分类号：TP319　　密　级：

东北石油大学

Northeast Petroleum University

硕士研究生学位论文

论文题目：基于 RBAC 的柔性工作流研究与应用

硕 士 生：龙 晓 春

指导教师：赵建民　副教授

学科专业：计算机应用技术

2012 年 3 月 25 日

Thesis for the Master degree in Engineering

Research and Application of Flexible Workflow Based on RBAC

Candidate: Long Xiaochun

Tutor: Zhao Jianmin

Specialty: Computer Applied Technology

Date of oral examination: 25th March, 2012

University: Northeast Petroleum University

学位论文独创性声明

本人所呈交的学位论文是我在指导教师的指导下进行的研究工作及取得的研究成果。据我所知，除文中已经注明引用的内容外，本论文不包含其他个人已经发表或撰写过的研究成果。对本文的研究做出重要贡献的个人和集体，均已在文中作了明确说明并表示谢意。

作者签名：龙晓春　　日期：2012.3.28

学位论文使用授权声明

本人完全了解东北石油大学有关保留、使用学位论文的规定。学校有权保留学位论文并向国家主管部门或其指定机构送交论文的电子版和纸质版，允许论文被查阅和借阅，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文，可以公布论文的全部或部分内容。东北石油大学有权将本人的学位论文加入《中国优秀硕士学位论文全文数据库》、《中国博士学位论文全文数据库》和编入《中国知识资源总库》。保密的学位论文在解密后适用本规定。

学位论文作者签名：龙晓春　　导师签名：

日期：2012.3.28　　日期：2012.3.28

基于 RBAC 的柔性工作流研究与应用

摘　要

随着科技的进步、经济全球化的发展，工作流技术在提高企业生产与服务效率方面所起的作用越发重要。面对现代企业业务过程的精细化与流程的复杂化，传统工作流系统遇到了前所未有的挑战，其局限性日益突出。迫切需要一个灵活性与自适应性较高的工作流管理系统来满足其动态业务需求，因此，柔性工作流成为了企业以及相关学术领域研究的新热点。

本文详细地分析了工作流参考模型、工作流管理系统体系结构，对 RBAC 技术进行了全面地研究，重点分析了工作流系统中用户-角色-权限三者的关系，构建出用户权限管理模型，并将其应用于工作流系统用户授权。从流程过程柔性管理与用户柔性授权管理两点出发，开发了基于 RBAC 的柔性工作流系统，在实际系统中得到了较好的检验与应用。

首先，分析了 RBAC 技术在用户权限管理方面优势，以工作流系统中用户授权为切入点，根据工作流系统的特性，构建出基于 RBAC 的柔性工作流模型。具体分析了如何实现用户-角色-权限的授权关系，依据最小权限原则、最小权限时限原则、职责分离原则、用户负载均衡等约束条件使合法用户在恰当的时间拥有合法的角色实现对流程客体资源的灵活操作。提出了推式分配策略与拉式分配策略，实现系统用户权限的柔性管理。

其次，在工作流相关理论研究的基础上，提出了流程过程柔性管理的具体实现方法。以节点与路径为出发点，研究工作流过程的柔性。将系统的主要对象进行有效划分，建立起类似数据库映射机制的基空间和元空间，系统依据元对象协议把对元对象的各种处理映射到基空间中的具体过程实例上去，实现了流程柔性定制与流转方向的灵活控制。以流程实例具体情况为依据，保证流程实例顺利运行且实现指定目标。

最后，完成了基于 RBAC 的柔性工作流系统的开发，将其应用于大庆油田试油试采分公司综合信息管理系统中，实现了业务流程的创建、实例化、跳转及追踪监控等功能，提高了系统在业务流程过程管理和用户权限管理上的柔性。

关键词：RBAC，柔性，工作流，流程过程管理，用户权限管理

Research and Application of Flexible Workflow Based on RBAC

ABSTRACT

With the progress of science and technology, the development of economic globalization, enterprises are facing increasingly fierce competition in the market, the role of workflow technology in the aspects of production and service efficiency is more important. However, due to the refinement and complexity of modern enterprise business process, workflow systems are experiencing unprecedented challenges. The limitations of the traditional workflow systems are increasingly prominent. To meet the complex and dynamically changing business needs, flexible workflow system with higher adaptivity and flexibility has become to be a new hot spot to the enterprise as well as academic field.

In this paper, it gives a detailed analysis of the Workflow Reference Model and architecture of the workflow management system. It studies comprehensively on the RBAC technology, does a focused research on the relationships of users-roles-permissions in the workflow system, and builds a model for users' rights management, and then extends the model to user authorization of workflow system. From the flexible flow process management and flexible user authorization management, a flexible workflow system based on RBAC has been developed and been well verified in the actual system applications.

Firstly, in this paper, it analyzes the advantage of RBAC technology in the users' right management, and takes the user authorization of system as a entry point, to build a model of user rights management to meet the needs of workflow system. Also, it has given a detailed analysis of how to implement dynamic licensing relationships of user-role-permission. According to the principle of least privilege, the time constraints of least privilege, the principle of separation of duties, and the user load balancing constraints, it makes legitimate users to have a legitimate role to achieve a flexible operation of the object resources in the process at the right time in the paper. And it puts forward the pushing-type and pulling-type allocation strategy, and realizes the users' right flexible management.

Secondly, based on the theoretical studies of the workflow, a specific implementation of the flexible management of the flow process is proposed by the paper. Nodes and paths are the starting point of the research, flexible process customization and the flexible control of the process flow direction has been achieved. It divides the main target of the system effectively, and constructs a similar database mapping mechanism of base space and meta-space. Based on the meta-object protocol, the system finishes a variety of treatments of meta-object mapping to the specific process instances of the base space, and achieves flexible customization of workflow process and flexible control of the flow direction. According to the

actual situation of every instances of workflow, it ensures the process instance implement smoothly and achieves the specified goal.

Finally, the development of a flexible workflow system based on RBAC has been completed by the paper, and it has been applied to an integrated information management system of Daqing Oilfield. Also, it has achieved some important functions such as the creation of a business process instance, jumping, monitoring and so on, and the system flexibility both in the management of business processes and user rights has been improved.

Keywords: RBAC, flexibility, Workflow, flow process management, users' right management

创新点摘要

为提高工作流管理系统的灵活性、动态适应性，本文设计和实现了基于 RBAC 的柔性工作流模型，并在实际系统应用中得到有效的验证。本文的创新点如下：

1. 本文以用户授权为切入点，将 RBAC 技术引入到工作流系统中，构建了基于 RBAC 的柔性工作流模型。将推拉式理论引入到用户权限管理模型中，提出了用户被动接受任务的推式策略与用户主动获取任务的拉式策略两种分配方式，并在系统中混合使用了两种策略，实现了系统的用户权限柔性管理。

2. 从节点和路径两个角度研究工作流过程的柔性。将系统的主要对象进行有效划分，建立起类似数据库映射机制的基空间和元空间，系统依据元对象协议把对元对象的各种处理映射到基空间中的具体过程实例上，实现了流程的灵活定制和流程执行路径的灵活选择。

目　录

第一章 绪　论

1.1 课题研究背景

计算机的高普及与互联网技术的飞速发展，促使信息资源在现代企业中呈现出异构分布、松散耦合的特点，传统的集中式信息处理方式已满足不了需求，保证相关任务的高效运转并接受严密的实时监控是一个必然趋势[1]。计算机支持的为实现业务流程自动化的工作流技术日益受到学术界与企业界的重视。由于复杂多变的客观环境，人们自身经验、知识、技能、需求等各方面的不断发展以及企业本身在发展过程中组织机构、政策文化、企业定位等变动，业务工作受到来自外部和内部两方面因素的影响[1,2]，因此就要求工作流技术也应该具有根据实际情况进行调整的能力，即具有柔性。传统工作流系统大多采用静态、固化的表现形式，在流程定制开发、流程建模以及实例运行等多个环节灵活性、适应性不够。柔性工作流是顺应发展需要而产生的，能够更加确切的描述企业业务过程，满足企业工作的实际需要[3]。柔性已成为衡量工作流系统性能的重要指标之一，吸引了越来越多的专家学者、研究机构与企业组织在展开深入研究。

目前，较为广泛的对流程柔性的研究基本上从活动入手，提高流程灵活性也仅从活动及活动间的关系来考虑[4]，活动的柔性主要体现在流程节点和路径方面的灵活性。活动是流程的一个重要部分，与此同时，我们应该重视活动的执行人。从人和活动的关系来看，人是活动的执行主体，流程中遇到的异常很多可归因于角色在过程中的交互与变化情况，用户动态授权在实际业务过程中是一个非常重要的问题，其灵活程度是工作流系统柔性的一个重要反映。几种较为常见的访问控制技术：自主型访问控制（DAC）、强制型访问控制（MAC）和基于角色的访问控制（RBAC）[5]。相对前面两种访问控制技术存在的不足，RBAC 技术的在系统访问控制、用户授权方面有一定的优势。结合工作流系统的特殊性质，本文综合考虑了流程实例中活动柔性定制与角色柔性分配两个方面，提出了基于 RBAC 的柔性工作流系统。

1.2 国内外研究现状

工作流技术自 20 世纪 70 年代产生以来，经历几十年的发展，如今已从最初支持办公自动化发展到制造业、金融业、电子政务、医疗行业等各行各业。目前，对于工作流技术的研究和工作流相关产品的开发进入了相对繁荣阶段，学术界与商业界都致力于寻求更多的相关先进技术，以促进工作流系统发展完善。在所取得的研究成果中，几种比较著名的工作流系统：

1. 基于 CORBA 的分布式工作流管理系统：ORBWork[6]

美国 Georgia 大学 LSDIS 实验室研发的一个著名的工作流系统：METEOR（managing end-to-end operations），一个能自适应的支持大规模复杂应用，且能保证系统中所有应用能运行在异构环境下的工作流管理系统。ORBWork 是 METEOR 项目的一个完全分布的工作流执行系统，以 Orbix 作为底层的通信支持，采用基于分布对象处理技术（CORBA）

实现系统的互操作和数据源的封装。METEOR 主要分三个开发阶段：1991 年启动研究的 METEOR1，侧重电信方面的应用；METEOR2 从 1994 年到 2000 年，侧重医疗和电子商务的应用；MET EOR3：目前正在进行的研究项目，侧重电子服务（e-service）和生物信息学的应用。

2. 基于永久消息队列的分布式工作流管理系统：Exotica/FMQM[7]

Exotica 是 IBM Almaden 研究中心所进行的研究项目，是针对企业工作分布性而提出的一种完全支持分布环境的工作流管理系统。流程各个执行节点都相互独立，他们之间通过持久消息（persistent messages）的方式来保存以及交换流程执行过程中的相关信息，完全改变了集中式管理结构中单点失败导致整个工作流系统不能正常运行的状况。

3. 基于分布式主动数据库技术的工作流管理系统 WIDE97[8]

WIDE（Workflow on Interlligent and Distributed database Enviroment）采用分布式数据库和主动数据库技术，由西班牙和意大利的几位学者联合开发。WIDE 的工作流模型包含了组织模型、信息模型和过程模型。三层体系结构：底层数据库采用 ORACLE 关系型数据库；第二层为数据库屏蔽层 BAL，向上层提供面向对象数据库的操作接口；最上层是服务层，包含一些必需的事务管理规则，也为系统用户的访问提供接口。

企业环境对工作流系统的性能提出了更多新要求，首要问题就是系统的灵活性与自适应性。目前研究成果中，比较典型的柔性工作流系统如下：

1. 荷兰埃因霍恩大学（Eindhoven University of Technology）的 Will vander Aalst 在 Petri 网的基础上提出了工作流网 WF-net，定义了工作流网的基本组件和转移机制，用变迁来表示活动，库所表示活动的使能。Petri 网是一种比较常用的建模方式，能很好的表达并发、资源共享、操作并行等相关业务需求的流程模型[9]。vander Aalst 流程运行时的动态修改操作分为：扩展、代替和排序三种。扩展是增加新的任务到原有流程中，代替是指用一个新的任务或者活动替代原有一个任务或活动，排序则指对原有流程中各任务的结构顺序重新排列[10]。

2. Software-Ley GmbH 等人开发出的工作流管理系统 COSA，从体系结构来看与工作流管理联盟提出的参考模型极为相似。COSA 的主要特点有：首先，支持过程的版本控制与过程重用，但对活动重用不支持。其次，支持过程的动态性，能够在流程运行时对过程进行修改。另外，COSA 是面向多服务器的分布式结构，系统开放性很好。COSA 在理论上对柔性的研究较为清晰，但在实际工作流产品中的应用程度并不充分，尚待做进一步的努力和探索[11]。

3. Wowww 提出了条件有向图模型（CDG），是一种基于活动网络图而构建的工作流模型。其中，网络节点代表流程过程的活动，边连接下一个即将被激活并执行的活动，而有条件规约的流程任务中，有向图边上直接提供条件的相应定义。CDG 工作流系统 W 为一个三元组<N，A，F>的表达形式，整个系统为一个有向图形式，图中各结点代表了流程任务各执行环节[12]。这种流程有向图与一般的有向图存在一定的差异：即图中的各条有向边都与一个逻辑表达式和命名表达式集相对应。CDG 是面向数据的有向图，

图中的边只负责数据对象在边两端的活动按正确方向传递，而对于信息的处理则有活动节点根据输入边的不同决定调用何种函数来完成。条件有向图的原理更为复杂，为了提高约束条件复杂的工作流系统的效率，增大业务活动处理交互的灵活度，研究人员利用WPR（Workflow Process Re-engineering）从分析活动间的数据依赖关系即一般性的角度入手，给出了具有通用性的工作流程优化方法对条件有向图进行了优化[13]。

4. 文献[14]于 1994 年推出的跨行业自动化工作流程管理系统 Ultimas，已经逐渐发展成为提供全方位过程管理的解决方案。其主要产品 Ultimas Workflow Suite 5.0 是一款开放式平台，采用 Web Services 及.Net 技术实现[14]。系统包含的主要模块有：过程模型分析、过程架构、企业组织图、过程分析报表和过程管理等，从而实现对企业过程的生命周期管理。Ultimus 的主要特点有：具有一定柔性的过程定制方法；提供过程监控功能以及分析的功能；具有异常处理功能；提供开放式的整合环境以及相关接口用于实现与 ERP 等系统的整合。

国内对工作流的研究相对滞后，在理论方法与实际系统研究上都还需要更加深入和广泛的探索。近年来，国内企业已经注重采用多种先进技术进行系统整合，政府也开始鼓励电子政务、办公自动化系统的发展，这些早期的工作流系统与企业中的 MIS 系统有一个很大共同之处：都需要处理大量“审批流”[15]。总得来看国内工作流系统主要呈现出几个方面的特点：

首先，审批流是国内工作流系统的一个主要定位，而这类系统大多与企业的组织结构有着密切关系，以组织服务为主导的流程应用成为特色。因此，保证各环节执行人的对流程任务的合法操作非常重要，诸如“会签”、“回退”、“撤销”、“跳跃（也称自由流）”等逐渐成为国内一些特色的工作流运转模式[16]。

其次，工作流产品方面的特点，从工作流技术在国内的产生和发展过程决定其工作流产品比较紧密的与特定行业绑定。如主要定位于电信行业的西安协同的 SynchroFlow；有生博大的 RiseOffice 主要偏向于电子政务系统中的审批流；注重协同领域的上海东兰的 LiveFlow；信雅达的 SunFlow 更偏重于金融行业等等。

最后，国内工作流应用普及程度太低。尽管在政府协同办公、电信行业、金融、医疗、财政、电子政务等领域的一些管理系统中工作流技术的优势越来越突出，但在石油、化工、铁道等一些传统行业中，由于其本身信息化普及程度就有待提高，因而对工作流系统的应用还远远不够。

1.3 课题研究目的与意义

工作流技术是实现计算机辅助协同工作的有效技术，工作流管理系统能够实现流程的全部或者部分自动化管理，实现合理配置企业中的人力物力资源，提供流程实时监督和审查，从而大大提高了工作效率。因此，如何才能使工作流管理系统在具体流程实例中足够灵活以适应实际业务过程的变化，符合市场敏捷性的要求，即如何提高工作流管理系统的柔性，成为企业亟待解决的问题。

本文以用户授权为结合点引入 RBAC 技术，并根据工作流系统的特性，进行柔性工作流管理系统的研究和应用。用户授权主要内容是用户-角色-权限三者的映射关系，RBAC 技术是一类较为成熟的权限管理技术，能很好的符合现代企业组织管理模式的需求。RBAC 技术的思想是：在用户和权限之间设置一个相对稳定的中间机制角色，用户不再直接与权限关联，而是通过角色间接的对系统的权限资源进行访问控制，降低了用户授权管理复杂度，从而实现企业业务管理效率的提高[17]。企业组织人员引起的变化相对其他物化的影响因素所能带来的变化，更具复杂性与灵活性。因此，需要对用户-角色-权限进行合法地、灵活地分配与管理，在复杂多变的实际业务过程中，保证主体对流程各种客体资源的灵活操作，实现流程实例顺利执行。

柔性工作流系统使企业信息管理系统更加自动化、个性化、灵活化，本课题从角色访问控制的角度出发，对如何提高工作流系统的柔性进行研究，构建基于 RBAC 的工作流模型，开发出具有较高柔性的工作流管理系统，适应企业业务流程灵活管理的需求，因此，具有重大的理论研究意义与实际应用价值。

1.4 本文主要研究内容

本文共分为五章，论文的组织结构如图 1.1 所示，具体内容安排如下：

第一章绪论，主要阐述了本课题的研究背景，研究目的与意义，研究历程与发展现状。并以图示方式清晰的描述本论文的研究内容及组织结构，为后文的研究工作提供方向性的指导。

第二章工作流技术的概述，介绍了现阶段关于工作流较权威的概念、WFMC 提出的工作流参考模型及在此基础上得出的工作流管理系统体系结构。详尽的分析了柔性工作流的基本概念，柔性工作流起源、发展及研究现状，并描述几种柔性实现理论与方法。

第三章 RBAC 技术的分析，介绍了目前访问控制技术中几种具有代表性的模型，对比分析其优缺点，深入剖析 RBAC 技术在工作流管理系统中应用的可行性与优越性。阐述 RBAC 技术的发展历史，详尽描述了其模型元素，对 RBAC 模型在应用中存在的一些问题进行分析并提出一些改进办法。

第四章基于 RBAC 的柔性工作流模型研究，依据工作流系统的特性，构建出基于 RBAC 技术的用户权限管理模型，给出了模型元素的形式化定义，并将其融合到传统的工作流模型中，形成基于 RBAC 的柔性工作流模型。分别从流程过程与用户权限管理两个方面分析了如何提高工作流系统的柔性，并给出了相应的实现方法。

第五章基于 RBAC 的柔性工作流系统的设计与应用，以大庆试油试采分公司的实际项目为依托，基于所构建的柔性工作流引擎，并结合项目的实际需求，对相应的模型架构进行了部分调整和改进。重点分析了系统中流程动态管理和基于 RBAC 的用户动态管理两个模块，讨论了流程在执行路径与用户权限管理两个方面柔性的在系统中的具体实现。

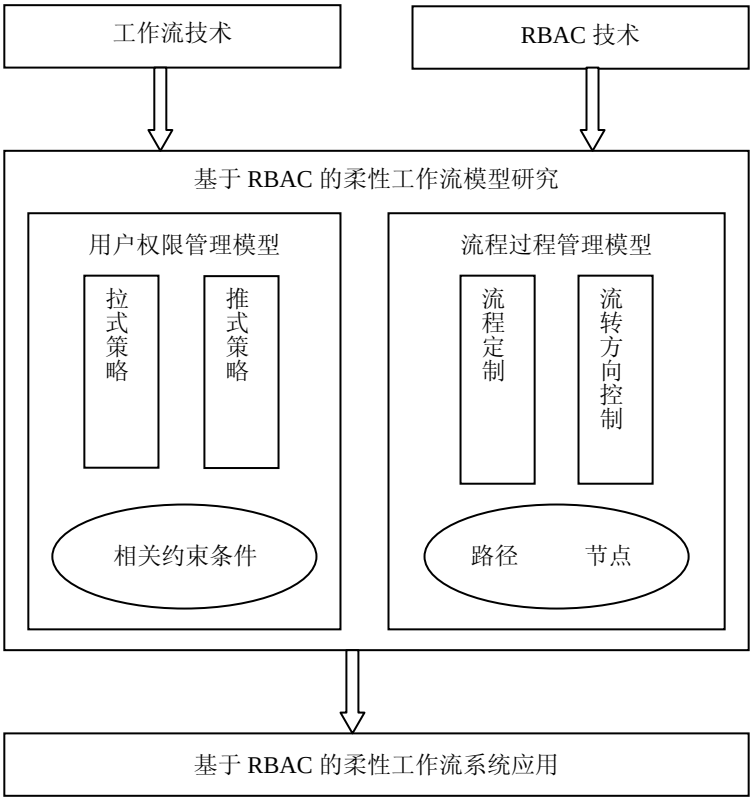
工作流技术
RBAC 技术
基于 RBAC 的柔性工作流模型研究
用户权限管理模型
拉式策略
推式策略
相关约束条件
流程过程管理模型
流程定制
流转方向控制
路径
节点
基于 RBAC 的柔性工作流系统应用

图 1.1 论文组织结构图

第二章 工作流系统概述

2.1 工作流的定义

对工作流的定义，不同研究者提出了不同的说法，目前为止还没有得出一个完全一致的定义。专家学者们分别从不同的角度与着重点对工作流的概念进行了描述，列举如下一些有代表性的定义。

1. 工作流管理联盟 WFMC 的定义[18]：工作流是企业经营过程能够完全或者部分自动化的执行，给出一系统的过程规则，保证系统不同执行人之间可自动传递和执行与流程相关文档、信息或者任务，从而实现流程最终的业务目标。

2. IBM Almaden 研究中心的定义[19]：工作流定义了整个经营过程完成所需要用到的各类参数，它是一种用计算机来表示经营过程的模型。这些模型参数包括对过程中各个独立执行步骤的定义、不同步骤之间的执行次序和相关条件、建立各类数据流、流程步骤的参与者以及流程活动执行时所需调用的应用程序。

3. Giga Group 的定义[20]：工作流是经营过程中可运转的部分，包括任务的顺序以及任务的执行者、支持各任务相关信息流、监控并评价任务的执行情况及报告机制。

4. Arait Sheth 提出的工作流定义[21]：工作流是涉及到由不同处理主体来完成多任务协调执行的活动。每一项任务都需要对其所应该包含的工作作出定义，这些定义的形式多样，可以是一段意思完整的文本描述，也可以是表格、消息以及计算机程序等。任务执行的主体一般指人，也可是计算机系统（如一个智能的计算机应用程序或者数据库管理系统）。

