柔性工作流模型在矿政管理系统中的应用

Application of Modeling for Flexible Workflow in a Mining Information Management System

王 颖 彭省临 戴塔根 鞠明辉 (中南大学 地球与环境工程学院 410083)

摘要:详细分析了柔性工作流系统的特性,定义了一个柔性工作流模型,并利用面向对象的建模方法,开发了一个分布式矿政管理系统模型。最后描述了其系统结构。

关键词:工作流 柔性 面向对象 矿政管理系统

1 引言

矿政管理系统是应用数据库技术、GIS 系统技术以及网络技术实现矿业信息的动态管理的一种系统。包括矿产资源多级勘查登记、采矿登记发证。此外,还包括日常业务流程的规范化,自动化管理以及信息共享空间信息的管理和查询。时至今日,该类系统在诸如 GIS 平台,数据存储、管理以及转换等方面做了广泛的研究与应用,取得了不少成果。但是,在诸多有关矿政管理系统的理论与实践中,对如何融入工作流技术、加强流程管理等方面还有待进一步研究。

本文详细的分析了柔性工作流系统的特性,定义了一个柔性工作流模型,在此基础上,利用面向对象的思想及抽象建模技术,开发了一个工作流管理系统模型,并在为某国土局的基于工作流的管理系统开发中得到应用,产生了较好的效益。

2 柔性工作流系统特性分析

"柔性"一词是针对目前系统在模型定义和执行中表现出的僵化、呆板而提出的。对传统工作流模型进行分析和研究,可以发现有以下几方面的不足[2]:

(1) 工作流模型被集中定义,无法很好地支持工作流的柔性;

(2) 工作流模型路由固定,不能根据实际情况的变化动态变换路由;

(3) 工作流执行的任务在模型定义之初就已固定,不能根据企业经营任务的变化及时更正;

(4) 工作流的任务不能动态划分,任务粒度尚不够细化;

(5) 相关资源在模型定义之初就被绑定,不能适应由于任务和环境变化而引起的资源动态分配。

究其本质,工作流模型其实是企业业务流程的形式化表示,主要包括活动节点和过程约束两部分。活动节点是构成工作流模型的细胞,过程约束则保证节点之间的控制关系和信息流向预期的方向目标流动。因而,工作流的柔性来自于其模型对柔性的描述能力。而要使工作流模型具有柔性,则要求活动节点和过程约束具有动态性,即可变换的特性,以及相互关系的结构动态性。

3 柔性工作流模型定义

要求工作流模型能够表达出企业业务流程的所有细节显然在实际开发当中并不可取。原因在于:①先前最周到详尽的设计也不可能将后来的所有变更考虑到,因而降低了模型的柔性;②过于繁琐的细节描述将影响他人对模型整体的把握;③很多的分支将永远不会执行,造成无端浪费[3]。然而,即使最复杂的业务流程也可简单的表示成若干节点串行并行的模型。本文利用面向对象建模的思想,将企业所有业务的共同部分抽象出来,用一抽象模型来描述,对于实际运行的具体流程,则通过在抽象模型上适当扩展节点和连接弧得到。

3.1 模型定义

(1) 定义 1。节点 N 表示业务过程中一个实际工作步骤,其形式化描述为一个四元组 $<n, t, R, L>$。其中,n 为节点名称;t 为节点类型(如起止节点、普通

节点)，符号如图1所示；R 为可参与到此活动的所有用户 r 的集合；L 为连接弧 l 的集合。

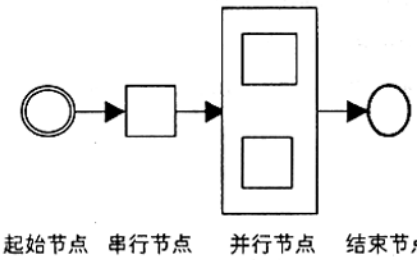

图 1　审批流程抽象模型

(2) 定义 2。连接弧 l(即节点之间的路由规则)为一个三元组 <a, b, f>。其中 a 为 b 的前驱节点；b 为 a 的后续节点；f 为 a 与 b 之间的对应规则。

(3) 定义 3。抽象模型 AM(Abstract Model)是一个四元组 <n, N－ , L－, m>。其中，n 为抽象模型 AM 的名称；N－为 AM 中节点的集合；L－为 AM 中连接弧的集合，m 为 AM 的管理者。

(4) 定义 4。扩展模型 EM(Extended Model)是在业务实际运行中对抽象模型的补充与具体化。它是一个九元组 <n, N*, L*, m, o, t, c, p, u>，其中，n 表示 EA 名称；N* 为 EA 中节点的集合；L* 为 EA 中连接弧的集合；m 为 EA 的管理者；o 为一布尔变量，标志 EA 中的对象 obj(在 EA 中运行的具体工作流)是否完成；t 表示 obj 的最长运行时限；c 表示 obj 所处的节点或节点集(若 obj 处于某并行阶段，则 obj 会同时位于多个节点中)；p 表示根据完成状态匹配的节点，在并行阶段，处于并行的节点在执行时并无先后顺序之分，此时，每执行完一节点，则应在 p 中作相应标注。u 为对应于当前节点集的处理人员。

