOLC=(Starter, \{Processors\}, Packager)$ 。其中, Starter 为流程激活器, 可以是个人操作, 也可以是一个应用程序或其他流程的一个流程处理器(Processor);  $\{Processors\}$  是 Processor 集合, 每个处理器对应于“文档”的一个特定状态下的一个或一组活动(Activity); Packager 为一个结束文档生命周期的封装器, 一般对应为归档操作或者是作为另一流程的激活器。

**定义 2** 一个激活器可描述为:  $Starter=(Actor, \{Data\})$ 。其中, Actor 为参与者对象;  $\{Data\}$  为初始化数据模型, 可以是 Application Data(如文档数据模型对象)、Environmental Data(如 Application Data 的存储信息)、Workflow Relevant Data(如预定义的 Processors 数据, 即流程数据)。

**作者简介:** 查伟忠(1971 -), 男, 硕士, 主研方向: Web应用; 杨季文, 教授

**收稿日期:** 2006-12-16 E-mail: zhawz@suda.edu.cn

**定义 3** 一个处理器可描述为： $Processor=(\{Activities\}, \{Rules\}, \{Transitions\})$ 。其中， $\{Activities\}$ 可以是一个或一组相关的活动，活动可以是一个具体功能的实现，也可以是一个子流程； $\{Rules\}$ 是与 $\{Activities\}$ 对应的规则，定义了执行活动的规则和各活动之间的联系及切换条件； $\{Transitions\}$ 是一组输入输出转换逻辑，输入逻辑对应为与前一个 Processor 的关系，输出逻辑对应为如何触发下一个 Processor，即定义下一个 Processor 的输入逻辑。

**定义 4** 一个活动可描述为： $Activity=(\{Actions\}, Role, State)$ 。其中， $\{Actions\}$ 为一个或一组相关的动作，即完成该活动所需的功能集合；Role 为执行该 Activity 所需的角色信息，通过定义角色保证了执行活动的合法性；State 定义了活动的状态。

**定义 5** 一个规则可描述为： $Rule=(Condition, Selector, \{Methods\})$ 。其中，Condition 定义了活动的切换条件；Selector 为条件选择器； $\{Methods\}$ 为活动切换方法，即条件选择器根据运行时信息和定义的切换条件，选择适合的活动切换方法完成活动关系的描述。

**定义 6** 输入转换逻辑可描述为： $Transitions=(\{Inputs\}, Method, Data)$ 。其中， $\{Inputs\}$ 为前一个或多个处理器的输出信息，主要为流程数据；Method 为处理 Inputs 的逻辑和方法；Data 为 Method 方法的处理结果数据模型。在 OA 系统中可以用“串行输入”和“并行输入”来描述输入条件。对应于“串行输入”，Method 主要完成对输入信息的整形(Shape)过程，而对应“并行输入”，Method 的处理逻辑比较复杂，一般可以抽象为以下几类：

(1)  $M=A \wedge B$ ，其中，M 为方法；A 和 B 为前导处理器。这种方式在 OA 中比较常见，主要用于多部门协同工作，即对多个前导处理器的输出进行同步聚合，A 和 B 必须处于结束状态(State="Completed")，该方法才执行，否则当前活动处于等待状态(State="Waiting")。

(2)  $M=A \vee B$ ，这种方式在 OA 系统中也比较常见，主要用于“窗口式”服务，一般采用“先进先出”的堆栈规则，即多个前导处理器中只要有一个结束，当前处理器即处于工作状态(State="Ready")，该方法即运行，进而改变处理器状态为运行(State="Running")，而且发出“Aborted”信息，使其他已在工作或未工作的处理器处于终止状态(State="Aborted")。

(3)  $M=A \vee B \oplus \overline{A \vee B}$ ，这种方式为方式(2)的变体，区别在于当前处理器工作时并不发出终止信息，其他前导处理器继续运行，其处理结果也不被丢弃，而是作为“次要”数据被保存。

(4)  $M=R(A, B)$ ，其中，R 为方法规则。这种方式主要用于自动机处理，即当前处理器为一个机器程序(如一个 Agent 程序)。在处理该类方法时，一般引入规则引擎和规则库，以使系统更具灵活性和可管理性。

**定义 7** 输出转换逻辑可描述为： $Transitions=(Trigger, Data)$ 。其中，Trigger 为触发器，决定了流程的下一个阶段所需的处理器及相互之间的关系；Data 为当前处理器存储的数据模型，作为下一个处理器的输入信息，其中包括更新的 Application Data、Workflow Relevant Data 等必需的数据模型。在 OA 系统中，输出转换逻辑可以描述为“串行输出”和“并行输出”，对于“串行输出”，触发器往往由人工操作执行，