5. W.M.P. Vander Aalst 的定义[10]：工作流由一系列相关工作任务组成的偏序集。集合中这些工作的序列存在多种关系，如要使工作任务 X 与 Y 满足 X < Y，必须满足的条件是：当且仅当任务 X 在 Y 执行开始前就已经处于就绪状态。

6. 文献[22]给出了这样的工作流定义：工作流是业务经营过程计算机化的实现，在先进的计算机环境支持下，为了实现业务流程集成和流程自动化，而建立起由工作流管理系统执行的业务模型。

综上所述，这些工作流定义主要从以下几个方面反映了经营过程的问题：

- 从结构上定义工作流，即什么是经营过程，主要包括哪些活动或任务；
- 从控制流与信息流上给出定义，即怎么做：各活动执行的条件、过程规约以及过程中交互的资源；
- 从组织角色来定义工作流，即由谁来做，明确执行者是人或计算机应用程序；
- 从过程执行结果角度给出定义：即做得怎么样，通过工作流管理系统对流程运行情况进行实时监控，以保证流程安全、稳定、顺利的执行。

2.2 工作流模型

工作流模型是对经营过程的抽象表示，工作流需要在计算机环境下运行，首先需

要将经营过程转化为计算机可处理的形式化的定义。工作流模型为用户提供建模时所需要的各种组件及元素，理想的工作流模型能够灵活地适应用户在建模过程中所提出的各种要求。如今，人们虽然已构建出很多健壮的、有意义的工作流业务模型，但现实情况的复杂多变性使其距理想状态还是存在一定差距。

2.2.1 WFMC 参考模型

工作流管理联盟 WFMC 于 1994 年 11 月 29 日发布了工作流系统参考模型（Workflow reference model），这一模型对工作流系统的相关概念给出了详尽的分析，并且描述了 WFMS 的六大主要组成部分、各部分的主要功能以及各部分间的五个接口[23]。如图 2.1 所示。

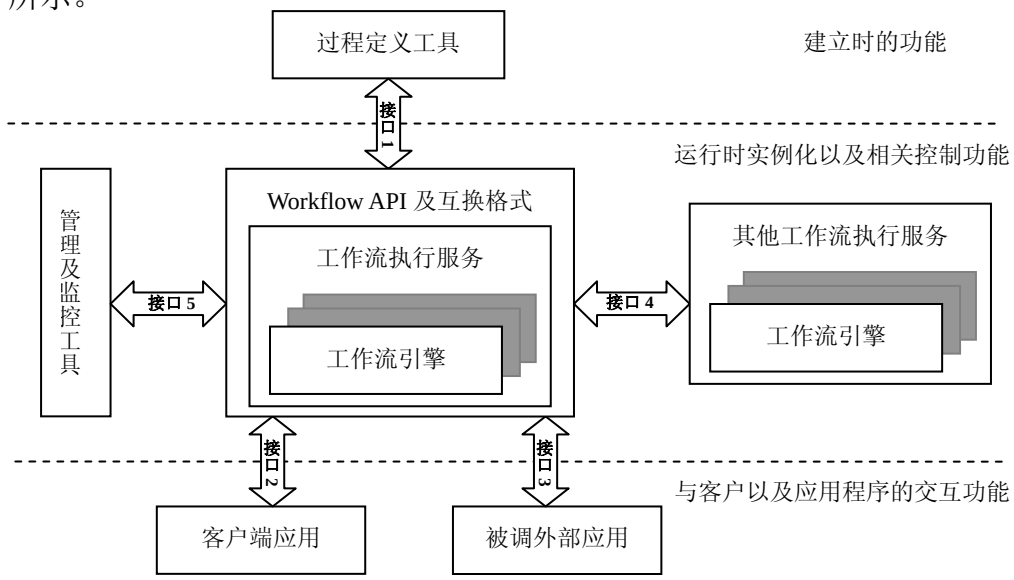

图 2.1 WFMC 工作流管理系统参考模型

工作流管理系统的六大组件[24]：

1. 过程定义工具：在工作流模型建立时，其主要功能是提供给用户一种对实际业务流程进行分析、建模的手段，生成一段计算机能识别的形式化描述，即输出一个能被工作流执行服务解释并执行的过程定义。

2. 工作流执行服务：它是 WFMS 的核心部分，为工作流实例提供运行时服务环境。工作流引擎是工作流执行服务的操作主体，是执行系统的内核。工作流建模完成后，由执行服务借助工作流引擎激活并解释过程定义的全部或者部分，并以定义内容为依据，控制流程实例创建、激活、挂起和终止等操作；根据触发条件与实际参数，实现流程实例活动状态的转换；提供访问外部应用程序的接口、维护工作流数据和工作流相关数据；按工作流执行节点、任务执行顺序，实现过程活动之间的游历，生成有关的工作项并通知合法权限的用户处理任务等。

3. 工作流客户应用：它主要是针对过程实例运行过程中需要人工参与的活动，是一种提供给用户进行操作的机制。工作流执行服务在这些节点处，会依据定义以及实际信息参数生成很多需要人工处理的任务，通常被称为工作项（包括待处理的数据对象以及

处理时的一些约束条件等）。而各合法用户都与一个工作项列表相对应，即这一角色或岗位所需要完成的所有工作。如何保证客户应用高效、高灵活度、高质量的要求，是工作流管理系统先进性的一个重要衡量标准。

4. 被调用应用程序：主要是指过程实例的运行过程中无需用户参与的一些任务操作。工作流执行服务直接调用应用程序，在流程定义中给出这类应用程序的相关详细信息（如类型、地址等），提供建立会话、双向管理活动以及对应用数据进行处理等几类服务。目前采用的几种调用方式包括本地过程调用、远程执行调用、ORB、用一个标准的接口同执行服务进行交互的应用代理等。

5. 其他工作流执行服务：大型的工作流管理系统，工作流过程可能由多个子流程构成，仅仅由单个工作流引擎负责运行时各种调度，可能响应较慢，造成工作效率的降低。有些子流程甚至本身就需要不同的工作流执行服务来提供异质的运行时环境才能保证流程的顺利进行。为满足系统对服务的这类要求，就必须两个或以上的工作流执行服务交互配合、协同工作。

6. 管理及监控工具：它的功能是监控及管理工作流管理系统中过程实例的状态，这些管理主要包括角色及用户管理、资源控制、审计管理、流程过程管理与过程状态控制等。管理员通过此工具可以及时获悉工作流实例的运行情况，如过程实例信息、活动实例信息、工作项信息及远程操作信息。

工作流执行服务是工作流管理系统的中心，它与其他五个主要部分之间的交互形成了五个接口[24]：

接口 1：工作流建模工具和工作流执行服务之间的接口，为工作流过程定义信息互换提供了标准的格式及调用规则。包括对工作流模型的分析和读/写操作，实现流程定义阶段与执行阶段相分离，用户可以选择不同的工作流建模工具和工作流执行环境，还可以使各类工作流产品合作共同提供无差异的运行服务环境。

接口 2：客户应用程序 API 是最重要的接口规范。介于工作流执行服务器与客户应用间，它规定客户端应用与工作流服务之间的功能操作方式以及此 API 中各种操作的命名规范。WFMS 通过此接口提供用户交互界面，可实现用户对过程实例运行情况的干预（包括过程控制、活动控制、过程状态、活动状态、处理工作项列表等操作）。

接口 3：与直接调用的应用程序间的接口规范。在过程运行执行时工作流引擎直接触发启动一些应用程序，应用程序直接同工作流执行服务交互，以实现一些专门的应用需求。

接口 4：工作流管理系统之间的互联的接口，工作流引擎之间需要进行数据交换（如顺序调度，同步处理等）。企业间的大型工作流应用需求，需要实现异构 WFMS 的良性互连，有必要研究更为有效的互连协议，定义互联模型（包括互连一致性级别以及操作元素集等内容）。

接口 5：工作流服务和工作流管理工具之间的接口，该接口规范详细描述了工作流执行时活动的各种相关数据信息，并在必要时调整流程实例的运行。

2.2.2 工作流管理系统发展趋势

工作流管理系统是一套严格定义的软件系统，是工作流技术支持的综合性计算机信息管理系统，通过对各种相关信息和资源的合理调度来协调业务过程中的各项活动及任务，高效实现业务目标[25]。为提高系统管理的可伸展性、灵活性、实用性、健壮性，工作流管理系统的呈现出下述几个方向的发展趋势。

1. 智能化的工作流管理系统[26]

人工智能（AI）提出的学习过程并不是一开始就给出系统中所有信息的定义，而是随人机交互过程的进行，启发式或扩张性的实现系统信息的动态增长。作为群件系统典型代表的 WFMS，从建模到执行以及遇到变更与异常情况时动态调整等，都要求 WFMS 本身具有类似 AI 提出的这种学习能力，即从实际流程中准确提取各种参数（如数据的处理方式、处理结果以及时间、人员、资源等）得出相应的过程定义。业务过程运行阶段，受内外部各种因素的影响，静态模型往往存在一些不足，因此流程模型需要在流程执行时根据实际变更而动态改变。

2. 协同同步的工作流管理系统[27]

目前 WFMS 的工作原理是某一环节上的用户完成工作后，将处理结果自动传递给流程中下一环节的用户处理，即异步的结构化协作。但在一些较大型的工作流系统中，更有效地完成某些环节的任务需多用户实时协作。实现工作流管理系统的协同同步可以从两方面入手：一方面，通过扩展目前 WFMS 系统中的图形化过程定义工具，实现多用户同步协作以完成工作流的形式化定义。另一方面，流程各环节任务的工作项不再与某些特定的用户相对应，而是给出一个共享工作项列表。拥有合法身份与权限的用户都同时都能看到这些工作项，且申请打开并处理工作项。同时，采用协作同步的工作方式，在过程定义、维护及分配工作项以及客户应用方面遇到一些新的问题，例如工作项被处理时相关的数据及其状态应保持一致，客户程序应能支持协同同步等等。

3. 支持移动用户的工作流管理系统[28]

随着移动计算设备（如笔记本、平板机及手持设备）的普及，企业员工能随时随地处理业务，移动办公已经成为一种新的工作方式。移动计算环境下，用户与服务器建立起连接后，首先将完成工作所需要的相关资料下载到本地，然后与服务器断开连接。而在本地工作一定时间后，再与服务器重新建立连接，并将处理结果传送给服务器。这种工作方式显然更突显用户的自主性与灵活性，WFMS 需增加工作流执行服务与客户应用间的相关协议来满足这一新特性。保证在用户处理工作项时，不会出现数据的不一致、读脏数据等问题。文献[28]提出了一个包括连接断开前、连接断开中及重建连接的三阶段协议。使用户能将工作项以及与之相关的各种资源下载到本地，而在断开与 WFMS 的连接后完成工作项的处理，再重新取得同系统连接并将处理完的工作项及处理结果保存到 WFMS 中去。

4. 事务型的工作流管理系统[29]

事务型工作流的研究主要是试图在传统的 WFMS 过程执行中，融入 Saga、Flexible Transaction 等一些高级事务模型以保证流程活动执行的完整性。但目前，工作流管理系统对事务处理中涉及到的很多相关概念（如事务的四大基本属性：ACID、事务失败语义等）仍未给出明确定义。工作流管理系统如何将事务完整性机制应用到过程实例的活动集中，是一个重要的研究问题。活动间存在一定的数据依赖关系，一个活动的执行失败时，在撤销该活动的同时还需要考虑：与其存在数据依赖的其他活动是否受到影响，如何恢复这些活动之前的状态，撤销之后是否需要给出新的参数，以及活动重新执行需要满足那些条件约束等。基于对工作流活动间的依赖关系，文献[29]中明确定义出五种如何补偿活动集的策略。这些策略存在一个共同点：都在补偿发生时系统完成对依赖关系的计算。为保证流程活动的可靠、同步与协调性，对活动间依赖关系的管理和维护也被作为一种服务来实现。

2.3 柔性工作流系统

现代企业处于一个复杂多变的环境中，业务过程重组经常发生，因此需要工作流系统能够对变化做出快速准确的反应，并且保持工作流处于相对正确的状态。柔性成为了衡量系统性能的重要指标，柔性工作流能够更加灵活地适应存在各种不确定因素的企业业务过程。

2.3.1 柔性工作流基本概念

所谓的柔性工作流即可以灵活地适应系统中各种不确定情况的工作流系统。“柔性”（flexibility）的本意是指物体在受到外力的影响时（翻转、扭转、弯曲）不会被轻易改变，或者可以通过调整和修改恢复原有形态，即物体在受到外物作用时应该具备抵制影响的能力。而在工作流管理系统中“柔性”是指：面对流程中可预计和不可预计的不确定情况时，原有系统无需被完整替换，而能够针对受影响的部分做出适当调整，保持其余部分保持不被改变，即可维持系统处于稳定正确状态的一种能力。流程的柔性实际上是如何在变化性与保持流程可被识别的稳定性之间找到一个平衡。因此，在受到内外部各种影响因素作用时，柔性工作流系统在能够保证对用户完成实际业务工作支持的同时，需要尽可能提高灵活性。系统需要做出适当调整，首先需要在变化中找出一部分不变的内容，并尽力保持不可变性，作为系统进行自动适应性调整的“支点”，以便用户更好地应对现实工作中可能遇到的各种变化、异常和不确定的问题。工作流系统的柔性主要体现在流程模型生成不同结构的流程实例、流程实例执行路径的柔性选择以及流程实例活动执行者的柔性分配等具体方面[30,31]。

与柔性工作流密切相关的概念主要有灵活性、适应性和动态性。灵活性是在工作流过程定义部分不能或者不需要确定下来的一些内容，可以在工作流系统执行时根据具体实例情况进行完善的能力。适应性主要强调工作流在执行过程中遇到的各种内外部因素变化和异常事件，能够对流程实例和流程模型的结构进行动态调整，及面对异常及时做

出反馈与应对的能力。工作流系统在运到异常或变更时，可以保持原有工作流实例不出错的继续依据其过程定义运行，并采取适当的策略向相应新定义演进，则称该系统具有动态性。

2.3.2 柔性工作流的研究现状

目前，专家学者已从不同角度和方向展开了对柔性工作流理论方法的研究，主要有以下几种总结：

周建涛等人认为研究柔性工作流首先需要找到引起系统不灵活的原因，因此，他们从流程系统中各种可预知及不可预知的变更和异常等出发展开了对柔性工作流的研究工作，并得出了相关的理论方法[3]。

Han 等人通过对工作流元模型的研究，及流程网络图中寻求开放点（OPEN-POINT）的方法，以及提出综合使用这两种方法来解决工作流管理系统在灵活适应性方面存在的（adaptive workflow）问题[32]。

Heinl 等人提出了从两个方面入手来提高工作流系统的柔性：一是流程过程灵活的选择执行路径；二是用户可根据实际情况对流程进行动态调整，并提出了在动态调整中可采用的一些处理策略[33]。

Schonenberg 等人提出了四种方法用于支持工作流系统的柔性，并初步给出了各种方法的工作原理与实现方式，并以此为参照，罗列出了用以对当前柔性工作流系统进行柔性指标评价的影响因子[34]。

Nurcan 等人则侧重于对工作流柔性本身应具有的基本性质，以及各类柔性工作流系统的实现方式上的研究，在其研究理论方法中进行了详尽讨论与描述[35]。

Weber 等人从工作流典型的生命周期出发，流程一般可分为定义、建模、运行等阶段，并讨论了在这些不同阶段，实现柔性工作流应采用不同的实现方式[36]。

通过对上述提高工作流系统柔性的理论方法的分析与总结，主要可分为以下四类：动态选择、动态细化、动态偏离以及动态变更。基本包括了如何利用柔性工作流技术来实现对各种业务工作更好的支持[3,31,37]。

表2-1 柔性工作流理论方法对比

实现方式	能力评价方法	优点	缺点	适用领域
动态选择	工作流模式	模型保持不变，流程用户只需关注如何完成所分配的活动并根据具体情况选择模型提供的执行路径。	建模人员能力要求较高，不可能完全描述所有情况，模型会变得十分庞大难于理解和维护；而且也不现实，不支持流程结构的动态调整。	适用于所有领域， 尤其是有标准规程可参考，工作内容结构化程度较高 异常情况较少发生的应用领域，如生产制造。
动态细化	变更模式和变更支持	可有效应对不确定性问题，降低了设计时对建模人员的要求。	受到流程模型约束 需要事先确定可被“细化”的部分，不支持流程结构的动态调整。	适用于存在不确定性，需要参与人员发挥创造性解决问题应用领域，如产品研发。

续表 2-1

实现方式	能力评价方法	优点	缺点	适用领域
动态偏离	工作流模式	可提高流程用户对业务工作进展的掌控能力，并在一定程度内提高用户处理异常的能力。	受流程模型约束， 若对实例运行状态调整不加约束则会引起运行状态混乱，若加约束则提高对建模人员能力的要求；不支持模型结构动态调整。	适用于常有未知异常发生，但又无需修改整个流程的情形，尤其是围绕着文档、数据等信息制品工作的应用领域，如保险理赔处理。
动态变更	变更模式和变更支持	可动态调整流程结构，能有效应对各种异常情况和内外部因素变化带来的流程调整。	需要解决动态变更错误的问题，对变更操作人员的能力要求较高。	适用于所有领域， 尤其是易受技术进步和政策变化等内外部因素影响而调整流程，且流程的运行实例较多的应用领域，如医疗。

2.4 小结

本章给出了几种有代表性的工作流定义，对工作流参考模型以及工作流管理系统进行了概括和描述。研究了柔性工作流系统的产生、发展，并对柔性工作流系统的基本概念、研究现状等进行了分析。

第三章 RBAC 技术的分析

3.1 访问控制技术概述

在计算机通信中，网络安全的防范与保护极其重要，访问控制（Access Control，AC）是一项主要的安全保护策略。访问控制的基本策略是：基于身份认证原理，系统中被授权的合法用户依据权限约束可以对系统中可获取的客体资源提出访问请求并进行相应控制操作。通过这一机制，实现合法主体对其授权客体的合法操作，同时防止合法主体任意访问权限之外的资源或者非法主体入侵系统造成破坏。访问控制模型中的三个主要元素：主体、客体、主体-客体的访问控制约束[38]。

1. 主体（Subject）：主体通常指人，也可是能够主动发起操作请求的智能设备、应用程序、进程或者服务等。主体通过拥有权限实现对相关客体资源的访问控制。

2. 客体（Object）：计算机信息系统中可以被相应主体进行访问控制管理的各种信息和资源都可以称为客体，例如计算机处理器、打印机、存储在计算机上的各种文件资料、网页、数据库中的数据表、视图等。

3. 主体-客体访问控制约束（Access）：主体在对客体进行访问控制时，需要遵循一套相应的规约准则，以保证访问权限的合理性。访问控制的方式多种多样，通常是主体对客体资源的某种操作，例如进入计算机系统、登录应用系统、文件的读写、数据库的修改操作、获取网页等。

访问控制技术能较好的在各种计算机系统中实现与应用，首先由于其用于表达访问控制约束的概念性框架即访问控制模型具有较高可靠性与灵活性，访问控制模型的规范性决定了访问策略的严谨可靠[39]。主体与客体之间通过权限的授予来明确其访问控制的合法合理性，访问控制约束就是判断主体提出对客体的操作请求是否属于操作许可集，若请求为许可集子集则验证通过，操作合法，否则，请求得不到允许，拒绝对客体的非法访问操作。访问控制模型如下：

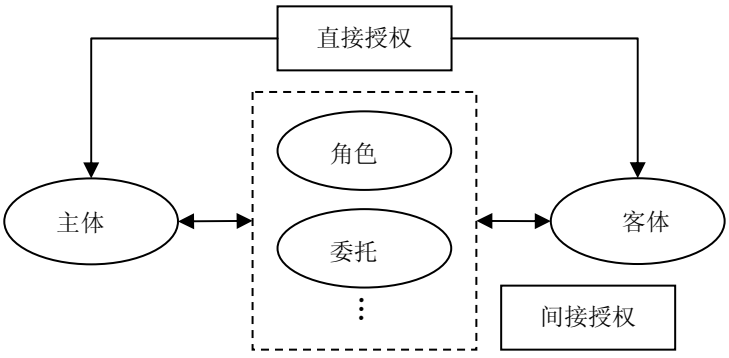

图 3.1 访问控制模型

访问控制模型两个最基本的实体是主体（S）和客体（O），主客体间通过权限（P）实现相互关联。授权方式有两大类，主体（S）可以直接与权限（P）进行关联，从而直接被授权（SP），也可通过中间机制角色或者委托间接与权限联系而被授权。

3.2 典型的访问控制策略

访问控制是保障信息安全的核心技术之一，以保证资源的合法使用，实现数据的安全性与完整性。访问控制技术发展迅速，在灵活性与安全性之间需要找到一个平衡点。近年来，在企业中应用最广泛的访问控制方法主要有三种：自主型访问控制（Discretionary Access Control，DAC）、强制型访问控制（Mandatory Access Control，MAC）、基于角色的访问控制（Role-based Access Control，RBAC）。

3.2.1 自主型访问控制策略

自主型访问控制策略（DAC）：是一种以主体为主导的访问控制机制，拥有客体资源的实体即主体依据客体的特征以及自身的需求，可以完全自主的决定自身对客体的访问操作权限，而在其他主体提出对客体资源访问控制请求时，该主体能够自主进行授权，同时，主体还能撤销和收回这些权限。

自主型访问控制策略简单实用、灵活性好、易于扩展，在商业领域中应用最为广泛，但同时也明显存在安全方面的问题。一方面，主体对权限分配的完全自主可能造成权限的滥用，产生不可预计的安全隐患。另一方面，客体拥有者将权限分配给主体后，权限在主体间的传递性，客体的访问控制权限将过于分散，从而使得自主访问控制应用系统无法保证客体的访问权限许可[40,41,42]。

3.2.2 强制型访问控制策略

强制型访问控制策略（MAC）：它是与自住型访问控制截然不同的一种访问机制，主体和客体通过安全级别的标识来确定能够进行访问控制。MAC 针对主体和客体本身的特性，划分安全级别，主体的安全级别代表着主体的访问控制能力，客体的级别则反映了客体资源的敏感程度。访问请求提出后，MAC 机制通过比对请求提出方（主体）与请求接受方（客体）两者的安全级别系数，来决定请求是否通过验证，操作能否进行。

强制访问控制技术在安全方面得到了保障，主体和客体的安全级别由系统统一定义，集中管理，且任意主体不能自由修改、传递及委托，从而避免了不合法的访问。但强制访问控制不够灵活，安全级别一经指定修改不便，控制方式太过严苛死板。因此，这类技术通常被用于对安全性能要求较高且各个级别限定较稳定无需经常改动的军事系统中[40,41,42]。