3.2　建模实例

利用抽象模型，可以屏蔽掉各具差异的流程实例，将各种复杂业务流程简单地表达成串行或并行型模型。例如，以某国土资源局对探矿权申请的审批流程为例，包括：收件、初审、并行会审、复审、局领导签批、登记归档等六个步骤，其中还包括若干循环。如图 2 所示。

在实际开发该局政务管理系统的过程中，该局并不希望开发方将流程固定，他们指出两点原因：其一，他们的审批流程并不止探矿权申请一个，还有诸如采矿权新立、变更、延续等许多流程；其二，由于该局处于机构改革的前夕，各部门面临新的调整，各审批流程将有相应变化。

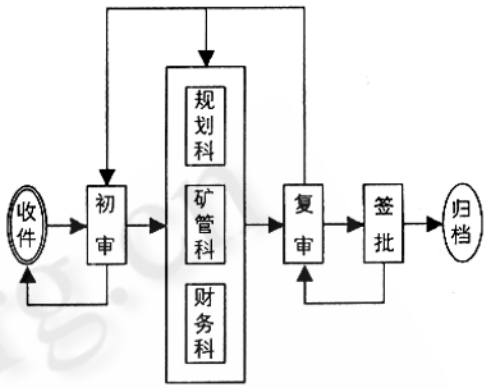

图 2　探矿权申请流程示意图

基于客户方的这些要求，我们首先定义了一个只含四个步骤(起点、中间串行节点、中间并行节点、终点)的抽象模型。各节点的属性值均为空，也没有循环，如图 1 所示。

以此抽象模型为模板，根据实际需要，由流程管理者设定具体步骤、节点名称、节点参与者(用户)、流程走向等诸多事宜，一个具体流程得以形成。

以下是生成探矿权申请流程的实现算法。

3.3　模型的执行算法

3.3.1　管理者启动审批流程的抽象模型 AM

```
Abstract class AM_ExamWorkflow{
public String Name;
public Node node[ ];
public String manager;
public Node = new Node[5];
public AM_ExamWorkflow( ){
.....
 }
// 设置节点路由
public abstract void Set_Route ( String Node_
name ){
.....
 }
// 获得节点路由
 public abstract void Get_Route ( String Node_
name){
```

```
        ……
        }
        }
    /* 定义节点类 */
    public class Node{
     String name;
     String type;
     String user;
    public Node ( String n, String t, String u) {
     name = n;
     type = t;
     user = u;
     }
     }
```

3.3.2 生成扩展模型——探矿权申请流程 EM

```
    EM_ProspectWorkflow extends AM_ExamWorkflow
{
    public Node node[ ] ;
     //EA 实例(具体工作流)是否完成
    public Boolean IsOver;
     //EM 实例的运行时限
    public Int TimeLimit ;
     //EM 实例所处的节点集
    public String CurPU ;
     //根据完成状态匹配的节点
     public String PracticPU ;
     //环节(集)的处理人员
     public String User ;
     node = new Node[8];
     public EM_ProspectWorkflow ( ) {
     ……
    }
    /* 给 Node_name 赋当前节点名,函数根据模型,自动设置输出节点和输出条件。 */
        public void Set_Route ( String Node_name){
        ……
        }
    /* 给 Node_name 赋当前节点名,函数根据模型,先检查输出条件,符合条件则返回输出节点,否则返回提示信息 */
        public Sting Get_Route ( String Node_name ){
        ……
        }
    /* WorkflowName 中输入实例名,函数自动返回自己在流程中所处的节点(位置) */
        public String find_PracticPU ( String WorkflowName ){
        ……
        }
    /* 重新设置实例当前所在节点和匹配节点 */
        public void Set_CurPU ( String WorkflowName )
{
        ……
        }
    }
```

3.3.3 生成探矿权申请实例 app

3.3.4 执行实例

假设某探矿权申请实例位于“并行会审”阶段。因为并行节点的执行顺序是任意的,故此处假定矿管科首先执行该步骤。并行会审之前是初审,初审如果得到通过,则:CurPU = “规划科、矿管科、财务科”,PracticPU = CurPU。

```
  Public class Execute_ProspectWorkflow {
     public static void main (string arg[ ]) {
     EM_ProspectWorkflow app = new
  EM_ProspectWorkflow ( ) ;
/* 找到 app 所处位置 CurPU = “规划科、矿管科、财务科” */
app. PracticPU = App. find_PracticPU ( app. name ) ;
  //在用户控制下执行流程执行相关操作
     …… ……
  /* 重新设置当前节点和匹配节点:CurPU = “规划科、矿管科、财务科”, PracticPU = “规划科、财务科” */
  app. Set_CurPU ( app. name ) ;
  }
  }
```

当规划科、财务科均执行完毕后,PracticPU = “”。Set_CurPU 函数将流程向下驱动,此时,设置 CurPU = “复审”,PracticPU = “复审”。