或依据预定义的串行路径信息由程序自动执行。“并行输出”一般可以抽象为以下 4 类：

(1)  $T \Rightarrow A \wedge B$ ，即一个处理器的结束导致了并发的多个处理器的产生，这种方式比较常见，一般会同时定义一个会聚点(一个处理并行输入的处理器)，但多个并发的处理器并不一定都需要聚合，主要决定于会聚点的性质(见输入转换逻辑 Method 类型描述)。

(2)  $T \Rightarrow A \vee B$ ，即一个处理器的结束导致了可能的多个处理器的产生，但只能选择一个处理器最终执行，在这种方式下，选择的动作一般是人为的选择。可以对应为输入转换逻辑的“串行输入”。

(3)  $T \Rightarrow S(A, B)$ ，其中，S 为一个由规则驱动的选择器，一般设计为一个自动机(如一个 Agent 程序)，即由自动机动态获取运行时信息，根据预定义的规则触发下一个处理器。在 OA 系统中，常用这类模型处理任务的固定分解、分配及固定条件路径的选择、循环等事务。

(4) 抄送模型，即完成主路径的同时，生成一条辅助路径。这种模型在 OA 系统中也经常出现，一般处理原则为只记录数据而不处理结果。

**定义 8** 一个动作可描述为： $Action=(Actor, Data)$ ，其中，Actor 为执行者，一般为一个界面操作，也可以是一个 Agent 程序；Data 为动作的结果数据模型。

**定义 9** 一个角色可描述为： $Role=(Name, Capabilities, Authorizations)$ ，根据具体的应用环境，特别是身份认证机制，角色的定义内容项并不非常严格，这里仅给出一个参考模型，其中，Capabilities 定义了该角色的能力集合；Authorizations 定义了该角色访问相关资源的权限。

**定义 10** 活动的状态是一个标志信息： $State=\{\text{"Waiting"}, \text{"Ready"}, \text{"Running"}, \text{"Sleeping"}, \text{"Canceled"}, \text{"Aborted"}, \text{"Completed"}\}$ 。状态之间的转换关系见图 2。

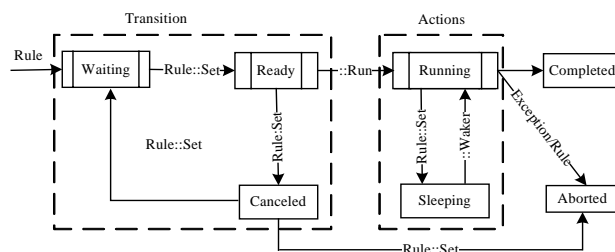


图 2 活动状态转换逻辑图

**定义 11** 一个封装器可描述为： $Packager=(Actor, \{Data\})$ ，其中，Actor 为参与者对象； $\{Data\}$ 为结果数据模型，如完整的文档数据、流程数据、分析报表等。

在 OA 系统中还有一些特殊的动作，如“回退”、“催办”、“收回”、“委托代办”等，这些动作不适合作为单独的处理单元或模型来设计，而应作为一个处理器中的一个特殊活动加以描述，通过限定该处理器的规则和输入输出逻辑来配置和执行这些特殊动作。

### 3 OA 系统流程元模型描述

WFMC 在 XPDL 中标准化了工作流过程(Process)的定义模型<sup>[3]</sup>，归纳了工作流过程的元模型及封装方法，强调了充分利用建模工具定义工作流管理系统中的基本流程模型，以达到复用流程模型的目的。其定义规则非常适合于流程的自动化执行。在 OA 系统中采用标准化的元模型来建模流程模

型,可充分提高OA系统的可管理性、集成性、开放性及灵活性,对促进OA标准化建设具有非常重要的意义。图3是参考了XPDL建立的OA系统流程元模型及其关系。

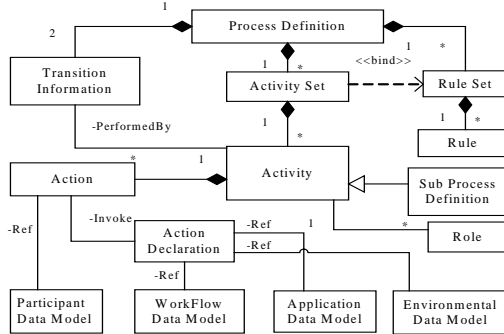


图3 OA系统流程元模型

**规则1** 一个处理器对应为一个过程定义,过程定义由 workflow引擎负责解释、执行,处理器的执行情况由 workflow引擎进行跟踪。

**规则2** 过程定义可以预先建模,也可以是自由流。自由流由用户操作进行定义,一般通过设计视图控制器来实现,即视图控制器记录用户的操作过程,分析操作结果,生成相应的过程定义;或由指定的应用程序(如 Agent 程序)根据运行时信息和对应的元模型自动生成。