3.2.3 基于角色的访问控制策略

基于角色的访问控制策略（RBAC）：针对 DAC 与 MAC 两者存在的缺陷，RBAC 的提出综合了优点，成为目前大型企业中应用最好的访问控制机制。RBAC 在主体与客体之间引入角色这一中间机制，主体与角色相对应，而对客体的操作权限则与角色相关联，主客体不再直接关联，从而大大降低了系统中两大主要实体的耦合性，提高了系统访问控制机制的安全性、灵活性和适应性。

角色是一个抽象的概念，RBAC 中角色是一个非常重要的中间机制。对客体的访问控制权限分配给相应的角色，而主体需要操作客体资源则首先需要通过扮演对应角色，从而拥有角色的全部或者部分权限。在大型企业业务中，主体可能需要跨角色即同时拥有多重角色身份才能完成实现对某客体的访问控制[40,41,42]。

RBAC 技术的两个主要特征：1、大大降低了授权管理的复杂性，客体操作权限的分配与收回主体完全不受影响，同时主体的变动也不会引起访问控制约束的改变，管理员只需将角色-主体分配关系作相应变动即可。2、依据企业组织结构的不同岗位设置相应角色，能够很确切的反应现实组织结构复杂的关系，因此对于实际工作中权限的分配拥有关系的安全管理策略也易于理解和管理。

3.3 RBAC 模型

3.3.1 柔性工作流基本概念

美国国家标准化技术委员会（NIST）Ferraiolo 等人在 20 世纪 90 年代提出了基于角色的访问控制技术（Role-based Access Control，RBAC），其基本思想是在主体和客体之间引入了角色这一中间体进行主体和角色关联、角色和客体关联，从而实现主体对客体的控制管理。

RBAC 参考模型经历四个发展阶段：RBAC0 模型、RBAC1 模型、RBAC2 模型和 RBAC3 模型。RBAC0 作为基本模型，是应用 RBAC 系统的最小要求。角色继承模型 RBAC1 和约束限制模型 RBAC2 都包含 RBAC0，但 RBAC1 增加了角色的层次概念，RBAC2 在角色分配、权限分配及会话建立等方面均加入相关约束条件。RBAC3 是对 RBAC1 和 RBAC2 的综合，是一个相对完整的模型[43]。

RBAC0：是最基础的 RBAC 模型，包含了用户、角色、权限、会话等访问控制的基本元素以及各元素之间的关系。角色与权限相对应，用户拥有角色来获取对客体的操作权限，通过建立会话来实现对合法角色的激活。

RBAC1：满足 RBAC0 的要求，RBAC1 提出了角色的继承机制。该模型中角色与角色之间存在着层次结构，能够更好的与实际职能岗位相符合，用户如果被赋予上一层角色则相应的也拥有了该角色所有下级角色对应的访问控制权限，即角色的继承关系。

RBAC2：RBAC2 在基础模型之上，增加了一些相关的约束条件，对用户角色分配、角色权限分配、用户建立会话等各过程给出一系列的限制条件。增强了访问控制实现方式的可靠性与安全性，使 RBAC 模型更加完善。

RBAC3：这一模型是 RBAC1 和 RBAC2 的综合体，即在基础模型上既包含了角色继承机制又实现了各种实体关系间的限制，是最终完整的体系结构。

3.3.2 RBAC96 模型简介

1996 年，美国 George Masno 大学 R.Sandhu 等人对 RBAC 基本模型进行深入研究，增加了角色继承机制和权限约束限制机制，提出了一个完整的 RBAC 模型，称为

RBAC96 模型[43]。RBAC96 模型如图 3.2 所示。

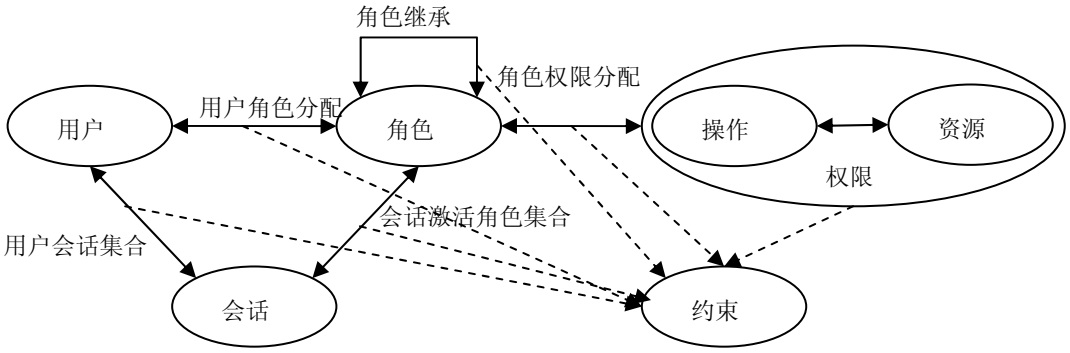

图 3.2 完整 RBAC（RBAC96）模型

1. 用户：用户是访问及操作计算机系统中数据以及文件、WEB 页面等以数据形式表示的其他各类客体资源的主体。主要是指组织中的人员，也可是智能设备或者一些软代理等[43]。

2. 角色：角色是一个抽象的概念，是用户与权限中间一个连接的桥梁。角色一方面可以表示各类用户的职能，另一方面也需要反应出对客体资源的访问控制能力，即用户完成某种职责必需先获取一定的权限。角色的划分通常以企业组织结构为依据，考虑组织中各岗位的实际职责，从而分配适当的权限[43]。

3. 权限：权限由两个部分组成：资源和操作。资源（Resource）是访问控制系统中的操作接受方，通常被称为客体；操作（Operation）是一个实际的动作行为，由主体发出实施在系统资源上，如登录、读写、更新等。权限就是一个许可集，表示系统中的各类客体资源能够被执行的各种操作[43]。

4. 会话：会话是连接角色与用户的一个纽带。用户与角色之间存在一对多或者多对多的关系，在一个系统中用户可能拥有多重身份，但是这些角色最初都处于钝化的状态，用户在某一具体时刻通过建立会话激活某一角色，从而将角色相应的权限变为可用状态。会话由用户发出，角色是会话的接受方，用户一次可以建立与一个角色的会话也可同时建立多个会话，但一个会话只能是来源于一个用户的请求与一个角色的接受[43,44]。

5. 约束：为避免冲突，提高模型稳定性，RBAC2 在基础模型上添加的一系列限制条件。模型中各实体及实体间的关系都需要给出一定限制，在用户-角色-权限三者的分配时需要设置约束条件，如职责要分离、角色互斥限制、角色基数、先决条件约束、角色继承层次约束等都是 RBAC 机制中一些典型的约束[43,44]。

6. 角色与用户的关系：在 RBAC 机制中，用户与权限被角色屏蔽开，不直接关联，用户要实现对客体资源的访问操作，就需先被赋予一定者角色集，从而获取角色所携带的各种权限。角色与用户的关系，是一对一或多对多的分配关系，即一类角色可分配给多个用户，同时一个用户也可被赋予多重角色身份[43,44]。

7. 角色与权限的关系：权限表示客体资源能被执行操作的许可集，RBAC 以岗位为依据划分角色，依据角色的职能给角色分配权限，实现合理合法授权。角色与权限的授

予关系，参考实际情况可以一类角色拥有多种权限，同时一种权限也能被赋予给多类角色[43,44]。

8. 角色与角色的关系：同组织结构中的上下级别相一致，角色之间也存在着层级关系。明确了角色的继承关系对用户角色分配以及角色权限分配上都有重要的意义。如角色 ra 继承于角色 rb，用户拥有了角色 ra 后，即可获得 ra 相应的权限也可获得 rb 所拥有的权限[43,44]。

3.3.3 RBAC 模型存在的问题

1. RBAC 模型中明确提出了“角色”这一概念，也对角色的继承与互斥关系、角色分配中基数限制等给出定义，但在流程运行过程中，运行条件的改变会引起角色特性的变化，所以需要有动态特征来描述角色[45,46]。所以需引入一些动态的属性对角色进行描述，如角色的实例标识、会话标识、激活时间、生存期等。

2. 传统的 RBAC 模型中，与权限有直接关联的实体是角色，从而决定了在对权限的管理至多能细化到“角色”这一粒度上。而这与现实世界中的角色扮演情况往往是不符合的，一类角色可能拥有执行两项或以上任务的权力，只要建立了与某一角色的会话，该角色所对应的权限集都分配给了用户。在实际工作中，在一个正常的工作时间点上，一个用户通常只能扮演一种角色担当一项任务的执行人[45,46]。因此，以上两点存在着明显的矛盾性，角色与用户的映射关系使得用户在执行某一任务时拥有多于完成该项任务所需要的最小权限，这时发现在角色级别定义的最小权限约束只成为假象，并不能真正达到最小权限约束效果。

3. RBAC 没有给出如何细化流程任务，不便于权限最小分配原则的应用。一个任务可能有许多基本动作构成，直接对任务授权的粒度也不是最小的。一项工作流任务可能有多个工作步组成，若两工作步有先后关系，用户在未执行前一工作步时不能执行后面的工作步。所以有必要引入工作步的概念，在引入了工作步的概念后，工作流不再直接拥有权限，而是各个工作步对应的基本单元的权限的和，工作步只拥有它需要的最小权限，所以基本单元具有原子性[45,46]。

3.4 小结

本章对访问控制技术进行了介绍和研究，详细分析了 RBAC 模型结构及四个发展阶段，提出其存在的问题并给出相应的解决办法。

第四章 基于 RBAC 的柔性工作流模型研究

4.1 基于 RBAC 的柔性工作流模型

工作流系统一般包括流程定义、流程建模、流程执行三个主要步骤，其中，流程定义决定了模型的构成从而最终决定流程实例的运行情况。然而现代企业业务需求的多变性使得一次给出完整流程定义的方式显得不切实际，在遇到一些定义时未预计到的异常情况，或者出现与预计结果不相符合的状况时，原有的流程系统就无法继续顺利运行下去[47,48]。因此，需要一个更稳健的工作流系统，它能够很好地处理各种不确定因素，灵活应对异常状况对系统造成的冲击，提高工作流应对不确定因素的能力，即提高系统的柔性。对于如何提高系统的柔性，目前，工作流领域的专家学者以及相关企业已从多角度入手展开研究，也取得了一定的成果。本文主要从两个方面着手，分析并实现系统柔性的提高，一是系统流程过程的柔性，二是组织中用户权限管理的柔性。同时，在对工作流模型、RBAC 技术研究的基础上，提出了一个更灵活、更安全的柔性工作流模型。

4.1.1 模型的提出

通过对工作流管理系统的研究可知，WFMC 提出的工作流模型作为当前工作流领域较为权威的代表，主要包括六大组成部分及与之对应的五个接口，用以实现多构件之间的连接。但该基本模型仅对模型元素以及各元素存在的关系进行了说明和定义，随着业务流程的越来越复杂，该传统模型在实用性和适应性方面的不足越发凸显。建模阶段与运行阶段几乎完全隔离开，并未对流程实例运行时可能遇到的异常情况做出预计，以至于在实际业务活动中模型的稳健性相当不够，突发情况的发生可能会导致需要对整个模型进行修改，因此模型的柔性问题亟待改进。

另外，工作流执行服务器与工作流客户应用的关系主要表现为任务与任务执行者之间的分配问题，二者仅通过一个接口连接，这样的模型设计出的流程实例在运行过程中存在着不够灵活之处。而基于 RBAC 的柔性工作流模型，引入 RBAC 技术，参考其在用户权限管理方面的先进思想，并结合工作流系统的实际组织结构，通过构建一个独立的用户权限管理模型，专门用于管理业务流程中的用户-角色-权限的关系。在工作流执行服务与客户应用程序两部分之间加入此模型，改变了二者的直接关联，并保持二者之间的逻辑调用关系，而作为主体的用户对任务的客体资源的操作这一具体实现过程对二者相对透明。这个过程分为两个步骤：首先是工作流执行服务器在产生任务时与用户权限管理模型建立连接，二者实现从任务到权限的映射，明确任务执行时需要作用的客体资源与进行的实际操作；然后是在用户权限管理模型与工作流客户服务程序之间，即实现任务的最终执行人与模型定义中的角色以及用户的对应，从而实现工作流任务与任务执行者合理、灵活的分配。

通过第二章对工作流参考模型、体系结构以及柔性工作流技术的研究，第三章对完整 RBAC 模型的分析，确定了以 WFMC 的工作流模型为基础，构建出基于 RBAC 的柔

性工作流模型，如图 4.1 所示。

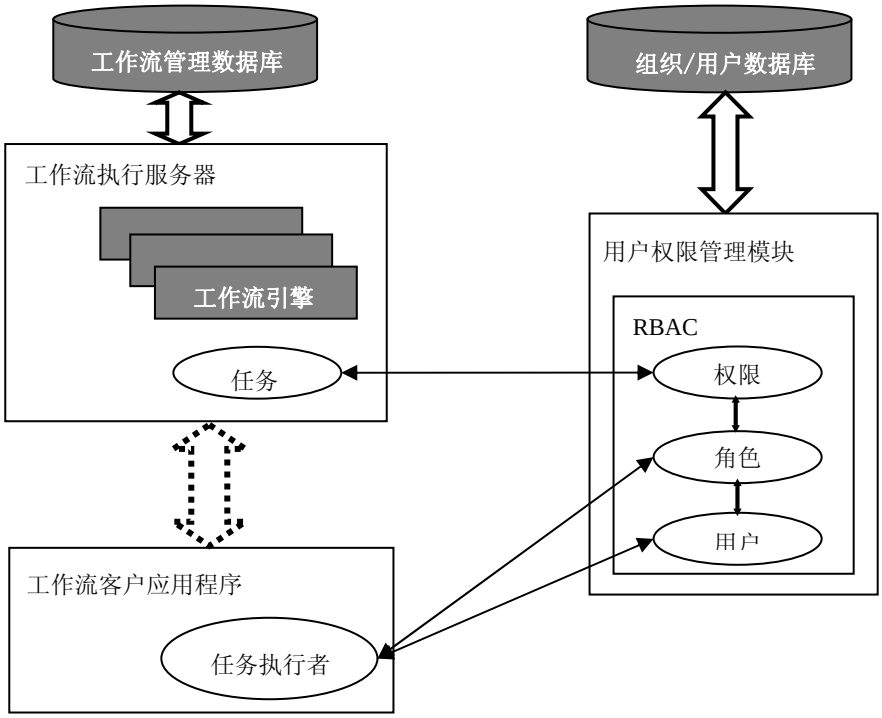

图 4.1 基于 RBAC 的柔性工作流模型架构

构建图 4.1 所示工作流引擎的同时，还要考虑到流程在执行过程中的柔性，在具体的实现过程中，将工作流系统中所涉及的对象进行有效划分，建立起类似数据库映射机制的基空间和元空间，其中，基空间为所构建的工作流系统，而与之对应的元空间是由抽象出来的各种元对象以及对其进行管理的元对象协议（Meta Object Protocol，MOP）构成，通过协议对工作流进行的动态调整，由系统把对元对象的各种处理映射到基空间中的具体过程实例上去，从而支持流程实例的灵活变更。

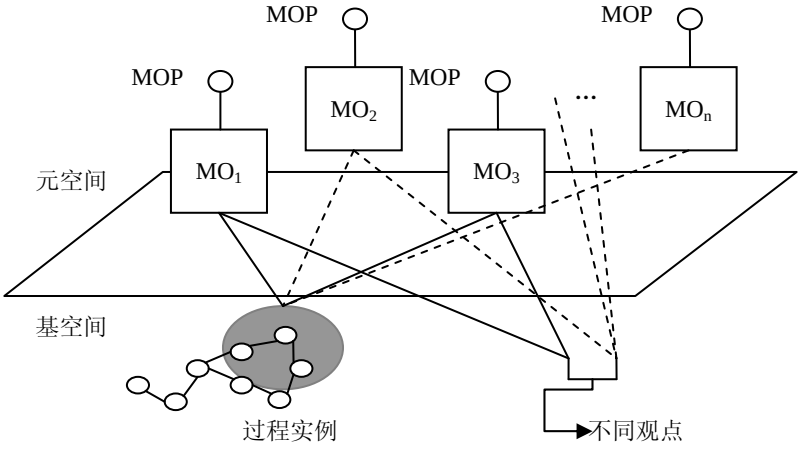

图 4.2 空间映射示意图

4.1.2 模型元素的形式化定义

基于上述设计，给出模型元素形式化定义如下：组织中的用户（USERS 简记为字母 U）、与组织结构相一致划分的各类角色（ROLES 简记为字母 R）、流程任务（TASKS

简记为字母 T）与用户权限管理模型权限（PERMISSIONS 简记为字母 P）的映射、工作流客户应用程序任务执行者 Executor（E）与用户权限管理模型中角色 R、用户 U 的映射。

1. 角色集 $R = \{R1, R2, ...Rn\}$，角色是一个引入的概念，是工作流模型中一个重要的组成元素。以组织结构为依据划分相应的角色类型，角色间也存在着复杂的关系，如上下层级关系、平级交互关系等。如与实际职能岗位相对应的角色 Ra、Rb，两者之间上下级别的关系可能体现为 Ra>=Rb。

2. 用户集 $U = \{U1, U2, ...Un\}$，用户作为任务的最终执行者是工作流模型的基本元素之一。组织中所有人员都可以成为用户集中的原子，用户与角色相比较具有很强的不稳定性，人员的更新换代都会引起用户集的变动。

3. 工作流机定义流程任务 $T = \{T1, T2., ...Tn\}$，实现任务集合中的每一个元素 T 与用户权限管理模型中 $P = \{P1, P2, ...Pn \mid Pi = \{(REi, OPi) \mid i = 1,2, ...n\}\}$ 一一映射。完成一项任务通常是指需要对哪些资源给予哪些操作，操作成功与否代表任务是否完成。还需要涉及到一些约束条件如：任务的执行人约束、任务的完成时限、任务的优先级别等。因此，任务与权限的映射关系又可进一步细分为，T 与资源集 $RE = \{RE1, RE2, ...REn\}$、操作集 $OP = \{OP1, OP2, ...OPn\}$ 的关系。

4. 用户权限管理模型实现权限（P）—角色（R）的分配：$PR \in P \times R$，角色 R 拥有的权限 P，$ROLES(P) = \{p \in P \cap (p, r) \in PR\}$。PR 代表角色对权限的拥有关系集合，而对集合中的任意 PRi 都有：$PRi : \{mP \leftrightarrow nR, m, n = \{0,1,2.....,n\}\}$。即角色与权限的映射可以是一对一也可以是多对多的关系。

5. 用户权限管理模型实现角色（R）—用户（U）的分配：$RU \in R \times U$，用户 U 拥有的角色 R，$USERS(R) = \{r \in R \cap (r, u) \in RU\}$。由于角色存在层次关系，若角色 Ra（拥有权限集为 Pa）为角色 Rb（拥有权限集为 Pb）的下层角色，用户 Ua 被赋予角色 Ra，此时 Ua 既拥有了角色 Ra 的权限也同时拥有了角色 Rb 的权限。角色间存在的平级交互关系，如在完成一项复杂的或者是安全性较高的任务时，用户 Ua 可能同时需要被赋予了角色 Ra（拥有权限集为 Pa）和角色 Rb（拥有权限集为 Pb）才能成为合法用户，任务执行时 Ua 与 Ra 建立会话需同时与 Rb 建立会话。

6. 任务执行者 E 可以是某一个确定的用户，而针对流程定义时不能确定到具体用户的情况，则将任务执行者与某一类角色进行映射，即拥有了可获取该类角色的任何人。

即：$E = \{(u \cup r) \mid u \in U, r \in R\}$。

7. 结构元对象 Stru_MetaObject，对工作流的流程方面进行描述和管理，通过其中构造的 get 方法，取得工作流的拓扑结构。

8. 行为元对象 Fun_MetaObject，对工作流活动的执行过程中所用到的信息进行描述，同时具有获取信息、重置信息的能力。

9. 状态元对象 Sta_MetaObject，对工作流的状态进行获取，同时具有挂起、激活、终止的能力。

通过这样的设计，工作流系统每次启动一个流程实例的同时，初始化与其对应的一系列元对象。这些元对象对应于流程实例的状态、功能、运行行为等，并随着实例的运行而随之改变，能实时反应流程实例的各方面情况[48]。工作流系统以元对象协议为依据，将对元对象的各种操作映射到与之对应的基空间流程实例上。因此，能通过此规范化协议的方式来实现对工作流系统的各种调整的控制，既能实现对原有流程的干预，又能较好的控制在允许的范围之内对系统进行修改，避免发生给系统带来破坏性的操作，保证了系统在流程运行过程阶段具备较大的自由性。

由上述定义可以得出，基于 RBAC 的柔性工作流模型包括的主要元素有组织模型、角色、用户、权限以及过程模型、任务等，而明确这些元素在流程运转以及权限分配方面存在的各种复杂关系，是实现流程柔性的首要工作。通过研究，确定了彼此之间对应关系如图 4.3 所示。

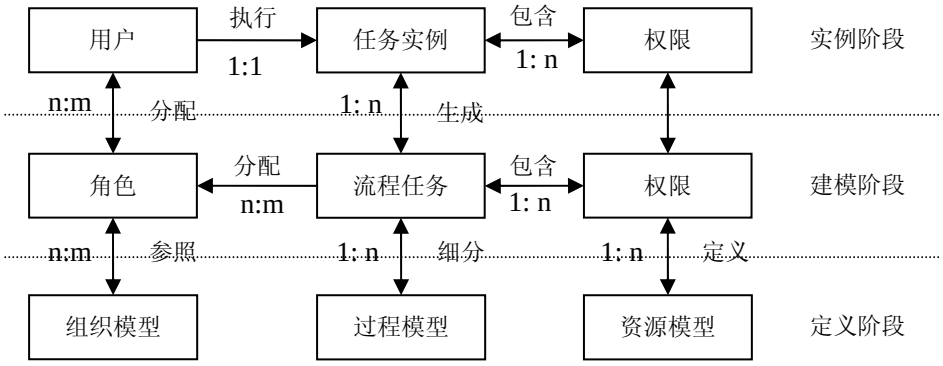

图 4.3 模型元素关系图

4.2 基于 RBAC 的柔性工作流模型

通过对上述模型的分析与元素的定义，本小节将从两个方面来详细剖析如何实现工作流模型的柔性，即流程过程的柔性和用户权限管理的柔性。

4.2.1 流程过程的柔性

工作流流程过程主要由节点与路径两个基本部分组成[49]，节点表示工作任务需要执行的各个环节，而路径则表示由各个具有先后流转次序的节点构成的一条执行路线。本文主要从这两个方面来讨论流程过程的柔性。