4　系统体系结构及特点

4.1　系统体系结构

图 3 显示的是笔者设计的柔性工作流管理系统的体系结构。这是一种分布式结构。即数据库服务分布和系统服务分布。

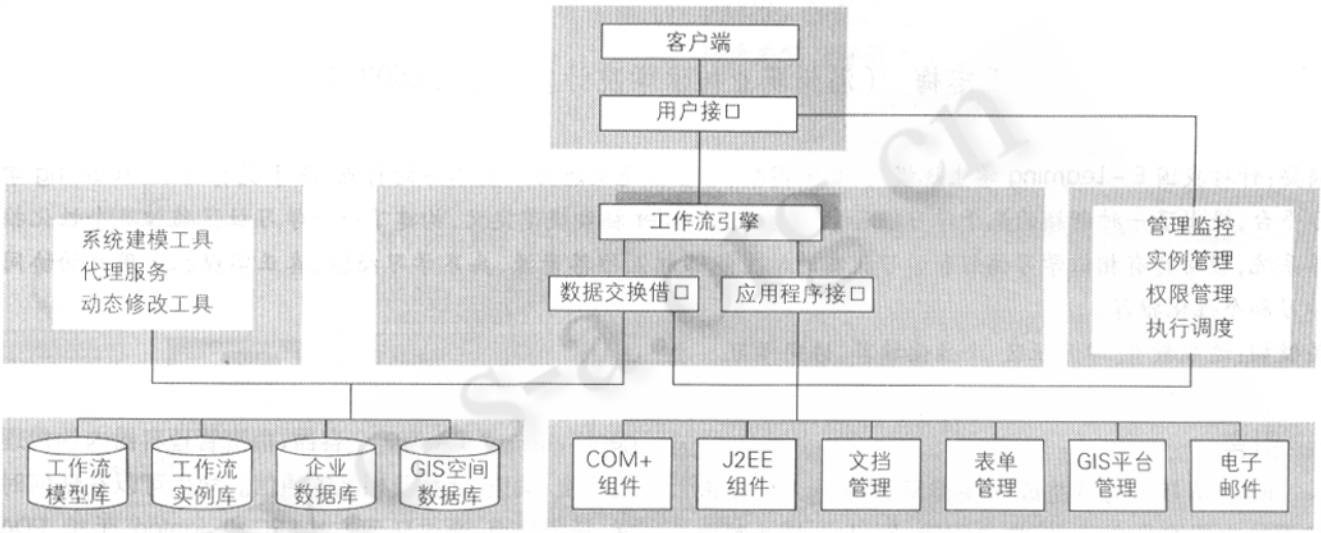

图 3　某国土局矿政管理系统体系结构图

在本系统中，需要多个服务器：工作流引擎服务器、数据库服务器、GIS 服务器及其它服务器。整个系统包括建模、代理服务、模型动态修改、用户权限管理、实例管理、监控管理、执行调度管理等多项功能。各服务器同时工作，协同调度，实行数据共享，避免了数据重复。

4.2　系统主要部分功能及特点

(1) 系统建模工具。以抽象模型为基础，提供可视化界面用以建立企业业务模型；

(2) 动态修改工具。流程在运行时往往遇到许多不可预见的错误或问题，这些情况很难在流程模型中定义。动态修改工具允许用户动态修改流程实例，使之不完全按规定模式流转。

(3) 实例管理器。流程在运行中会产生各种实例数据，实例管理器在流程初始化时，根据模型定义在数据库中生成实例数据。

(4) 代理服务。由于多 Agent 技术研究相互独立而又有联系的个体在分布式环境下的协调合作，它从本质上非常适合描述工作流模型中活动之间及子过程间的协作关系。在本系统中，每个 Agent 被设计成 CORBA 对象的智能化封装，Agent 将其内部 CORBA 对象提供的服务函数映射为自己的私有方法，并通过代理的协作知识，确定自己所管理的活动的执行次序。

(5) 权限管理。每个操作员被动态赋予相应权限，操纵不同节点及企业资源。

5　结语

计算机网络通信技术和工作流管理技术的出现和发展，为企业、政府的组织结构重组，优化管理模式提供了良好的软硬件支持。然而，传统的工作流建模方式往往无法满足政府企业部门内工作流程动态改变所提出的柔性要求。本文在深入分析工作流柔性特点的基础上提出了一种基于面向对象方法的建模方法，并构建了一个分布式的系统结构。

工作流建模理论与实践的不断成熟，加上信息技术其他领域的进步，将促使工作流管理系统不断完善和推广！

参考文献

1　李毅东、姜新文，《一个基于 Internet 的柔性工作流驱动的办公自动化解决方案》[J]，计算机与现代化，2004，3(6)：86－88.

2　曾炜、阎保平，《工作流模型研究综述》[J]，计算机应用研究，2005，22(5)：11－13.

3　何鹤立、倪小平，《工作流管理系统的柔性技术》[J]，计算机工程，2004，30(6)：63－64.