**规则3** 一个处理器的输入输出转换逻辑抽象为一个标准的活动,一般分别对应活动集的第一个和最后一个,并各自对应一个或一组规则。

**规则4** 激活器和封装器抽象为一个标准的过程定义。一般把激活器的输入逻辑置为空,把封装器的输出逻辑置为空。

**规则5** 具体的功能执行由动作完成,一个动作执行由

Action Declaration 来描述,其中定义了动作对应的方法对象(如对象方法调用及相关的调用参数,一般通过反射机制来运行,更通用的可以是一个 Agent 程序或一个 Service 程序)、数据对象和执行者(一般是操作人员)信息。

**规则6** 为了降低过程模型的复杂性和庞大性,过程定义中涉及的数据对象应单独建模,主要包括多个应用数据模型(application data model)、环境数据模型(environmental data model)、用户数据模型(participant data model)及角色模型(role model)、规则模型(rule model)等,而在具体的过程定义中只包含了相应的引用声明。

在具体的实施过程中,首先应建立一组 XML Schema 来描述流程元模型,依此建立相应的元模型工件库和规则库,在此基础上实现 workflow引擎,该引擎主要包含解释模块、功能管理和定位模块、执行模块和监控模块。

#### 4 结束语

在 OA 系统的标准化建设过程中,建立具有行业甚至区域范围的流程元模型规范,是整个 OA 标准化建设过程中极其重要的工作。本文通过分析 OA 系统的生命周期特征,归纳了 OA 系统的流程模型,通过参考 WFMC 的 XPDL 标准建立了 OA 系统的流程元模型,并提出了相应的设计思想,对 OA 系统流程标准化建设作了有效的探索。

#### 参考文献

- 1 WFMC. Workflow Process Definition Interface——XML Process Definition Language[Z]. Document Number WFMC-TC-1025, 2002.
- 2 WFMC. Process Definition Interface——XML Process Definition Language[Z]. Document Number WFMC-TC-1025, 2005.
- 3 Hollingsworth D. The Workflow Reference Model 10 Years on Workflow Handbook[Z]. 2004.

(上接第91页)

区域方法可以减少相关边界的计算次数,进而降低系统的 CPU开销。图9是方位角  $\theta$  为  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$  和  $270^\circ$  时,开放区域查询和范围查询边界计算次数的比较。

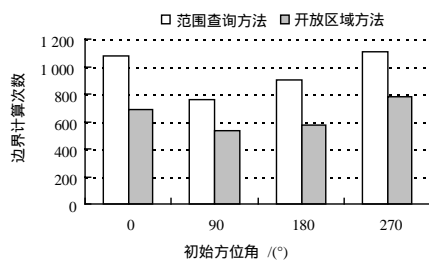


图9 开放区域方法和范围方法边界计算次数比较

可以看出,在 I/O 花费相等的情况下,开放区域方法的 CPU 开销更低。由上述实验可以看出,开放区域方法在性能上是优于传统的范围查询方法的。

#### 4 结束语

本文扩展了开放区域的基本抽象类,给出了基于开放区域扩展概念的定量方向关系查询技术的具体实现方法。该方法能够处理比定性分析更精确的查询情况,为应用领域提供

了有力的技术支持。同时,本文实现了基于该技术的通用性查询,降低了不同参考框架下应用不同查询方法的复杂度,并用实验证明了基于开放区域的技术相比传统范围查询方法在性能上的优势。可见,本文提出的基于开放区域的定量方向关系查询方法无论在功能上还是性能上都是一项具有前景的技术,降为空间查询技术的发展作出贡献。

#### 参考文献

- 1 Schmidt R. Various Views on Spatial Prepositions[J]. AI Magazine, 1988, 9(2): 95-105.
- 2 肖予钦,张巨,景宁. 基于 R 树的方向关系查询处理[J]. 软件学报, 2004, 15(1): 103-111.
- 3 Theodoridis Y, Papadias D. Range Queries Involving Spatial Relations: A Performance Analysis[C]//Proceedings of the 2nd Conference on Spatial Information Theory. 1995.
- 4 Shekhar S, Ravada S, Liu X. Spatial Databases: Accomplishments and Research Needs[J]. IEEE Trans. on Knowledge and Data Eng., 1999, 11(1): 45-55.
- 5 Tao Y, Papadias D. Selectivity Estimation for Predictive Spatio-temporal Queries[C]//Proc. of IEEE ICDE' 03. 2003.