1. 流程节点的柔性

节点（Nodes 简记为 N）的柔性主要是指在实际业务过程中通常会发生节点的新增与删除等情况，在流程建模阶段无法完成准确的预计，即是否设置了一些不必要的多余环节，或者是否还存在一些本身未被包含进来的关键环节。因此，除了充分考虑业务实际情况，尽可能地提高静态定义时节点设置的准确性之外，还应该考虑在流程过程中满足动态修改的要求。

我们可以通过图示进一步明确流程节点的柔性，如图 4.4 所示，在某流程的节点 Na 与 Nb 之间新增加节点 Nc，则需要相应修改节点集的原偏序关系<Na，Nb>为新的偏序关系<Na，Nc>和<Nc，Nb>。而在删除一个流程节点时，如图 4.5 所示，在流程中需要删除节点 Nc，保留其他节点 Na、Nb 不变，删除节点 Nc,相应的节点集合偏序关系由<Na，Nc>与<Nc，Nb>变为<Na，Nb>。

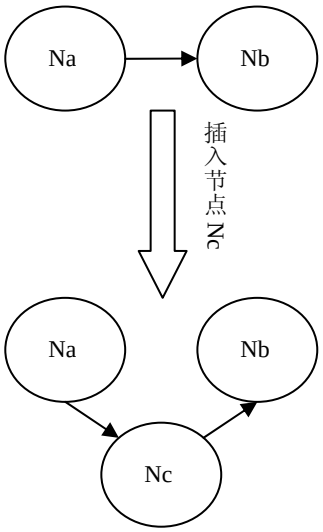

图 4. 4 新增流程节点

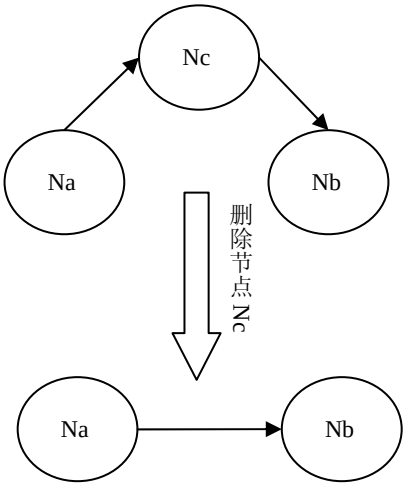

图 4.5 删除流程节点

2. 流程路径的柔性

在流程中与节点相辅相成的另一元素是路径（Paths 简记为 P），路径表现为节点间的流转关系，相对于节点而言，路径具有更加复杂多样的动态性。在流程定义时预先设定了流转路径，但在流程实际执行过程中，任意条件、参数以及客观需求等的变动都可能会引起流程最终走向的改变。因此，本文从三个角度来分析流程路径的柔性即：新增流程路径、撤销流程路径以及改变原有流程路径的方向。

新增流程路径：如图 4.6，可以看出在节点 Na、Nb、Nc 之间存在的原路径为从 Na 到 Nb 再到 Nc，而新增路径后，路径集的偏序关系从<P（Na，Nb，Nc）>变为<P（Na，Nb，Nc），P（Na，Nc）>。同样当发生需要撤销路径的情况，从图 4.7 可知，撤销了从 Na 到 Nc 路径后，对应的路径偏序集也从<P（Na，Nb，Nc），P（Na，Nc）>变为<P（Na，Nb，Nc）>。而对于改变原路径方向来说，可以有两种情况，一是直接将原两节点的流转方向反转，二是保持原有流程走向路径的同时，在两节点间再增加一条新的反方向的路径。反向路径后，偏序集关系由原来的<P（Na，Nb）>变成<P（Nb，Na）>；而增添

另一条路径后，偏序集关系由原来的<P（Na，Nb）>变成<P（Na，Nb），P（Nb，Na）>，如图 4.8 和图 4.9 所示。

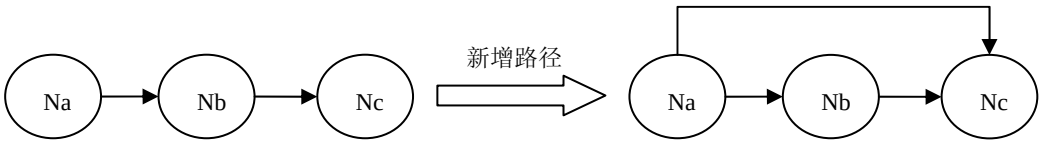

图 4.6 新增流程路径

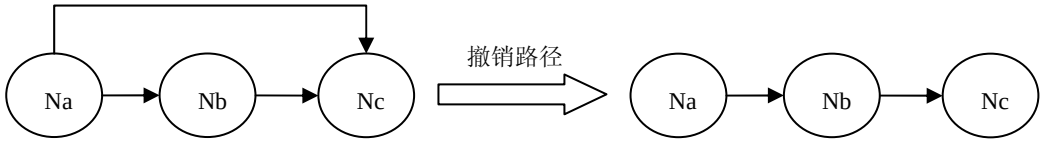

图 4.7 撤销流程路径

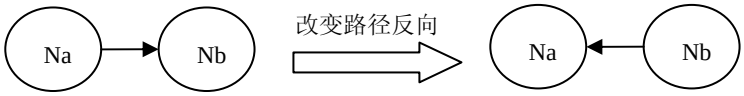

图 4.8 改变流程路径（反向）

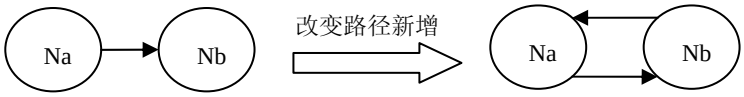

图 4.9 改变流程路径（新增）

通过以上的分析，同时借助之前所定义的元对象，可以进行流程的柔性控制，如：在进行节点增加时，可以对流程实例 WorkflowInstance（简记为 f）进行如下调整：

1) 首先进行状态检查，保证该调整只有对活动的节点才有意义。

Sta_metaobject.getWfactivityInsstate(Na);

Sta_metaobject.getWfactivityInsstate(Nb);

经过检查得知活动 Na 和活动 Nb 处于未开始状态，因此可对它们进行调整。

2) 使工作流实例暂停运行。

Sta_metaobject.SuspendWfinstance(f);

3) 在活动 Na 和 Nb 之间插入分支节点 Nc。

Stru_metaobject.insertNewnode(Nc);

在插入的过程中，在 Na，Nc 和 Nc，Nb 之间自动加入路径。

4) 激活工作流实例 f。

Sta_metaobject.activateWfinstance(f);

工作流实例 f 重新启动后，将同时激活活动 Na 和活动 Nb，从而完成调整过程。当然，在原有流程实例发生改变后，与之相关的节点信息需要重新载入，这些在编程实现时一并予以设计实现。

4.2.2 用户权限管理的柔性

工作流作为对企业业务流程自动化的技术支撑，必然会受到企业组织内部变化的影

响。从组织管理灵活性的要求出发，工作流系统需要提供一个专门的管理模型。总结企业组织的变化得知，其柔性主要表现为两个方面：一是组织机构本身的改革（如岗位部门的改变、企业管理模式的更改）；另一是组织内部员工的更新换代（如员工的新进、离职及晋升等）[50]。

对于一个企业组织结构的革新是极其常见的情况，企业在市场环境下面对日益激烈的竞争压力，必需要通过不断修正组织结构来更好地适应外部变化。新增岗位、撤销岗位或者是更改原有岗位部门从属关系等政策的实施，都需要及时对工作流中相应的元素集合做出调整[51]。在基于 RBAC 的工作流模型中，角色的划分是与组织结构相对应的，因此，角色的灵活性非常大，能否及时做出角色的新增、角色的删除及对原有角色做出适当的修改等都是企业工作流系统柔性程度的体现。

同时，企业组织结构的变化会造成对员工的调整，员工的新增、离职、晋升以及退休等都会造成原有工作流系统中参与者集合、用户角色的对应关系集等发生变动。虽然引入了“角色”这个中间机制后，参与者几乎不与权限存在直接的对应关系，不会引起访问控制权限的变化，但是人员自身的一些属性如工作能力、工作状态等比较难以预估且会影响到角色用户分配时的权衡比重。因此，对人员柔性的分析具有重要意义，有利于流程用户权限更恰当的分配。

4.3 柔性工作流的实现

通过 4.2 小节的分析，本节将从上述两个方面阐述流程柔性的具体实现，即实现流程过程的柔性与用户权限管理的柔性。流程过程柔性主要是指流程节点的灵活增、删、改和流程路径走向的更换；用户权限管理柔性主要是指流程实例任务执行者灵活分配。

4.3.1 流程过程柔性的实现

流程过程柔性从两方面来体现：流程节点和流程路径。在流程定义阶段，尽可能清晰地分析业务需求，梳理出主要的业务环节及业务步骤。在建模阶段，按照流程定义，明确各流程环节的相关属性、环节上的约束条件、指派环节相应执行角色、安排环节间的次序关系，完成流程模型的建立，这两个步骤跟传统工作流系统基本相同。

柔性工作流与传统工作流的最大区别在于：传统工作流一旦定义、建模之后，流程的环节、环节执行人以及环节间的先后关系等就被确定，而流程运行阶段是按照工作流模型生成流程实例运行并完成工作。在实例运行期间发生任何意外情况，实例就可能无法继续运行，若这些不确定的因素一直存在则可能导致此工作流系统从定义到模型全部做出改变。然而，柔性工作流在应对这些不确定因素时具有明显的优势，工作流定义决定模型，但工作流模型仅仅是流程实际运行阶段的一个基本参考。正如 4.2 小节的分析，通过对节点与路径的灵活控制，柔性工作流在实例化流程时，能根据业务需求，实现同一流程定义产生不同结构的流程实例；另一方面在流程实例运行过程中，合法执行人同样可以根据实际情况调整流程的走向，灵活控制流程以适应各种不确定因素的发生，最

终完成业务目标。

4.3.2 流程执行的工作模式

1. 顺序执行模式

工作流引擎采用“同步器”节点来表示模型中的计算逻辑。同步器“S”是一个一般意义上的同步器，同步器前后与相关实体相接，开始节点、结束节点是特殊的同步器。采用顺序执行模式时，依存的业务实例如图 4.10 所示。

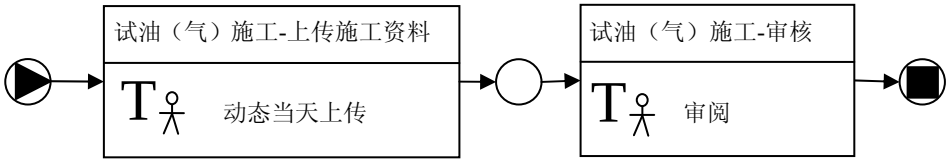

图 4.10 顺序执行流程示意图

2. 自由流模式

“自由流”是一种符合中国特色的业务流程模式[52]，即在某些特殊情况下，流程不按照既定顺序执行，而是在环节间自由跳转。在工作流引擎中，用 jumpTo 方法实现完成当前工单后跳转到指定的活动。自由跳转是有条件的，其首要条件是当前环节和目标环节在同一个“执行线”上，如果工作流引擎检查到当前环节和目标环节不在同一个执行线上，则拒绝执行跳转操作。“执行线”是自行创造的一个工作流名词，在图论中的定义是 Li（activity1）=Li（activity2）。

图 4.11 为模拟跳转模式，如其所示，试采设计完成初审，若下一步审核不需要执行，则初审人可选择直接跳转到工程部分编写环节，实现审核节点的跨越。

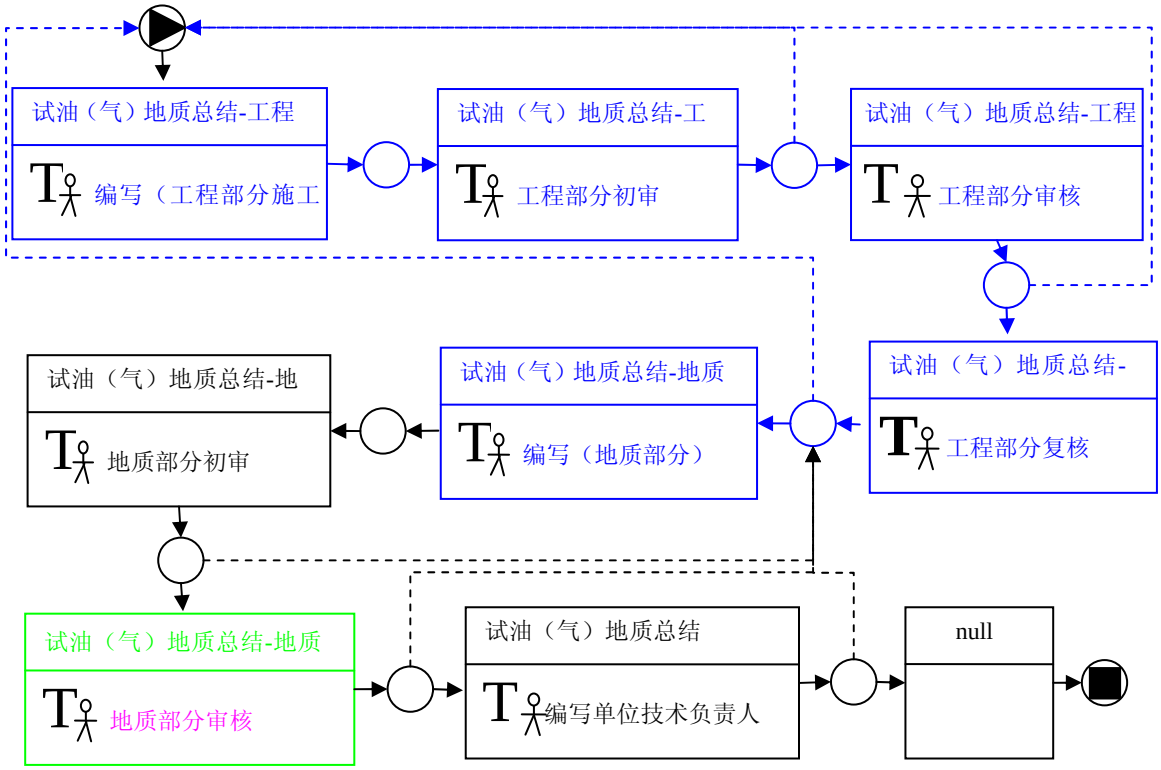

图 4.11 模拟跳转模式图

3. 回退模式

实际工作中，存在审批环节不通过的情况，此时需要将工作返回给上一环节执行人，同时在审核意见中给出评价，提出修改意见。在设计过程中，回退模式通过调用 jumpTo 方法，根据执行者的决定回退到前面任一环节，实现流程灵活运行。回退模型实现的原理与跳转类似，即回退是一种特殊的跳转，可跳转到之前的任一执行环节。同时在用户分配上也具有特殊性，即有针对性地将任务分配给返回之前签收并完成任务的所有执行人，而其他同级用户不会被分配该任务。在之后应用到的具体实例中，这种流程过程的柔性体现主要分为两个方面：一是每个流程实例启动者能根据实际业务需求来决定该流程实例的执行节点以及流程的执行路径；二是流程实例启动后，各个环节上的合法任务执行人也可以根据业务的变更或者其他情况，决定是否需要对当前流程实例做出更改以便更好的满足实际需求，完成业务目标，即增加新的流程节点、删除原有流程节点、跳转执行流程任务等操作都是被允许的。

4.3.3 推式策略与拉式策略

用户权限管理是工作流系统中一个重要的部分，组织机构与人员变动相对于系统中其他因素具有更复杂的不确定性，因此，提高系统的适应性需要非常重视用户权限管理方面的柔性[53]。

在用户权限管理的设计上，考虑到用户任务分配具有主动获取和被动分配的特点，可以将推拉式理论引入到本工作流模型设计中，并将其命名为推式策略与拉式策略。所谓推式策略是指从合法的用户集中选取一个或者多个用户，并将任务委派给他们，被安排工作的人员将在自己的工作列表中看到这些任务，并根据规定完成任务。而拉式策略则不同，是指任务产生之后不会被安排给任何一个用户，而是先将其公布在系统公共的共享工作列表中，用户主动对工作任务进行申请，系统再对该任务的所有申请用户的情况进行一次综合审核，从而最终决定将任务分配给谁。这两种用户权限分配策略各有其优缺点，本文所设计的模型中将两者混合用于各流程环节的任务分配中。

4.3.3.1 推式策略

推式策略是由工作流执行服务器将任务分配给所有合法用户，即用户登录后其相应待办工作列表中都会出现该新增任务。若任务的执行方式是 ANY，则任意一个合法用户先进行任务签收，并提交任务处理器完成该任务后，该任务在与执行人同级的所有用户工作列表中撤销；若其执行方式为 ALL，则表示该任务须由该级别所有合法用户签收并处理完成后才能进入流程下一步。

推式策略适用于岗位相对稳定，任务完成与角色的对应关系确定性较大这样的工作流系统中。这种职位明确，职责稳定的工作模式，采用推式策略有几个好处：一是推式策略步骤简洁、容易实现，并且算法简单明了，不容易发生错误，在流程任务分配需求复杂性不高的系统中具有明显的优势。二是系统分工明确，模型各部分间的耦合性低，用户权限管理模型完成用户-角色-权限三者的映射关系，工作流执行服务器完成流程任

务与用户权限模型中的权限之间的映射。具体工作过程如图 4.12 所示。

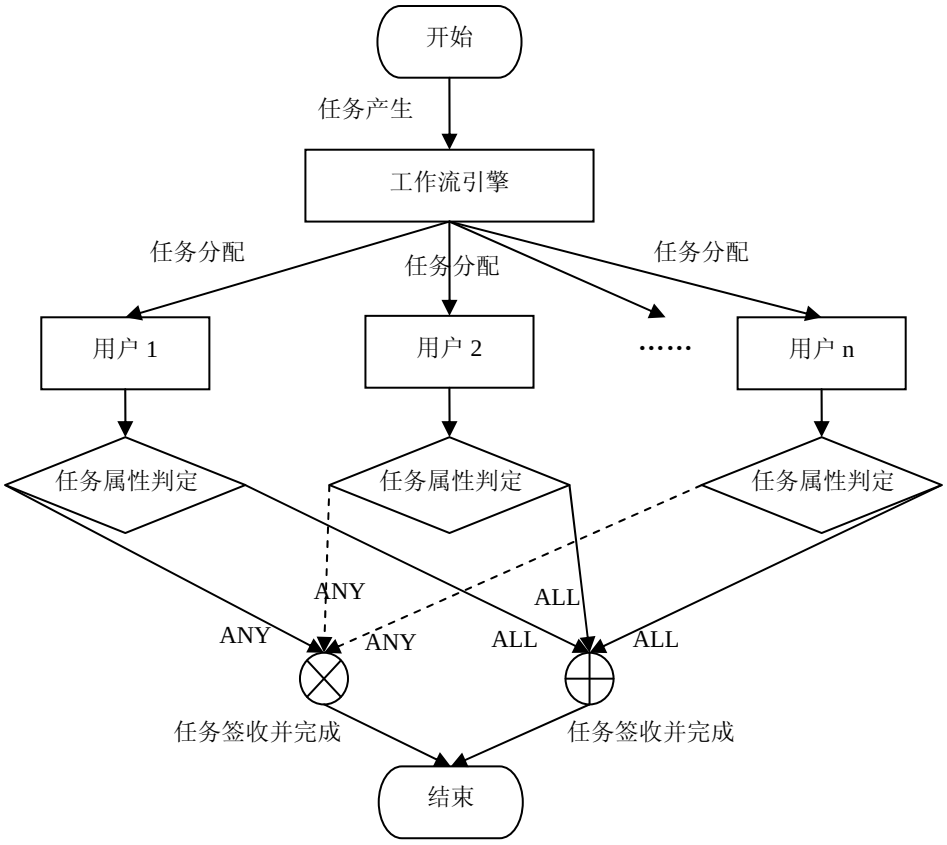

图 4.12 推式策略执行示意图

4.3.3.2 拉式策略

与推式策略不同，该策略是指任务产生后，工作流执行服务器首先不将任务分配给某个用户，而是将任务公布于系统中的一个共享工作列表中，合法用户可以申请该任务，具体执行过程可以参考图 4.13。

任务产生后一定时间，系统对该工单对应的用户的申请做出判定及分配。初步采用按用户的申请时间早晚来体现用户的工作积极性、用户的待办工作任务量来衡量用户的工作负荷、用户的已办工作来衡量用户的工作能力以及流程任务分配者对这些用户一个经验判断等因素作为用户情况的综合考量依据，最终得到这些用户的综合测评分，从而选取出最合适的用户集（一个或者多个），完成任务的分配。

拉式策略的实现较推式策略更为复杂，适用于任务分配过程更为灵活的复杂工作流系统。系统中合法用户通常不止一个，而当这些用户都被赋予合法角色拥有执行任务的权限时，究竟该如何确定最终的任务执行人才能更高效的完成任务，这是用户权限分配中柔性管理的一个重要问题。拉式策略注重用户的主动性，任务产生之后合法用户可以通过申请获取工作，工作流引擎通过用户评分机制判定如何分无任务。当岗位变动可能性大、及对应角色较灵活、岗位人员情况更为复杂时，采用用拉式策略能够更准确更灵活的实现任务的分配。

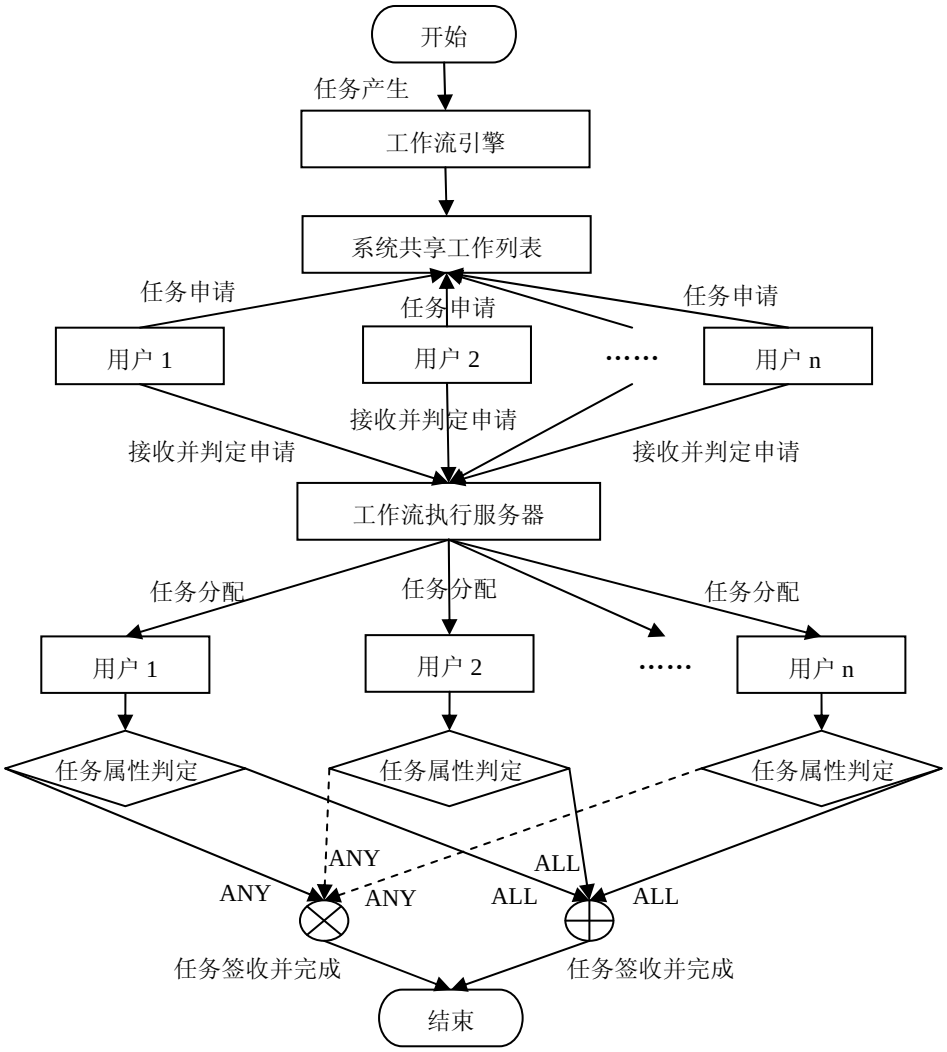

图 4.13 拉式策略示意图

4.3.4 用户授权几类重要的约束

用户权限管理中要保证安全和灵活，以下几个约束条件至关重要：最小权限约束、最小权限时限约束、职责分离约束以及用户负载均衡约束。

4.3.4.1 最小权限约束

用户与角色、角色与权限多对多的关系，决定用户可以被赋予多种角色，进而拥有所赋予的角色相应的多个权限。在实际流程任务分配时，用户拥有完成任务所需要的最小权限 least_Permission（u），即对客体资源实施最小操作去完成任务。相对简单的任务，用户可能只需要激活一个合法角色，但不需要激活角色相应的所有权限，而复杂的任务需要拥有多个角色中的不同权限来完成。定义两个标识变量 Fr 与 Fp，用于判断每个角色及角色相对应的各权限是否被激活。

$$Fr = \{Fr1, Fr2, ..., Frn \mid Fri = YES / NO, i = 1,2,....n = \{count(R)\}\} \quad (4.1)$$

$$Fp = \{Fp1, Fp2, ..., Fpn \mid Fpi = YES / NO, i = 1,2,....n = \{count(P)\}\} \quad (4.2)$$

$$least\_Permission(u) = \{p \in (PR)^* \mid (PR)^* = \{P^* \times R^* \mid P^* \in (PR, Fp), R^* \in (RU, Fr)\}\} \tag{4.3}$$

4.3.4.2 最小权限时限约束

流程开始运行时 Ts，工作流机对合法用户进行授权，而在用户在完成并提交任务时 Tf 或者达到任务完成的最大时限 Tm，工作流机应及时将权限收回 Te。

$$\forall Ar \in USERS(R), Ap \in ROLES(P) \rightarrow At(USERS(P)) = Te - Ts \leqslant Tm - Ts \tag{4.4}$$

其中，Ar 表示被激活的角色，Ap 表示被激活的权限，At（USERS（P））表示用户拥有权限的时间，若任务在最大时限内被完成，则 Te=Tf，否则 Te=Tm。

4.3.4.3 职责分离约束

流程中的某些关键任务，一般不会分配给同一用户，而应该安排不同的执行者完成不同的任务，实现权力分散。在引入用户权限管理模型后，本文从两个方面讨论职责分离约束：角色互斥约束与权限互斥约束。角色与角色的互斥，是指用户可以同时被分配角色互斥组（ROLESC）的多个角色，但不能在同一个流程中同时建立会话、激活互斥组中任何两个以上的角色。权限与权限的互斥，是指角色拥有权限互斥组（PERMISSIONSC）中一个权限后就不能再拥有组内其他权限。

角色互斥约束：

$$ROLESC = \{RC1, RC2, ..., RCn \mid RCi \in R, i = 1,2,....., n\} \tag{4.5}$$

$$\forall ARi, ARj \notin USER(R) \wedge ARi \in ROLESC \rightarrow ARj \notin ROLESC, i \neq j \tag{4.6}$$

权限互斥约束：

$$PERMISSIONSC = \{PC1, PC2, ..., PCn \mid PCi \in P, i = 1,2,....., n\} \tag{4.7}$$

$$\forall Pi, Pj \in ROLES(P) \wedge Pi \in PERMISSIONSC \rightarrow Pj \notin PERMISSIONSC, i \neq j \tag{4.8}$$

其中，AR 为流程任务执行时激活的角色。

4.3.4.4 用户负载均衡约束

用户建立会话激活角色的数量需要受到限制，最大值不能超过 USER（R）中的合法角色总数。在实际流程运行时，负载均衡约束要求衡量用户的实际工作属性（包括用户的工作积极性，现有工作任务量，工作能力），将任务分配给可能最高效率完成该任务的用户（best_User）。

$$ATTRIBUTES(U) = \{POSITIVE, ABILITY, TASK\_COUNT\} \tag{4.9}$$

$$SCORES(U) = \{P * Wp + A * Wa + TC * Wtc \mid Wp + Wa + Wtc = 100\%\} \tag{4.10}$$

$$best\_Users = MAX(SCORES(U)) \tag{4.11}$$

其中，Wp，Wa，Wtc 表示各属性的权重，SCORES（U）表示用户属性综合所得的分值，最后将任务分配得分最高的用户（MAX（SCORES（U））。

4.3.5 用户授权的实现步骤

本文用户权限管理的具体实现主要包括以下四个部分：权限定义、角色定义、权限配置、权限审查。

1. 权限定义：明确定义权限是系统权限管理的首要步骤。在进行角色-权限的配置工作前，必需要对系统中各种对象的访问控制权限给出清晰的定义。权限的定义即对象和数据访问控制的定义，权限通常以一个三元组 P（o，t，p）来描述，其中：o（object）为访问客体；t（type）为访问类型；p（predicate）为谓词。表示在谓词 p 为真时对于对象 o 可进行 t 类型的访问。具体步骤如图 4.14 所示。

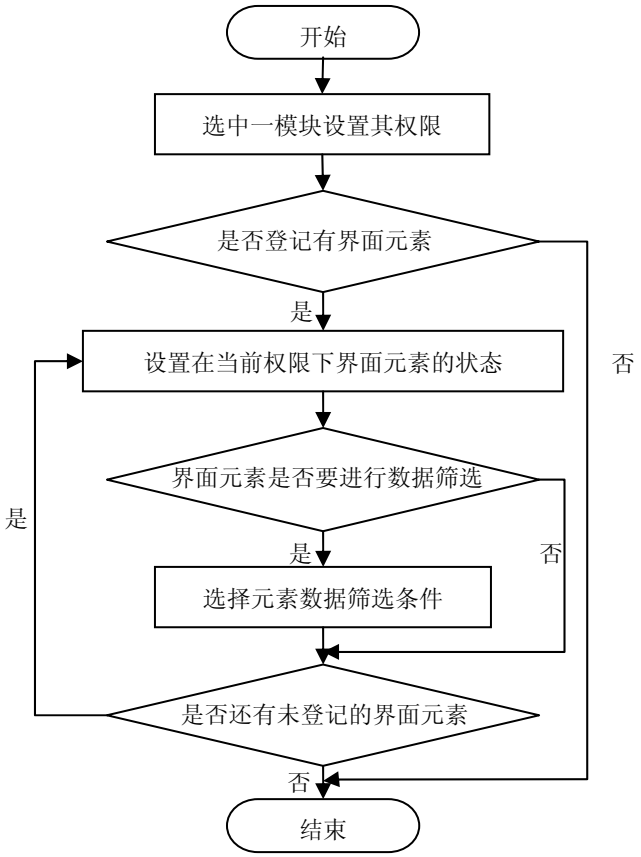

图 4.14 定义权限的流程图

2. 角色定义

基于角色的访问控制核心思想就是提出“角色”这个用户和权限连接的中间机制，形成用户-角色-权限这样一种多对多的关系。本系统为用户提供了定义角色的工具，用户可以根据组织结构、自身的权位、职能等来实现角色的定义。角色定义过程如图 4.15 所示。

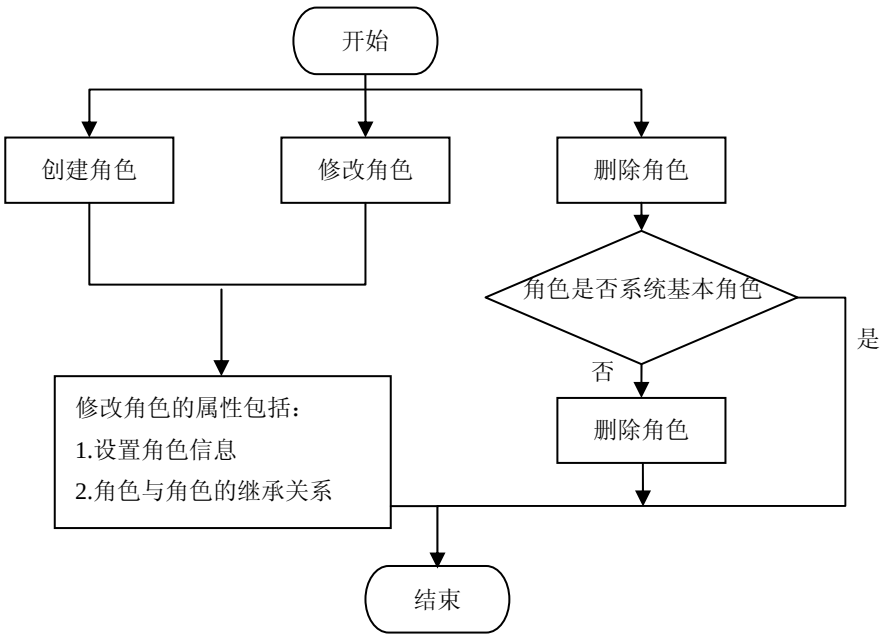

图 4.15 角色定义流程图

3. 权限配置

权限的配置是系统授权管理的重要工作，实际用户只有拥有一定角色从而获取相应权限才能进行访问操作。与组织结构相对应建立的角色通常都对应着这一职能岗位上的一组访问权限，系统中的合法用户可以扮演多重角色，同样，一个系统权限也能被配置给多个角色，权限配置过程如图 4.16 所示。

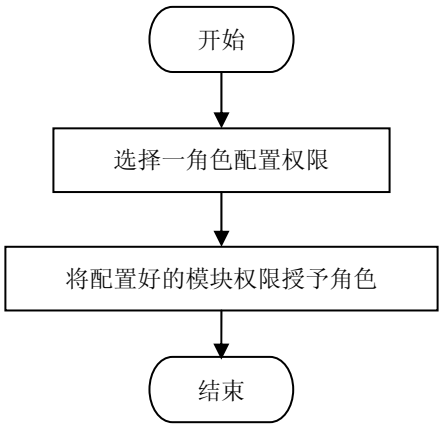

图 4.16 权限配置流程图

4. 权限审查

权限的审查也是用户权限管理主要工作步骤之一，完成权限分配后，系统需要对已授权用户登录系统后的能力表进行检查，确认此权限分配是否合理，若不合理则需要考虑收回这些不合理部分的权限并对其进行重新配置。目前系统在收回部分不合理权限的管理方面还有待改进，只能针对一个用户组内的用户进行权限收回操作。

4.4 小结

本章提出了基于 RBAC 的柔性工作流模型，建立了基空间与元空间的映射机制，提出了推拉式任务分配策略，对流程过程管理和组织用户权限管理这两个方面进行了柔性分析和设计，并给出了具体的实现方法。

第五章 基于 RBAC 的柔性工作流系统应用

5.1 试油试采综合信息管理系统需求分析

5.1.1 项目概要

试油试采分公司综合信息管理系统的是以试油、试采专业动、静态数据库为研究基础，建立一个统一的试油、试采信息综合应用环境，为生产管理人员提供及时应用试油、试采动静态资料进行综合分析判断的技术手段，为及时准确做出决策提供依据，同时为研究人员提供一个以工作流技术为支撑的灵活性适应性高的自动化试油、试采信息综合应用平台。

本项目采用多级模式进行架构，以试油试采分公司业务流程为主线，囊括了业务工作的全过程，借助现有的网络环境和协同思想，实现信息、业务流程、人员的有机结合，使用户轻松构建起一个数据共享、流程同步的综合性信息化工作平台。项目完成后便于管理人员、科研人员了解当前生产运行情况，及时发现问题，为生产决策、地质研究解决问题提供有益帮助。对日常管理、决策分析起到极大的促进作用，将取得较大的直接或间接经济效益。

5.1.2 业务需求分析

近年来，试油试采分公司大力推行信息化建设工作，在现场动态管理方面研制了试油日报、试油、试采专业数据库等相对独立的系统；在成果静态资料方面，基本完成了原有资料的电子化工作，创建了完善的数据库体系，为勘探生产、地质评价提供了有力的技术支持。但也存在一些问题，急需完善解决。

1. 现有的应用系统集成化程度不高、灵活性差

不同时期开发的各应用系统，资料展示内容稳定，设计形式固定。应用人员想要获得相关的全部资料数据，缺少一个有效的统一的工具手段，只能通过多个不同应用系统，在各系统流转不变，集成化程序不高，操作烦琐影响工作台效率。

2. 已有的各系统不能进行动、静态资料关联应用

地质管井人员进行现场跟踪管理，需要了解动态井现场信息，查看一些邻井的成果静态资料，进行多井资料对比分析，指导试油、试采现场油气显示落实情况。目前，还没有一套动静态资料关联应用的系统，可直接获取试油、试采井现场数据，对地质综合研究、及时发现现场油气显示并调整现场方案提供有效帮助。

3. 缺少一个统一的试油、试采信息综合应用平台

目前还没有一套能够集中获取各项试油、试采资料的综合应用平台。随着勘探开发技术的不断创新和改进，数据的内容和形式也不断扩展，灵活、多变的资料组合方式更适合将来资料应用的发展，这需要我们开发一个个性化、开放性、扩展性都很高的信息综合应用系统。

5.2 试油试采综合信息管理系统技术路线

5.2.1 系统开发环境与开发模式

开发环境及工具：本系统以 Windows XP 为开发环境，以 Visual Studio 2008/2010 为开发工具，后台数据库采用 Oracle9i，并结合 XML 和工作流架构技术等。

开发模式：系统基于.Net 开发模式，采用 MVC 体系结构，实现分布式管理及分布式存储。MVC（Model-View-Control），即模型-视图-控制，它是.Net 应用中最常见的体系结构，常用于交互式的 Web 应用，尤其是存在大量的页面访问、客户请求及数据交互时的应用。相比较而言，工作流系统体系结构在流程过程控制和交互较少的情况时应用更多[54]。本系统采用 Web 应用服务器的中间件的技术，采用先进的交互式网上应用技术；采用多层的体系结构，遵循 ASP 和 XML 等先进的国际标准。多层应用框架从体系结构上包括客户端（client Tier）、中间层（Middle Tier）和企业信息系统（EIS Tier，包括数据库、ERP 等）。系统体系结构图如下图 5.1 所示。

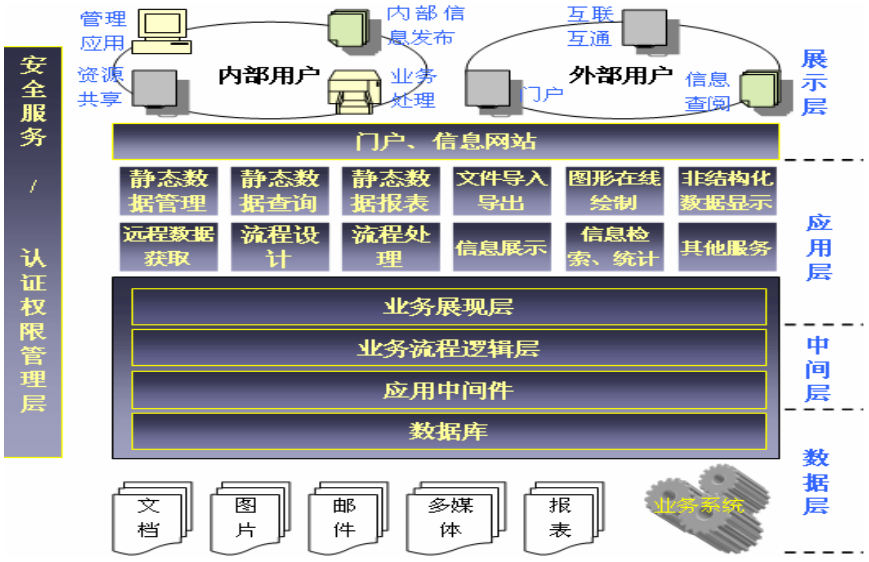

图 5.1 系统体系结构

1. 灵活的客户端：包括多种类型的客户端，在防火墙内的可以是 Application 或浏览器方式，在防火墙外主要是浏览器方式。

2. 功能强大的中间层：如业务逻辑组件运行环境、浏览器运行方式服务器、工作流引擎及各种配套的服务（如：电子邮件服务、名称服务、异步消息服务等）。

3. 丰富的企业信息系统：后台的企业信息系统可以是各种数据库、各种办公自动化系统及其他的传统的信息系统。

5.2.2 系统运行环境

系统对运行环境有一定的要求，具体有如下几方面：

- 网络环境要求：现有局域网环境；
- 计算机硬件环境：可根据需要配置 Windows 服务器；内存 1G 以上；

- 系统运行软件环境：Windows XP、Windows NT 或 Windows 2000；
- 应用软件：建议 IE5.5 以上版本；
- 数据库管理系统：后台数据库类型及版本 Oracle9i。

5.3 工作流过程管理模块的设计与实现

5.3.1 流程的功能组件与数据库设计

5.3.1.1 主要组件与功能机制

流程定义管理机制：本系统流程定义采用 Eclipse3.2 开发平台，通过设计流程环节、合理配置环节各参数、流程实例模拟检测等步骤完成流程可视化定义，并生成 XML 格式的文本流程定义文件（如图 5.2 所示）。本系统提供了 XML 格式的流程定义文件的增删改管理机制，实现对流程定义文件的灵活有效管理。

流程模型设计器机制：本系统对流程设计提供了一个可视化的设计器机制。包含了流程设计所需要的基本组件，并以图形化的方式提供一个设计界面（如图 5.3 所示），用户可以根据实际业务的需要建立新的流程，并同样生成完整的 XML 格式的流程定义文件（如图 5.2 所示）。用户不需有丰富的 XML 语言知识背景，只需掌握基本的设计技术、并梳理清晰流程需求，就可以独立完成新流程的设计。

```
<fpdl:Activity Id="yqcyl.yqcyl0" Name="yqcyl0" DisplayName="油气层压裂地质设计—编写" CompletionStrategy="ANY">
    <fpdl:Description>按照勘探分公司的指令地质大队编写地质部分</fpdl:Description>
    <fpdl:ExtendedAttributes>
        <fpdl:ExtendedAttribute Name="FIRE_FLOW.bounds.height" Value="58"/>
        <fpdl:ExtendedAttribute Name="FIRE_FLOW.bounds.width" Value="152"/>
        <fpdl:ExtendedAttribute Name="FIRE_FLOW.bounds.x" Value="49"/>
        <fpdl:ExtendedAttribute Name="FIRE_FLOW.bounds.y" Value="12"/>
    </fpdl:ExtendedAttributes>
    <fpdl:Tasks>
        <fpdl:Task Id="yqcyl.yqcyl0.edit0" Name="edit0" DisplayName="编写（地质部分）" Type="FORM" CompletionStrategy="ANY" DefaultVie
            <fpdl:Performer Name="R001" DisplayName="地质大队生产室">
                <fpdl:Description>地质大队生产室</fpdl:Description>
                <fpdl:AssignmentHandler>WebLib.Web.WorkFlow.SYSJ.CurrentUserAssignmentHandler, WebLib.Web</fpdl:AssignmentHandler>
            </fpdl:Performer>
            <fpdl:EditForm Name="编辑表单" DisplayName="编辑表单">
                <fpdl:Description>编辑表单</fpdl:Description>
                <fpdl:Uri>~/WorkFlow/SYSJ/sysj11edit.aspx</fpdl:Uri>
            </fpdl:EditForm>
```

图 5.2 XML 流程定义文件图

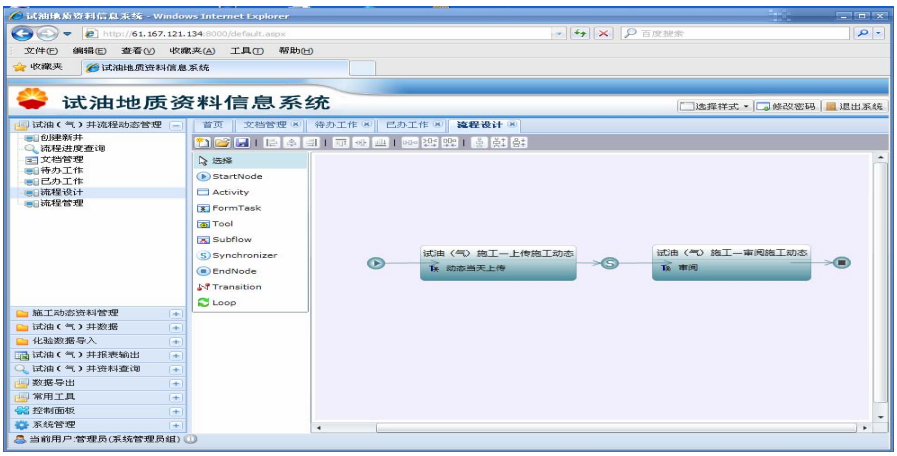

图 5.3 流程设计器界面图

流程实例化机制：启动流程、创建流程实例是流程管理最基本的功能机制。

流程监控机制：流程的监控是符合实际工作需要的一项重要功能机制。伴随着流程执行情况的发展而变动，良好的流程实时监控机制对于一个工作流系统来说相当必要，能使流程参与者及时的获悉流程运行状态，在异常发生时及时采取应对措施，而不至于影响工作的正常进展。

5.3.1.2 流程相关数据库设计

在数据表的设计上，我们遵循以下的规则：系统数据表设计采用第三范式标准，3NF 在平衡数据库性能、扩展性以及数据完整性上效果最好[55]。考虑到数据库可能的各种变化，要求每张表至少有三个字段以上，字段类型尽可能准确，增加删除标记字段。键值的设计合理，遵循键值设计原则，尽量选择系统生成的键作为主键，而不使用用户自定义键，为关联字段创建外键，避免复合键。遵循数据库实体完整性、参照完整性、用户自定义完整性三大完整性约束[55]。采用视图、序列、触发器等，反复检查核对并保证数据库中表、字段、视图、存储过程等结构的命名规范。与流程相关的部分数据表概念模型如图 5.4 所示。

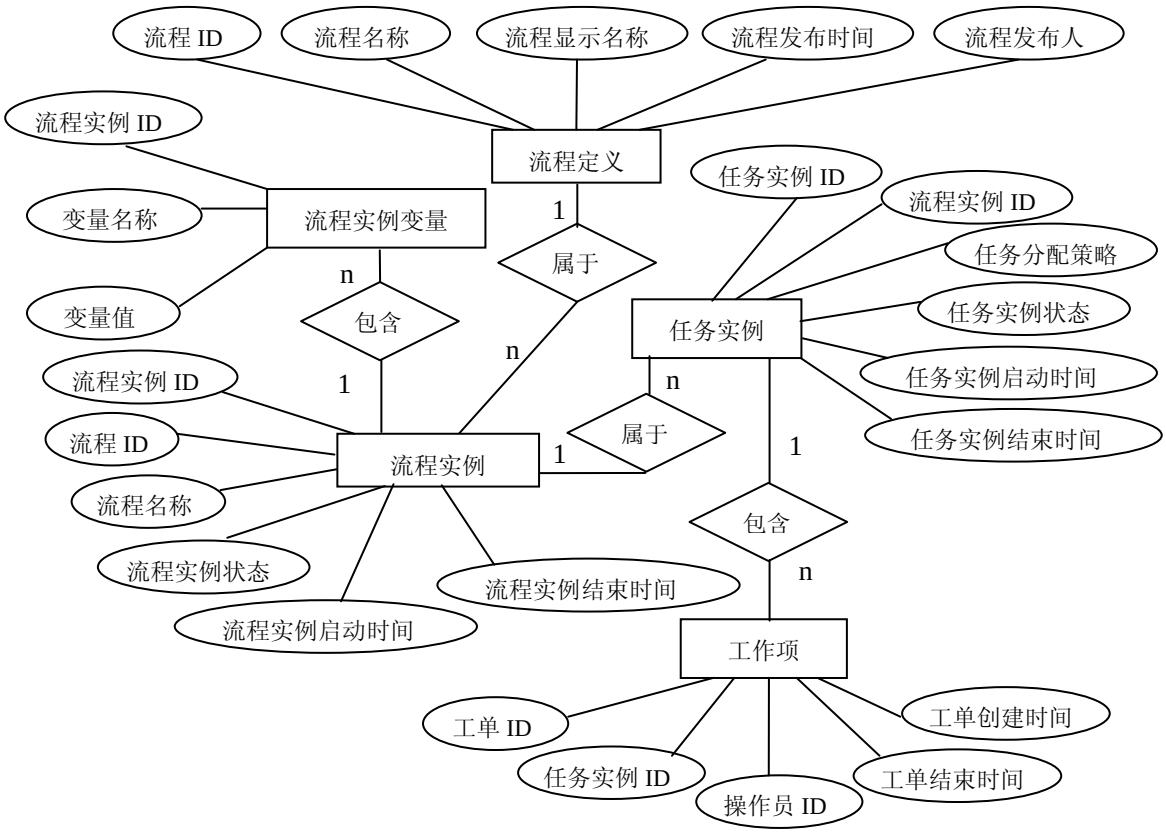

图 5.4 流程相关的部分数据表概念模型图

流程定义：定义完整的流程以 XML 文件保存，在本系统中直接存储为大字段类型。还包含了编号、版本、流程名称、流程创建人、创建时间等字段。

流程实例：流程实例化完成后，实例信息存储在该表中。主要包括实例编号、流程 ID、流程实例状态、开始时间、结束时间等字段。

任务实例：流程实例包含多个流程任务，任务实例包含字段：任务实例 ID、任务

ID，流程实例 ID，启动时间，结束时间等。

工作项：每个任务实例又可能包含多项具体工作，主要字段：工单号，任务实例 ID，操作员，创建时间，结束时间等。

流程变量：在各流程运行时，需要设置一些列相关流程变量，主要字段：流程实例 ID，变量名，变量值等。

5.3.2 系统流程的分析与模型构建

依据本文上一部分给出的试油试采分生产管理流程泳道图和工作流程图，构建出了实际业务过程十七个基本的流程模型，本系统采用 Eclipse3.2 建模工具建构模型，设计其约束条件，并通过模拟器进行测试，保证流程顺畅运行，满足设计目标。几个典型的流程模型图以及相关描述（见图 5.5 及 5.6）。

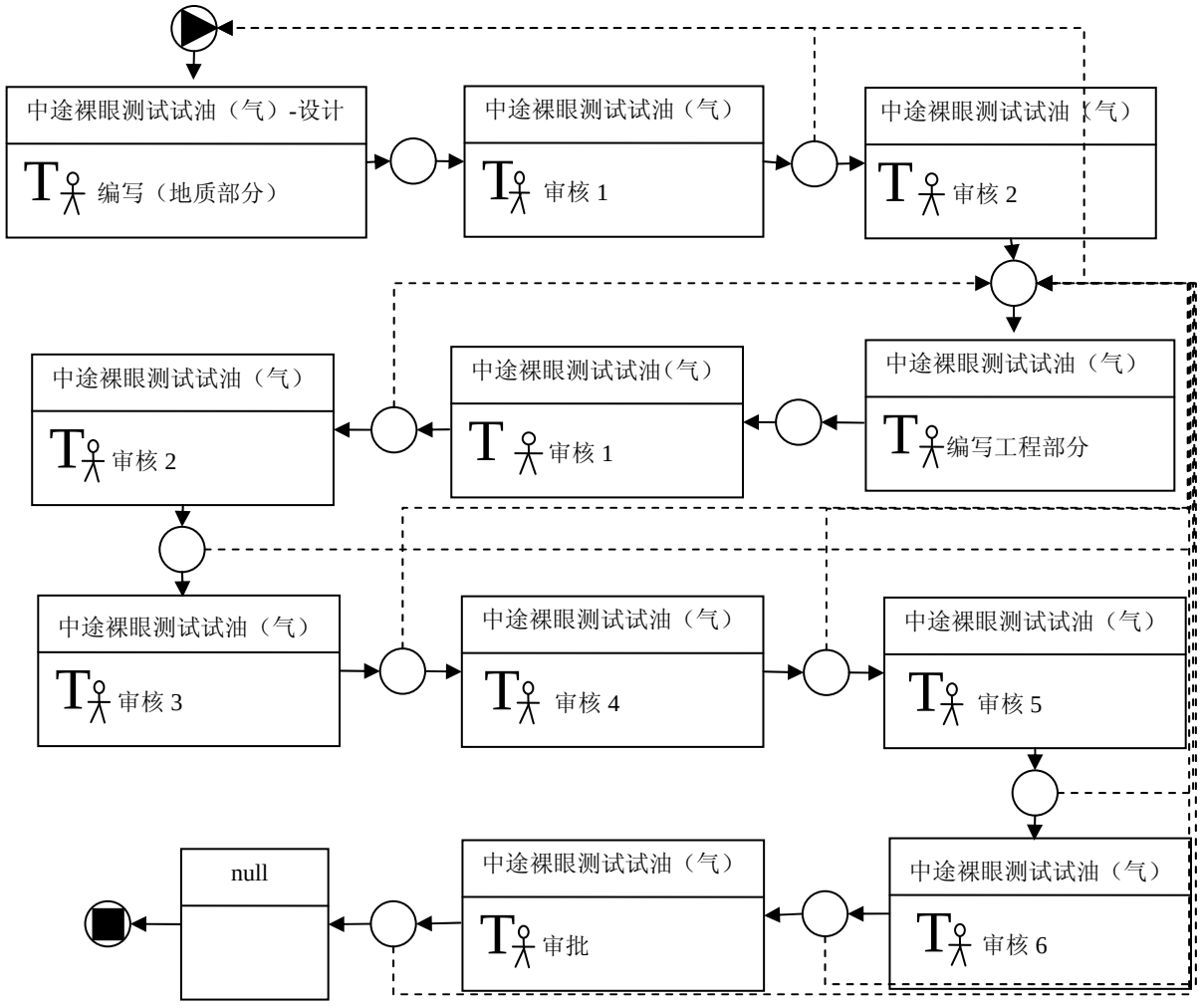

图 5.5 中途裸眼试油（气）设计及审批

工作流程描述：由勘探公司下发通知单，依据钻井地质设计、测井设计、录井设计，查询邻井资料，结合本井的实际情况，了解研究区域地层及油气水分布情况，按照勘探分公司的指令地质部门编写地质部分，试油大队根据地质部分编写工程部分，合并为中途裸眼试油（气）设计。

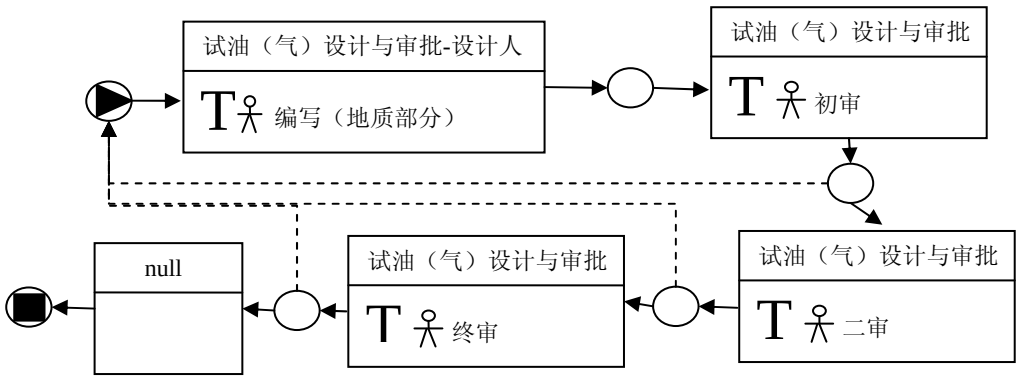

图 5.6 试油(气)施工设计及审批

工作流程描述：依据试油（气）设计及压裂作业指导书，按照勘探分公司的指令编写试油（气）施工设计。

5.3.3 柔性工作流系统运行实例

通过对试采工作流程的工作具体工作任务进行梳理与分类，所包含的主要内容如图 5.7 所示。

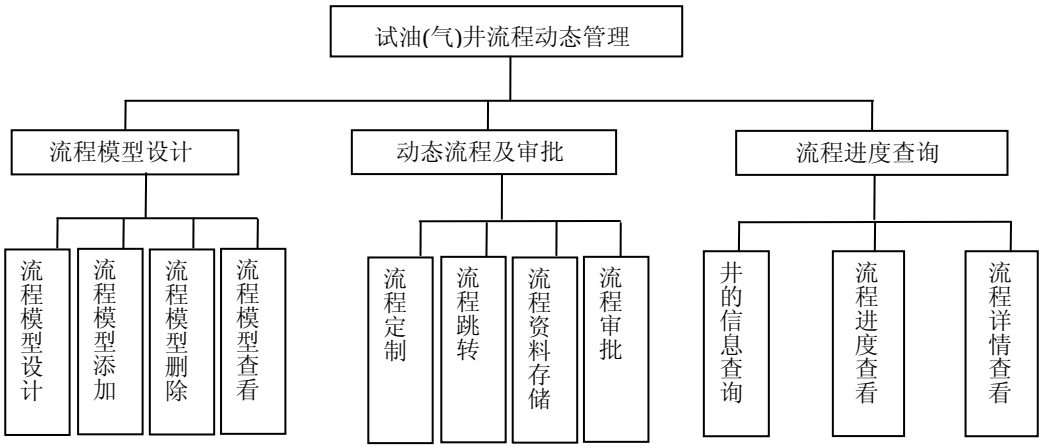

图 5.7 流程动态管理模块图

在实际项目应用中，针对流程的实例化以及流程的执行过程，系统在流程过程管理方面的柔性做了充分的理论研究与实际算法实现工作，主要从流程实例启动时发起人实现流程灵活定制、流程实例运行时各环节合法用户对流程流转走向的柔性控制两个方面来阐述。详见下列系统界面图。

5.3.3.1 流程定制

下面以压井工程设计及审批这一流程为例，描述系统如何实现流程的定制。具有相关权限的合法用户登录系统之后，启动一个新的流程实例，完成基础数据的录入，流程流转中涉及到的相关资料的上传。实例发起人开始根据本次实例提出的具体需求，完成流程的定制，如图 5.8 所示。

流程的定制是指以流程模型为基本参照，一个流程模型通常要求尽可能全的包含业务过程的所有环节。而在实际工作中，启动者依据业务需求，决定该流程实例需要选取执行的流程步骤，这些环节可以是不连续的，但要求至少一个、至多为模型总环节数。

这样每次启动的流程实例结构都不尽相同，但都是流程模型完整实例的子集，灵活的流程定制是本系统柔性的一个体现，不需要的流程步骤可以被略过，简化了工作的复杂性，提高了工作流系统对实际业务需求的支持。

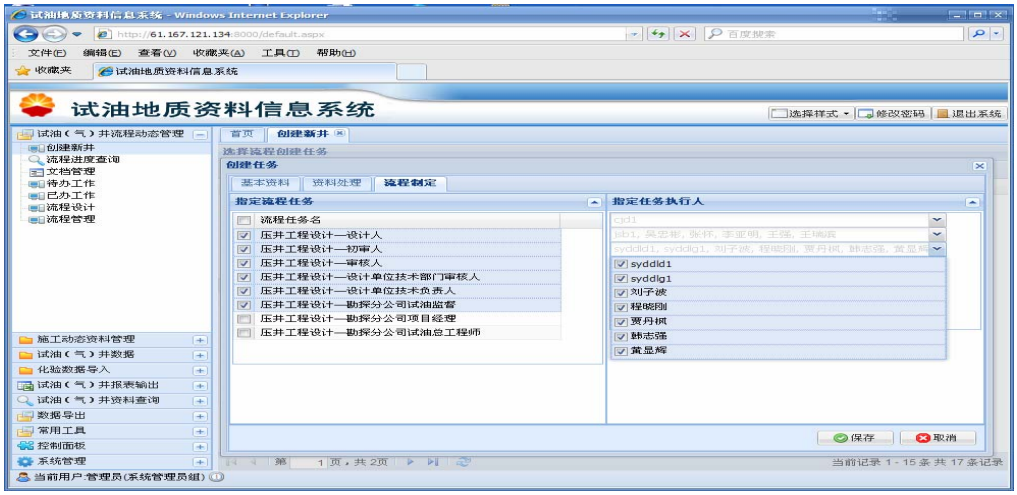

图 5.8 流程定制界面

5.3.3.2 流程走向的柔性控制

流程实例启动之后，流程在各环节中正常运转。根据第四章的分析，我们知道，流程运转过程中存在着一些不确定的因素，容易引起业务过程的变动，从而会对正在执行的工作流程有动态调整的需求。本系统在每个流程执行人完成流程基本任务（如签收、审核、提出意见等操作）之后，如需要对该流程的走向做出调整，即可以通过本系统提供的相应接口，选择流程下一个执行环节。默认为流程按照定制时的活动先后关系流转到下一环节；若需要略过某一些环节，或是新增某些执行环节，以及可以跳回到之前的环节，则用户可以灵活的设置下一个执行环节，从而能改变原有流程实例的流转路径，如图 5.9 所示。

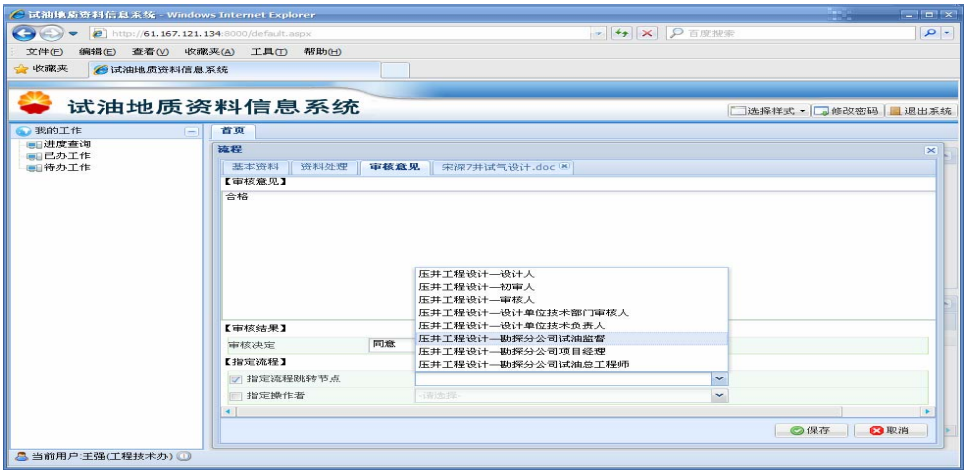

图 5.9 流程路径的柔性控制

5.3.3.3 流程监控

在流程动态管理模块中，还有一个重要的机制是流程的监控。必需要对流程实例的运行情况进行监控、分析，才能知道流程任务是否按要求被合法用户完成，从而保证业

务目标的顺利实现[56]。新的流程实例启动后，系统提供了流程进度查询功能，实现对流程进度的实时监控。在本系统中监控主要分为分两个粒度。

1. 是以井为单位进行的监控。以井号与重开次数作为查找依据，分为已施工与未完成井两大类进行展示，监控与该井相关的所有流程执行情况，以及查看井是已完井还是正在施工井，列表显示该井指定执行的所有流程状态即是否已启动、各流程的进展情况、各流程当前的执行人，如图 5.10 所示。

图 5.10 井相关流程执行过程监控

2. 以流程为单位的监控。这是更细级别的监控，能及时的反应出具体的某一流程实例进行到哪一步骤、各个步骤完成情况、各个步骤的执行人以及流程运转中相关的一些流程变量值等基本情况，如 5.11 图所示。

图 5.11 流程实例级监控

流程监控相关数据表对井的流程进行监控，需要对流程执行情况一些相关数据，进行实时的增加和更新。设计两个相关数据表，进行流程实时数据的记录。图 5.12 为流程监控表关系模型图。

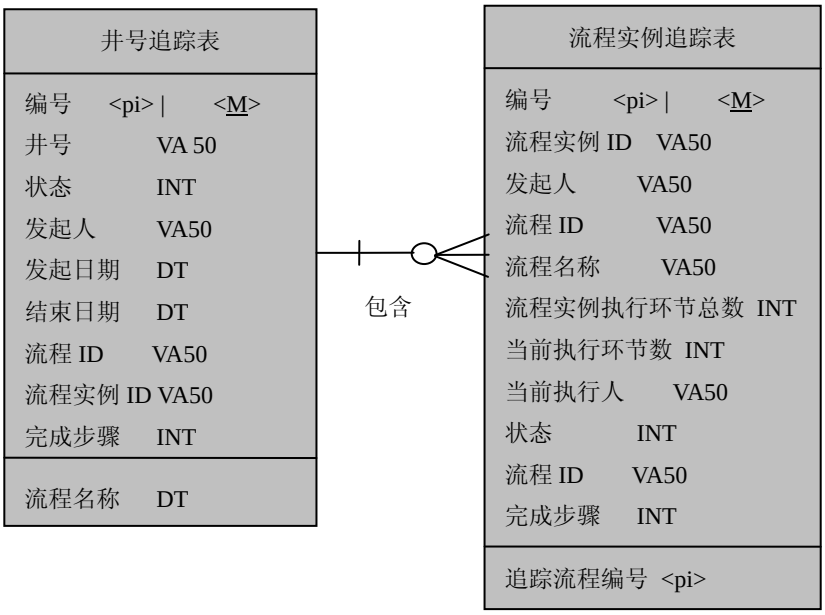

图 5.12 流程监控相关表关系模型

井号追踪表：即以井号为关键字建立的流程实例，每口井包含一个或者多个流程实例。主要字段还包含：发起人、流程 ID、流程实例编号、发起和结束日期、井所处状态等。

流程实例追踪表：更细级别的实现对流程的实时监控，主要包含流程 ID、流程名称、实例 ID、实例总环节数、当前执行环节数、实例运行状态等字段。

5.3.4 数据结构与主要操作

通过本章前两个小节的分析与图示，大致表述了本系统中工作流程过程管理模块的功能与组件机制，这一小节继续对流程管理模块实现与流程定义、流程实例、流程主要元素相关的主要算法与主要数据结构做详细描述。

1. 主要数据对象：

```
public class WorkflowDefinitionInfo //流程定义类
    {
        public const String FPDL_PROCESS = "FPDL"; //定义文件格式
        public String Id { get; set; } //流程定义文件 ID
        public String ProcessId { get; set; } //流程 ID
        public String Name { get; set; } //流程名
        public String DisplayName { get; set; } //流程显示名
        …
     }
public class ProcessInstance : IProcessInstance, IRuntimeContextAware,
IWorkflowSessionAware //流程实例类
    {    public String Id { get; set; } //流程实例 ID
```

```
        public String ProcessId { get; set; } //流程定义 ID
        public Int32 Version { get; set; } //流程版本号
        public String Name { get; set; } //流程实例名
        public String CreatorId { get; set; } //流程实例创建者
        public DateTime CreatedTime { get; set; } /流程实例创建时间
          …
    }
public class TaskInstance : ITaskInstance, IAssignable, IRuntimeContextAware,
IWorkflowSessionAware //流程任务实例类
  {
        public String Id { get; set; } //任务实例 ID
        public String TaskId { get; set; } //任务 ID
        public String Name { get; set; } //任务实例名
        public String DisplayName { get; set; } //任务实例显示名
        public String ProcessInstanceId { get; set; } //任务实例流程实例 ID
        public String ProcessId { get; set; } //任务实例对应的流程 ID
        …
    }
public class Activity : Node //流程活动类（即流程节点子类）
  {     public Transition EnteringTransition { get; set; } //当前活动输入弧
        public Transition LeavingTransition { get; set; } //当前活动输出弧
        public List<TaskRef> TaskRefs { get; set; } //该活动的任务列表
        …
    }
public class Edge : AbstractWFElement //流程边类
   {
        public Node FromNode { get; set; } //当前边的出发节点
        public Node ToNode { get; set; } //当前边的到达节点
        public String Condition { get; set; } //从前节点到后节点边转化条件
    }
```

2. 主要操作：

1) 程实例、流程任务实例、工单等包含创建、启动、激活、终止、挂起、恢复等一些操作基本操作：

```
IProcessInstance createProcessInstance(String workflowProcessName, String
creatorId); //创建流程实例
IProcessInstance abortProcessInstance(String processInstanceId);
```

//终止流程实例

IProcessInstance suspendProcessInstance(String processInstanceId);

//挂起流程实例

IProcessInstance restoreProcessInstance(String processInstanceId);

//恢复挂起态的实例

void archiveTaskInstances(IActivityInstance activityInstance); //激活任务实例

2) 流程实例、流程任务实例以及工单相关的一些查找操作方法如下：

IProcessInstance findProcessInstanceById(String id); //通过 ID 获取流程实例

ITaskInstance findTaskInstanceById(String id); //通过 ID 获取任务实例

List<IProcessInstance> findProcessInstancesByProcessId(String processId);

//查找一个业务流程的所有流程实例

List<ITaskInstance> findTaskInstancesForProcessInstance(String processInstanceId, String activityId); //查找某一流程实例的某一环节上的所有流程任务实例

List<IWorkItem> findMyTodoWorkItems(String actorId, String processInstanceId);

//查找某个操作员在某一流程实例上的所有操作工单

3) 与流程跳转执行相关的一些操作方法如下：

void completeWorkItemAndJumpTo(String workItemId, String targetActivityId); //结束当前工单，并指定下一执行环节

void completeWorkItem(String workItemId, DynamicAssignmentHandler dynamicAssignmentHandler, String comments);

//结束当前工单，并动态指定下一环节操作人

void completeWorkItemAndJumpTo(String workItemId, String targetActivityId, DynamicAssignmentHandler dynamicAssignmentHandler, String comments);

//结束当前工单，并指定下一执行环节以及环节执行人

5.4 用户权限管理模块的设计与实现

5.4.1 系统的组织结构设计

权限控制模型主要有如下三类：SO（subject-object view of access control）模型、以任务为基础的授权控制模型（Task-Based Authorization Model）、以角色为基础的访问控制模型（Role-Based Access Control Model）。RBAC 模型包含 5 个基本元素[43]：使用者（Users）、角色（Roles）、许可（permissions）、资源客体（objects（OBS））、操作（operations（OPS））。当然，额外的还包含了两个概念：分配（assignment）和会话（session）。RBAC 还有另外的几种模型，比如 Hierarchal RBAC，Constrained RBAC 等。在工作流系统中采用以角色为基础的访问控制模型（RBAC），图 5.13 是工作流系统的对象模型图。

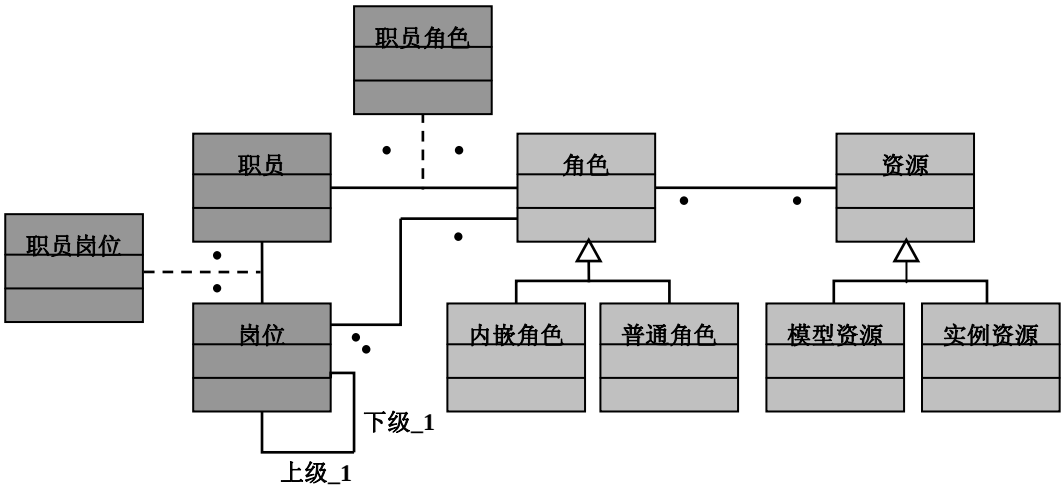

图 5.13 工作流系统对象模型图

1. 对职员的岗位授权：指定岗位所具有的权限，然后指定职员所属岗位，这种授权方式是根据职员的岗位来授权的，如果职员离岗或换岗，相应的权限随岗位不同发生改变。岗位授权可以指定岗位代理人。此处的岗位相当于角色组（具有一组角色）。

2. 对职员直接授权：该授权是根据职员本身的属性或状态授权。比如：根据员工状态授权（试用员工和正式员工的权限不同），根据级别（科级干部和厅级干部的权限不同），直接授权不应指定代理人。如图 5.14 所示。

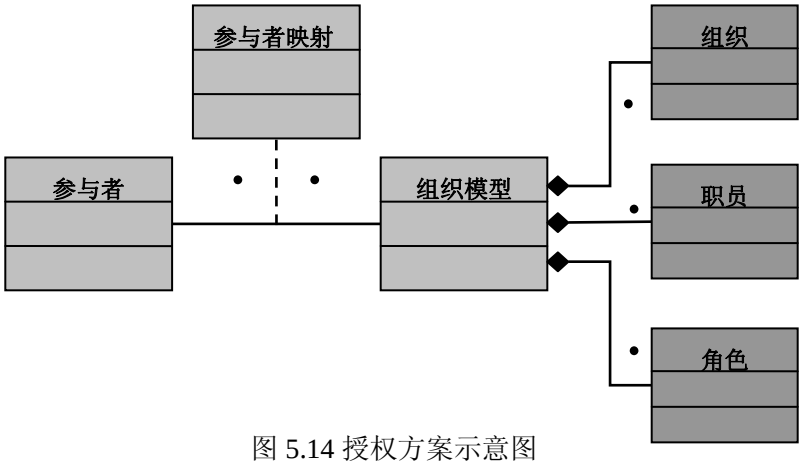

图 5.14 授权方案示意图

3. 代理权限：在上面模型中给出了“职员岗位”、“职员角色”两个关联实体，可以定义代理的属性，指定权限的代理岗位和代理职员，这样代理岗位就拥有被代理岗位的权限，代理的职员拥有被代理职员的权限。

在实际应用系统中，主张以岗定人，以岗定权，所以分配权限时应以第一种方式为主，第二种方式只是辅助性的，是对第一种方式进行补充操作。

5.4.2 用户权限柔性管理的实现

用户权限的管理即用户拥有一定的角色去完成相应的任务，是伴随着流程过程执行而发生的。通过第四章的分析与设计，在本系统中，在流程定制与流程路径更改时都相应提供了对用户权限的柔性管理。在流程定制时，发起人能对定制好的执行环节对应的

合法执行人进行选择，而其它未被定制的环节对应的执行人选项是灰色即处于不可用状态。在流程运行过程中，与制定下一环节相对应，本系统提供了指定环节执行人的功能，合法用户列表与选择的环节保持一致。并在这些分配策略中充分考虑了最小权限约束、最小权限时限约束、职责分离约束以及用户负载均衡约束等。

进行用户合理分配的策略较为灵活，本系统综合采用了推式（PUSH）与拉式（PULL）两种策略，以流程模型的复杂性以及实际业务过程的需求为依据，很好的实现了用户权限的柔性分配。策略选取的影响因子很多，本系统主要以流程实例的环节总数和流程实例某环节的合法用户数为依据，分别对二者给定对应参考标识，用以决策如何恰当的选用分配策略。通过对实际需求分析，本系统选用“生产室主任”这一岗位对应的角色“R00201”进行任务分配时采用拉式策略，而其余角色采用推式策略。系统按拉式策略完成了用户申请任务、引擎判定用户申请并完成任务的分配等工作，系统中给出了一个这样的决策过程：通过对 applyTime 申请时间（判定工作积极性，权重值为 1/2，设定时间基值为 10 分钟）；mytodoWork 待办工作（判定用户工作负荷，给定权重值为 1/4，设置基本待办任务量为 2）；havedoneWork 已办工作（判定用户工作能力，权重值 1/4，完成任务量基值为 5）。对所有提出申请的用户进行评分，最终按得分高低选取最恰当的用户，同意申请并对其分配任务。

当流程实例环节不多，各环节的执行人较少且通常为固定的几个人时，就采用 PUSH 策略，这样能减少引擎的工作负担和系统的出错率，从而提高工作效率。反之，面对流程过程较复杂，执行人变动性较大的流程实例，则采用 PULL 策略，更好的保证用户授权安全灵活的满足系统的实际需求。以下为系统指定执行人界面，分别如图 5.15 和图 5.16 所示。

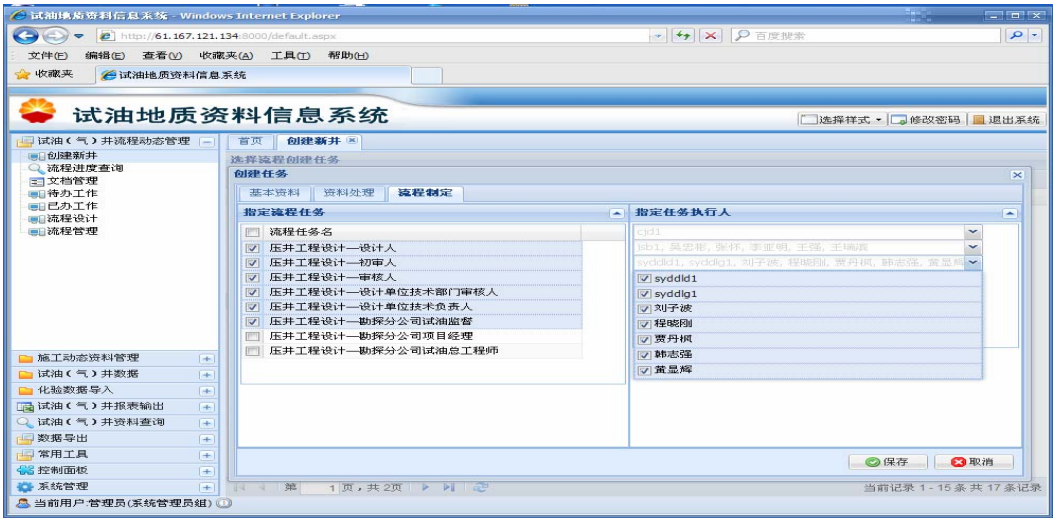

图 5.15 流程定制阶段指定各环节执行人

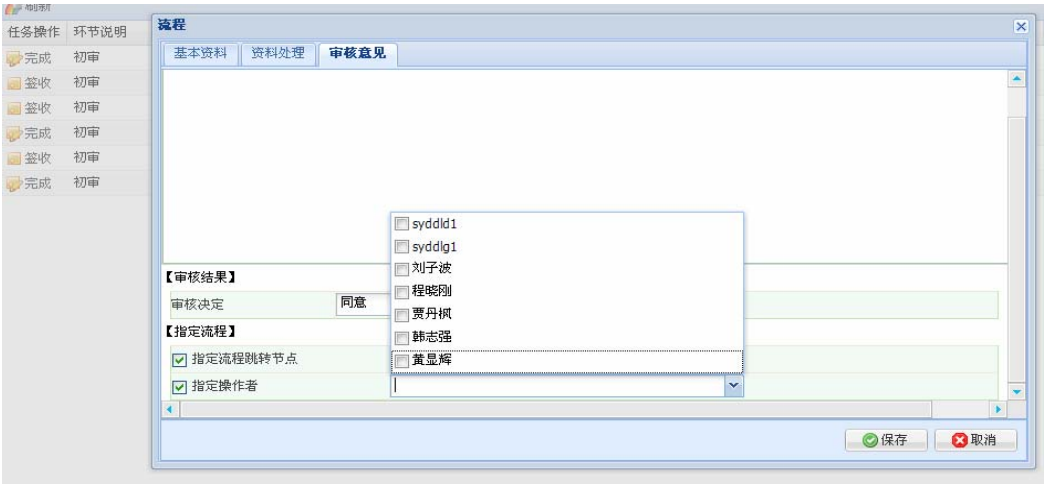

图 5.16 指定跳转节点审核人

5.4.3 数据结构与主要操作

通过 5.4.1 与 5.4.2 两个小节对本系统中用户权限管理模块的主要部分与工作界面的介绍，本节将对实现用户权限管理的主要算法与数据对象作进一步分析。

1. 主要数据对象：

```
public class ROLE //角色类
    {
            private int _ID; //角色 ID
            private string _ROLEID; //角色编号
            private string _ROLENAME; //角色名称
            …
      }
public class MEMBER //用户类
    {
            private int _ID; //用户 ID
            private string _USERID; //用户编号
            private string _USERNAME; //用户名称
            private string _PASSWORD; /用户密码
            …
       }
public class POWER //权限类
    {
            private int _ID; //权限 ID
            private string _MENUID; //菜单 ID
            private string _POWERUERID; //权限对应用户 ID
            private int _POWERTYPE; /权限类型
```

```
            …
        }
    public class ROLEPOWER //角色权限类
        {
            private int _id; //角色权限对应关系 ID
            private string _userid; //用户 id
            private string _roleno; //角色编号
            private int? _valueid; //权限 id
            …
        }
    public class USERINROLE //用户激活角色类
      {
            private int _ID; //用户激活角色对应关系 ID
            private string _USERID; //用户 id
            private string _ROLEID; //角色 id
            …
        }
```

2. 主要操作：

1) 用户、角色、权限、部门等都有基本的增删改操作，这里角色以为例：

```
public void AddRole(Entity.ROLE model) //增加角色
public void EditRole(Entity.ROLE model) //编辑角色
public void DeleteRole(string roleid) //删除角色
```

2) 用户授权管理中一些其他的重要操作方法：

```
public IList GetUserRoles(Hashtable ht) //获取用户对应角色
public DataTable GetRoleUsers(Hashtable ht) //获取角色对应用户
public WebLib.Sys.Data.Entity.ROLEMENUGetModel (string ROLENO,string
MENUID) //通过角色 id 与菜单 id 获取对应关系实体
public  Data.Entity.USERDEPT  GetModel(string  userid)  //获取用户部门对应关系
实体
public class RoleBasedAssignmentHandler : IAssignmentHandler
    {
        public void assign(IAssignable asignable, string performerName)
            {……} //角色的基础分配
    }
public class CurrentUserAssignmentHandler : IAssignmentHandle r
    {
```

```
        public void assign(IAssignable arg0, String arg1)
            {……} //当前活动执行人分配
    }
```

5.5 小结

本章完成了基于 RBAC 的柔性工作流管理系统的开发，对项目的背景与需求做了分析，阐述系统的开发环境与运行环境。实践第四章给出的提高系统柔性的各种理论方法，并在此实际项目应用中得到了很好的检验与完善。

结　论

本文从流程过程管理与用户权限管理两个方面出发，探讨如何实现工作流系统的柔性。以用户授权为结合点，将 RBAC 技术引入到工作流管理系统中，构建了基于 RBAC 的柔性工作流模型，提高了系统访问控制方面的灵活性与安全可靠性。分别从节点与路径考虑，实现了柔性的流程定制与流程路径的灵活选择。同时，完成了基于 RBAC 的柔性工作流系统的开发，并在试油试采综合信息系统应用中验证了其可行性、有效性和实用性。

一、本文工作总结如下。

1. 详细分析了工作流参考模型的六大组成部分、五个主要接口以及工作流系统体系结构，全面研究了 RBAC 技术，重点分析工作流系统中用户、角色、权限三者的关系，并对柔性工作流的工作机制做了分析，明确了如何提高工作流系统的柔性。

2. 针对工作流系统在用户权限管理方面的不足，以用户授权为切入点，将 RBAC 技术引入到原有工作流系统中，构建了基于 RBAC 的柔性工作流模型，对用户权限管理模型中的元素（如用户、角色、权限、约束等）进行了描述与定义。提出了推式和拉式柔性分配策略，并结合了最小权限、最小权限时限约束、职责分离、用户负载均衡等约束条件去规约用户的授权，兼顾灵活性与安全性，实现了用户权限柔性的管理。

3. 从节点与路径两方面入手，实现了工作流过程管理方面的柔性。通过对系统的主要对象进行有效划分，建立起类似数据库映射机制的基空间和元空间，依据元对象协议把对元对象的各种处理映射到基空间中的具体过程实例上，降低了流程定义、流程模型与流程运行的耦合度，实现了流程实例的灵活定制。同时，支持运行过程中流程环节的增设、删除及跳转，达到灵活选择流程路径的目的。

4. 基于所构建的柔性工作流模型，对实际工作过程进行分析、抽取及合并，实现了工作流定义，构架了系统所需的组织机构，将上述的理论研究在大庆油田试油试采分公司综合信息管理系统中进行了应用，实现了流程路径的灵活选择和任务角色的柔性分配，并取得了较好的效果。

二、需要进一步进行的研究。

本论文将 RBAC 技术引入到原有工作流模型中，在工作流系统柔性方面取得了初步的进展，但仍有很多问题尚未解决，希望在今后研究中进一步完善。

1. 工作流系统的柔性主要是指在系统应对不确定因素时的灵活度与适应性，本文提出了从两个方面来改进系统适应未知情况发生时的能力，但对于不确定因素本身未做出深入研究。对不确定因素的类型、产生原因、发生频率等的深入剖析，也将更有利于来研究如何应对这些意外情况，从而找到更好的切入点，提出更好的解决方案。

2. 本文对于推式（PUSH）和拉式（PULL）两个柔性策略的分析与应用还不够全面。企业组织结构是一个庞大的体系，而在其中的用户-角色-权限分配是一个相当复杂的关系，任务的分配、执行人的选取是相当重要的。所以应该给出更充分的柔性分配策

略的影响因子，设计出更为准确的算法，用以指导系统的权限分配，实现更合理更灵活的授权管理，将是一个很有意义的研究内容。

参考文献

[1] 罗海滨, 范玉顺, 吴澄. 工作流技术综述[J]. 软件学报, 2000,(07).

[2] 范玉顺. 工作流管理系统基础[M]. 北京:清华大学出版社, 2001.

[3] 周建涛, 史美林, 叶新铭. 柔性工作流技术研究的现状与趋势[J]. 计算机集成制造系统, 2009,11(11):1501~1510.

[4] 王东勃, 王润孝, 仝勖峰, 邱方亮. 基于动态结构的柔性工作流建模方法研究[J]. 计算机应用研究, 2009,(06).

[5] 邓集波, 洪帆. 基于任务的访问控制模型[J]. 软件学报, 2008,(01).

[6] Palaniswami D. Development of WebWork: METEOR2' s web-based workflow management system [MS Thesis].University of Georgia, 1999.

[7] Alonso G, Agrawal D, Abbadi E Aet al. Exotica/FMQM: a persistent message-based architecture for distributed workflow management. Technical Report, RJ9912, IBM Almaden Research Center, 2008.

[8] 刘培江, 傅秀芬, 陈长瑶. 基于主动数据库的工作流管理系统[J]. 计算机工程, 2008,(06).

[9] 冯卫兵, 郝克刚. 基于 Petri 网的工作流模型的分析[J]. 计算机工程与应用, 2009,(03).

[10] Van der Aalst WMP. The application of Petri nets to workflow management .The Journal of Circuits, Systems and Computers, 2009, 8 (1):21~66.

[11] Cai Ting, Gloor A, Nog S. Dataflow: A Workflow Management System on the Web Using Transportable Agents .Technical Report PCS-TR96-283. Dartmouth College, 2008.

[12] 宋宝燕, 王菊英, 于戈. 基于图形展开及图形归约的过程模型验证方法[J]. 小型微型计算机系统, 2008,(06).

[13] 李琳, 柴争义, 张丽. 基于扩展有向图的工作流建模的设计与应用[J]. 计算机应用, 2010,(09).

[14] 孙亚忠. 基于 Web 的工作流管理系统研究[D]. 武汉理工大学, 2009.

[15] 李巍. 自由审批流工作流引擎建模与实现[D]. 北京化工大学,2011.

[16] 周建涛, 史美林, 叶新铭. 一种基于 Petri 网化简的工作流过程语义验证方法[J]. 软件学报, 2008,(07).

[17] 于春生, 聂晶. 基于组和角色的工作流权限访问控制模型[J]. 计算机应用, 2011,(03).

[18] Workflow Management Coalition. Workflow management coalition terminology and glossary. Technical Report, WfMC-TC-1011, Brussels: Workflow Management Coalition, 1996.

[19] Mohan C. Recent trends in workflow management products, standards, and research. 2008. http://www.almaden.ibm.com/cs/exotica/wfnato97.ps

[20] Alonso G, Agrawal D.et al. Functionality and limitations of current workflow management systems.2009,http://www.almaden.ibm.com/cs/exotica/wfmsys.ps

[21] Rusinkiewicz M, Sheth A. Specification and execution of transactional workflows. In: Won Kim ed. Modern DatabaseSystems: The Object Model, Interoperability, and Beyond. Reading, MA: Addison Wesley Publishing Company, 2008.

[22] 张月菊, 王涛, 林拉. 跨组织工作流集成中间件语义转换部件研究[J]. 计算机技术与发展, 2010,(03).

[23] Workflow Management Coalition:The Workflow Reference Model (WFMC-TC00-1003Issue1.1) .1995.

[24] Workflow Management Coalition:Workflow Management Coalition Terminology and Glossary(WFMC-TC-1011Issue3.0) .1999.

[25] David Hollingsworth. The Workflow Reference Model:10Years On .Workflow Handbook 2004. 2008:295~312.

[26] Wang S,Sheng W,Hao Q. Agent based workflow ontology fordynamic business process composition[C] .Proceedings of the Ninth International Conference on Computer Supported Coopera-tive Workin Design, New York: IEEE Press, 2008:452~457 .

[27] 李慧芳, 范玉顺. 工作流系统时间管理[J]. 软件学报, 2007,(08).

[28] 曾文英, 赵跃龙. 基于移动计算的移动 Agent 技术研究[J]. 微电子学与计算机, 2009, (02).

[29] 郝丽波, 李建华, 夏明伟. 工作流事务性研究综述[J]. 计算机工程与设计, 2008, (13).

[30] 孙瑞志, 史美林. 支持工作流动态变化的过程元模型[J]. 软件学报, 2007, (01).

[31] 李竞杰，王维平，杨峰.柔性工作流理论方法综述.[J].计算机集成制造系统,2010,59(1l):1569~1578.

[32] HAN Yanbo, SHETH A, BUSSLER C. A taxonomy of adaptive workflow management[C]//Proceedings of Workshop of 1998 ACM Conference on Computer Supported Cooperative Work. New York,N.Y. USA: ACM, 1998.

[33] HEINL P, HORN S, et al. A comprehensive approach to flexibility in workflow management systems[C]//Proceedings of International Conference on Work Activities Coordination and Collaboration. New York,N.Y. USA: ACM, 2009, 79~88.

[34] VAN DER AALST W M P, BASTEN T, VERBEEK H M W, et al.Adaptive workflow: on the interplay between flexibility and support[C]// Proceedings of 8st International Conference on Enterprise Information Systems. Norwell, Mass.USA :Kluwer Academic Publishers, 2007:61~68.

[35] NURCAN S. A survey on the flexibility requirements related to business process and modeling artifacts[C]//Proceedings of International Conference on Systenm Sciences.Washington D.C.USA:IEEE 2008,378~387.

[36] WEBER B, SADIQ S, REICHERT M. Beyond rigidity-dynamic process lifecycle support[J], Computer Science R&D, 2009, 23(2): 47~65.

[37] 梁俊锋，张祖平，龙军．动态工作流挖掘模型及算法研究[J]．计算机科学，2011,(01).

[38] 邢光林，洪帆．基于角色和任务的工作流访问控制模型[J]．计算机工程与应用，2008,(02).

[39] 蔡国永，林煜明. RBAC 模型的扩充及其应用[J]．计算机工程与应用, 2008,(03).

[40] 赵勇，刘吉强，韩臻，沈昌祥．基于任务的访问控制模型研究[J]．计算机工程，2008,(05).

[41] 王小明，付红，张立臣．基于属性的访问控制研究进展[J]．电子学报, 2010,(07).

[42] Ferraiolo D，Sandhu R，Gavrila S. ProPosed NIST Standard for Role 一 Based Access Control. ACM TransactionS on Information and System Security, 2001.4(3):224~274.

[43] Sandhu R S. Role-based Access Control Models[J] .IEEE Computer, 2008, 29 (2) :38~47.

[44] Chang-JooMoon，Dae-HaPark，Soung-JinPark，Doo-KwonBaik. Symmetric RBAC model that takes the separation of duty and role hierarehies into consideration. Computer and Security 23，2008:1~15P.

[45] 付松龄，谭庆平．基于任务和角色的分布式工作流安全模型.国防科技大学学报.2004, 26(3):58~59 页.

[46] 汤象峰．基于 RBAC 的动态工作流系统的研究与应用[D]．武汉：武汉理工大学，2010.

[47] 王凯，白庆华．面向对象工作流管理系统模型设计[J]．计算机应用研究，2009,(11).

[48] 何杰光，傅秀芬．基于面向对象特性和 XML 的 RBAC 模型[J]．计算机应用，2009,(01).

[49] Wang, J, Rosca, D, Tepfenhart, W, Milewski, A, Stoute, M. Dynamic Workflow Modeling and Analysis in Incident Command Systems .IEEE transactions on systems, man, and cybernetics. Part A, Systems and humans, 2008,(5).

[50] 徐斌，袁健．基于 Web2.0 的用户权限管理研究与实现[J]．计算机工程, 2008,(13).

[51] 管昌生，蔡瑾．基于角色的动态工作流技术的研究[J]．计算机与数字工程，2009,(01).

[52] 张健，孙吉贵，李妮娅，胡成全，杨滨．工作流系统中一个基于多权角色和规则的条件化 RBAC 安全访问控制模型[J]．通信学报, 2008,(02).

[53] 吕宜洪, 宋瀚涛, 龚圆明. 大型应用系统用户权限构成分析及访问控制策略研究[J]. 小型微型计算机系统, 2007,(02).
[54] 倪晚成, 刘连臣, 吴澄. Web 服务组合方法综述[J]. 计算机工程, 2008,(04).
[55] 张原, 高向阳. 基于 XML 数据库的工作流系统研究[J]. 微电子学与计算机, 2008,(12).
[56] 万灿军, 李长云. 基于动态 AOP 的构件交互行为监测器[J]. 计算机应用, 2011,(02).

发表文章目录

1. Jian-Min, Zhao, Xiao-Chun, Long. A Modified Model for Flexible Workflow Access Control. 2011 年第四届计算智能与设计国际会议 （ISCID 2011），2011 年 10 月，中国，杭州.（EI 光盘版检索号：20115114606257）

Abstract: Draw on traditional access control technologies, this paper analyzes the advantages of Role Based Access Control Technology (RBAC) in realizing procedure permission dynamic management. Improve the problems of flexibility which exist in workflow access control, from the view of users, under the constraints such as the least privileges, separation of duties and load balancing of users, this paper constructs user security control model to realize the flexible authorization relations of Users- Roles-Permission. It also constructs the flexible workflow model of Role Based Access Control. In addition, it applies the model to construct a real work flow, in order to solve the problems of the flexible operation of resource and dynamic assigning tasks in the business process, and applies the engine into the specific business process.

Key words: Workflow, RBAC, flexibility, permission

2. Jian-Min, Zhao, Xiao-Chun, Long. A Flexible Workflow Model of Role-based Access Control. 2011 年国际工程与技术大会（CET 2011），2011 年 10 月，中国，上海.

Abstract: For the shortcomings of the traditional workflow in flexibility, this paper analyzes the advantages of Role-based Access Control Technology (RBAC) in realizing procedure permission dynamic management. With RBAC applied in it, this paper also constructs the flexible workflow model of Role Based Access Control and gives out the definition of the engine as well as the components concerned. In addition, it also solves the problems of the flexible operation of resource and dynamic assigning tasks in the business process, and applies the engine into the specific business process.

Key words: flexibility, RBAC, Workflow, Users- Roles-Permission

致　谢

在这里，我要对在硕士研究生期间教导我的师长、鼓励我的同学、支持我的家人以及所有给过我正面能量的人表达我最崇高的敬意和最衷心的感谢。

首先，我要感谢我的导师赵建民老师。近三年的师生相处中，赵老师在学习和生活方面都给了我极大的帮助。赵老师乐观积极的生活态度，友爱宽容的待人风格，严谨负责的工作作风，活跃开阔的思维方式都对我产生了极其深远的影响。谢谢老师的真诚友善，才让我们这个大家庭时刻都充满了欢笑和温暖；谢谢老师的包容大度，才让我们每个人都更懂得珍惜这份可遇不可求的缘分；谢谢老师的信任和鼓励，才让我们能更加昂首挺胸的踏上未来的征程。赵老师，我的良师益友，学生终生感激。

其次，我要特别感谢在研究生阶段给了我无私帮助和细心关怀的袁文翠老师。袁老师学术研究认真负责，在工作中耐心为我答疑解惑，使我在学术研究和技术能力上都获得了很大进步。在此，向这位最可爱最可敬的老师表达我深深的谢意和诚挚的祝福，祝愿老师健康、幸福。

再次，我要感谢我的朋友们。实验室的兄弟姐妹，欢声笑语，其乐融融的一教 D 座 104。更新换代，离别的聚餐除了沉甸甸的祝福外，觥筹交错间我们总会泛起不舍的泪花。寝室的好姐妹，谢谢你们在我最需要帮助的时候一直陪伴着我。我的朋友，不管是身边的还是远方的，感谢你们一直以来的关心，不厌其烦的为我出谋划策、排忧解难。

最后，我要感谢我亲爱的家人们。这么多年来，他们对我的支持和关爱，默默做出奉献和不求回报的牺牲，将成为我勇往直前的不竭动力。

衷心感谢为评阅本文而付出辛勤劳动的专家和老师们！

东北石油大学

硕士研究生学位论文摘要

论文题目：基于 RBAC 的柔性工作流研究与应用

硕 士 生：龙晓春

指导教师：赵建民　副教授

学科专业：计算机应用技术

2012 年 3 月 25 日

基于 RBAC 的柔性工作流研究与应用

摘　要

计算机的高普及与互联网技术的飞速发展，促使信息资源在现代企业中呈现出异构分布、松散耦合的特点，传统的集中式信息处理方式已满足不了需求，保证相关任务的高效运转并接受严密的实时监控是一个必然趋势。计算机支持的为实现业务流程自动化的工作流技术日益受到学术界与企业界的重视。由于复杂多变的客观环境，人们自身经验、知识、技能、需求等各方面的不断发展以及企业本身在发展过程中组织机构、政策文化、企业定位等变动，业务工作受到来自外部和内部两方面因素的影响，因此就要求工作流技术也应该具有根据实际情况进行调整的能力，即具有柔性。传统工作流系统大多采用静态、固化的表现形式，在流程定制开发、流程建模以及实例运行等多个环节灵活性、适应性不够。柔性工作流是顺应发展需要而产生的，能够更加确切的描述企业业务过程，满足企业工作的实际需要。柔性已成为衡量工作流系统性能的重要指标之一，吸引了越来越多的专家学者、研究机构与企业组织在展开深入研究。

目前，较为广泛的对流程柔性的研究基本上仅从活动入手，提高流程灵活性也是从活动及活动间的关系来考虑的，活动的柔性主要体现在流程节点和路径方面的灵活性。活动是流程的一个重要部分，与此同时，我们应该重视活动的执行人。从人和活动的关系来看，人是活动的执行主体，流程中遇到的异常很多可归因于角色在过程中的交互与变化情况，用户动态授权在实际业务过程中是一个非常重要的问题，其灵活程度是工作流系统柔性的一个重要反映。在受到内外部各种影响因素作用时，柔性工作流系统在能够保证对用户完成实际业务工作支持的同时，需要尽可能提高灵活性。系统需要做出适当调整，首先需要在变化中找出一部分不变的内容，并尽力保持不可变性，作为系统进行动态适应性调整的“支点”，以便用户更好地应对现实工作中可能遇到的各种变化、异常和不确定的问题。工作流系统的柔性主要体现在流程结构的动态调整、流程实例执行路径的柔性选择以及流程实例活动执行者的柔性分配等具体方面。

在计算机通信中网络安全的防范与保护极其重要，访问控制（Access Control，AC）是一项主要的安全保护策略。访问控制的基本策略是：基于身份认证原理，系统中被授权的合法用户依据权限约束可以对系统中可获取的客体资源提出访问请求并进行相应控制操作。通过这一机制，实现合法主体对其授权客体的合法操作，同时防止合法主体任意访问权限之外的资源或者非法主体入侵系统造成破坏。访问控制模型中的三个主要元素：主体、客体、主体-客体的访问控制约束。几种较为常见的访问控制技术：自主型访问控制（DAC）、强制型访问控制（MAC）和基于角色的访问控制（RBAC）。相对前面两种访问控制技术存在的不足，RBAC 技术的在系统访问控制、用户授权方面有一定的优势，成为目前大型企业中应用最好的访问控制机制。RBAC 在主体与客体之间引入角色这一中间机制，主体与角色相对应，而对客体的操作权限则与角色相关联，主

客体不再像以往那样直接关联，从而大大降低了系统中两大主要实体的耦合性，提高了系统访问控制机制的安全性、灵活性和适应性。

本文详细地分析了工作流参考模型、工作流管理系统体系结构，对 RBAC 技术进行了全面地研究，重点分析了工作流系统中用户-角色-权限三者的关系，构建出用户权限管理模型，并将其应用于工作流系统用户授权。从流程过程柔性管理与用户柔性授权管理两点出发，开发了基于 RBAC 的柔性工作流系统，在实际系统中得到了较好的检验与应用。

1. 基于 RBAC 的柔性工作流模型。首先，分析了 RBAC 技术在用户权限管理方面优势，以工作流系统中用户授权为切入点，根据工作流系统的特性，构建出基于 RBAC 的柔性工作流模型。具体分析了如何实现用户-角色-权限的授权关系，依据最小权限原则、最小权限时限原则、职责分离原则、用户负载均衡等约束条件使合法用户在恰当的时间拥有合法的角色实现对流程客体资源的灵活操作。提出了推式分配策略与拉式分配策略，实现系统用户权限的柔性管理。其次，在工作流相关理论研究的基础上，提出了流程过程柔性管理的具体实现方法。以节点与路径为出发点，研究工作流过程的柔性。将系统的主要对象进行有效划分，建立起类似数据库映射机制的基空间和元空间，系统依据元对象协议把对元对象的各种处理映射到基空间中的具体过程实例上去，实现了流程柔性定制与流转方向的灵活控制。以流程实例具体情况为依据，保证流程实例顺利运行且实现指定目标。

2. 完成了基于 RBAC 的柔性工作流系统的开发，将其应用于大庆油田试油试采分公司综合信息管理系统中，实现了业务流程的创建、实例化、跳转及追踪监控等功能，提高了系统在业务流程过程管理和用户权限管理上的柔性。

一、基于RBAC的柔性工作流模型研究

工作流系统一般包括流程定义、流程建模、流程执行三个主要步骤，其中，流程定义决定了模型的构成从而最终决定流程实例的运行情况。然而现代企业业务需求的多变性使得一次给出完整流程定义的方式显得不切实际，在遇到一些定义时未预计到的异常情况，或者出现与预计结果不相符合的状况时，原有的流程系统就无法继续顺利运行下去。因此，需要一个更稳健的工作流系统，它能够很好地处理各种不确定因素，灵活应对异常状况对系统造成的冲击，提高工作流应对不确定因素的能力，即提高系统的柔性。对于如何提高系统的柔性，目前，工作流领域的专家学者以及相关企业已从多角度入手展开研究，也取得了一定的成果。本文主要从两个方面着手，分析并实现系统柔性的提高，一是系统流程过程的柔性，二是组织中用户权限管理的柔性。同时，在对工作流模型、RBAC 技术研究的基础上，提出了一个更灵活、更安全的柔性工作流模型。

1. 模型的提出及元素的形式化定义

引入 RBAC 技术，参考了其用户权限管理方面的优点，结合工作流系统的实际组织结构，通过构建一个独立的用户权限管理模型，专门用于管理业务流程中的用户、角色、

权限的关系。

在工作流执行服务器与客户应用程序两部分之间加入此用户权限管理模型，使二者之间存在着逻辑上的调用关系，而具体的实现过程是相对透明的。这个过程分为两个步骤：第一步是工作流执行服务器在发出任务时与用户权限管理模型建立连接，二者实现从任务到权限的映射，明确任务执行时需要作用的客体资源与进行的实际操作；第二步发生在用户权限管理模型与工作流客户服务程序之间，即实现任务的最终执行人与模型定义中的角色以及用户的对应，从而实现工作流任务与任务执行者之间合理灵活分配。

将工作流信息系统中所涉及的对象进行有效划分，建立起类似数据库映射机制的基空间和元空间，其中，基空间为所构建的工作流系统，而与之对应的元空间是由抽象出来的各种元对象以及对其进行管理的元对象协议（Meta Object Protocol, MOP）构成。

定义了模型主要元素即元素间存在主要关系，包括：角色：$R=\{R1, R2,.....Rn\}$；用户：$U=\{U1, U2,.....Un\}$；任务：$T=\{T1,T2.,....Tn\}$；任务执行人：$E=\{(u\cup r)\mid u\in U, r\in R\}$；权限：$P=\{P1,P2,.....Pn \mid Pi=\{(REi,OPi)\mid i=1,2,.....n\}\}$，其中又包含了资源与操作两个部分，资源：$RE=\{RE1, RE2,.....REn\}$；操作：$OP=\{OP1, OP2,.....OPn\}$。角色与权限的分配：$ROLES(P)=\{p\in P\cap(p,r)\in PR\}$；角色与用户的分配：$USERS(R)=\{r\in R\cap(r,u)\in RU\}$。结构元对象 Stru_MetaObject；行为元对象 Fun_MetaObject；状态元对象 Sta_MetaObject。

工作流系统每次启动一个流程实例的同时，初始化与其对应的一系列元对象。这些元对象对应于流程实例的状态、功能、运行行为等，并随着实例的运行而随之改变，能实时反应流程实例的各方面情况。工作流系统以元对象协议为依据，将对元对象的各种操作映射到与之对应的基空间流程实例上。因此，能通过此规范化协议的方式来实现对工作流系统的各种调整的控制，既能实现对原有流程的干预，又能较好的控制在允许的范围之内对系统进行修改，避免发生给系统带来破坏性的操作，保证了系统在流程运行过程阶段具备较大的自由性。

2. 流程过程管理的柔性

工作流流程过程主要由节点与路径两个基本部分组成，节点表示工作任务需要执行的各个环节，而路径则表示由各个具有先后流转次序的节点构成的一条执行路线。本文主要从这两个方面来讨论流程过程的柔性,并以此为基础,完成了几种典型的流程模式的设计与实现。

(1) 流程节点的柔性

节点（Nodes 简记为 N）是工作流一个最基本的组成元素之一，在具体流程中体现为各个定义完整的环节（包括环节的任务、任务的执行人、任务的时限等）。节点的的

柔性主要是指在实际业务过程中通常会发生节点的新增与删除等情况，在流程建模阶段无法完成准确的预计，即是否设置一些不必要的多余环节，或者是否还存在一些本身未被包含进来的关键环节。在新增或删除节点的同时，节点的偏序集随之改变。

(2) 流程路径的柔性

在工作流系统中与节点相辅相成的另一最基本元素是路径（Paths 简记为 P），路径表现为节点间的流转关系，相对于节点而言，路径的动态性更为普遍和复杂。在流程定义时预先设定了流转路径，但在流程实际执行过程中，某些条件、参数以及客观需求等的变动都会引起流程最终走向的改变。因此，本文从三个角度来分析流程路径的柔性即：新增流程路径、撤销流程路径以及改变原有流程路径的方向。同时，流程路径的偏序集会随着路径的变化而改变。

(3) 几种典型流程模式的设计与实现

a) 顺序执行模式：工作流引擎采用“同步器”节点来表示模型中的计算逻辑。同步器“S”是一个一般意义上的同步器，同步器前后与相关实体相接，开始节点、结束节点是特殊的同步器。

b) 自由流模式：“自由流”是一种符合中国特色的业务流程模式，即在某些特殊情况下，流程不按照既定顺序执行，而是在环节间自由跳转。在工作流引擎中，用 jumpTo 方法完成节点本工单，并跳转到指定的活动。

c) 回退模式：实际工作中，存在审批环节不通过的情况，此时需要将工作返回给上一环节执行人，同时在审核意见中给出评价，提出修改意见。在设计过程中，回退模式通过调用 jumpTo 方法，根据执行者的决定回退到前面任一环节，实现流程灵活运行。回退模型实现的原理与跳转类似，即回退是一种特殊的跳转，可跳转到之前的任一执行环节。同时在用户分配上也具有特殊性，即有针对性地将任务分配给返回之前签收并完成任务的所有执行人，而其他同级用户不会被分配该任务。

3. 用户权限管理的柔性

用户权限管理是工作流系统中一个重要的部分，组织机构与人员变动相对于系统中其他因素具有更复杂的不确定性，因此，提高系统的适应性需要非常重视用户权限管理方面的柔性。在用户权限管理的设计上，考虑到用户任务分配具有主动获取和被动分配的特点，可以将推拉式理论引入到本工作流模型中，将之命名为推式策略与拉式策略。这两种用户权限管理策略各有优缺点，本文考虑到所设计的模型要具有柔性，所以将两者混合用于模型中。

(1) 推式策略

推式策略是由工作流执行服务器将任务分配给所有合法用户，即用户登录后其相应待办工作列表中都会出现该新增任务。推式策略适用于岗位相对稳定，任务完成与角色的对应关系确定性较大这样的工作流系统中。这种职位明确，职责稳定的工作模式，采用推式策略有几个好处：一是推式策略步骤简洁、容易实现；二是系统分工明确，模型各部分间的耦合性低。

(2) 拉式策略

拉式策略是指任务产生后，工作流执行服务器首先不将任务分配给某个用户，而是将任务公布于系统中的一个共享工作列表中，合法用户可以申请该任务。在一定时间后，初步采用按用户的申请时间早晚来体现用户的工作积极性、用户的待办工作任务量来衡量用户的工作负荷、用户的已办工作来衡量用户的工作能力以及流程任务分配者根据对这些用户一个经验判断等因素作为考量的依据，最终得出这些用户的综合测评分，从而选取出最合适的用户（一个或者多个），完成任务的分配。拉式策略实现更为复杂，适用于任务分配过程中更为灵活的复杂工作流系统，选定一个最适合的任务执行人使得任务更高效的完成。

(3) 几种重要的授权约束

a) 最小权限约束：用户与角色、角色与权限多对多的关系，决定用户可以被赋予多种角色，进而拥有所赋予的角色相应的多个权限。在实际流程任务分配时，用户拥有完成任务所需要的最小权限 least_Permission(u)，即对客体资源实施最小操作去完成任务。

b) 最小权限时限约束：流程开始运行时 Ts，工作流机对合法用户进行授权，而在用户在完成并提交任务时 Tf 或者达到任务完成的最大时限 Tm，工作流机应该及时将权限收回 Te。

c) 职责分离约束：流程中的某些关键任务，一般不会分配给同一用户，而应该安排不同的执行者完成不同的任务，实现权力分散。在引入用户权限管理模型后，本文从两个方面讨论职责分离约束：角色互斥约束与权限互斥约束。角色与角色的互斥，是指用户可以同时被分配角色互斥组（ROLESC）的多个角色，但不能在同一个流程中同时会话、激活互斥组中任何两个以上的角色。权限与权限的互斥，是指角色拥有权限互斥组（PERMISSIONSC）中一个权限后就不能再拥有组内其他权限。

d) 用户负载均衡约束：用户建立会话激活角色的数量需要受到限制，最大值不能超过 USER(R)中的合法角色总数。在实际流程运行时，负载均衡约束要求衡量用户的实际工作属性（包括用户的工作积极性、工作能力、现有工作任务量），将任务分配给可能最高效率完成任务的用户（best_User）。

(4) 用户权限管理的具体实现主要分为以下几个主要部分：权限定义，角色定义，权限配置，权限审查。

明确定义权限是系统权限管理的首要步骤，权限通常以一个三元组 P（o，t，p）来描述，其中：o（object）为访问客体；t（type）为访问类型；p（predicate）为谓词。表示在谓词 p 为真时对于对象 o 可进行 t 类型的访问。本系统为用户提供了定义角色的工具，用户可以根据组织结构、自身的权位、职能等来实现角色的定义。权限的配置是系统授权管理的重要工作，实际用户只有拥有一定角色从而获取相应权限才能进行访问操作。与组织结构相对应建立的角色通常都对应着这一职能岗位上的一组访问权限，系统中的合法用户可以扮演多重角色，同样，一个系统权限也能被配置给多个角色。权限的审查也是用户权限管理主要工作步骤之一，完成权限分配后，系统需要对已授权用户登

录系统后的能力表进行检查，确认此权限分配是否合理若不合理则需要考虑收回这些不合理部分的权限并对其进行重新配置。

二、基于 RBAC 的柔性工作流系统应用

试油试采分公司综合信息管理系统的是以试油、试采专业动、静态数据库为研究基础，建立一个统一的试油、试采信息综合应用环境，为生产管理人员提供及时应用试油、试采动静态资料进行综合分析判断的技术手段，为及时准确做出决策提供依据，同时为研究人员提供一个试油、试采信息综合应用平台。本系统采用多级模式进行架构，以试油试采分公司业务流程为主线，囊括了业务工作的全过程，借助现有的网络环境和协同思想，实现信息、业务流程、人员的有机结合，使用户轻松构建起一个数据共享、流程同步的综合性信息化工作平台，主要从流程过程管理与用户权限管理两个方面提高系统柔性。

1. 工作流过程管理模块的设计与实现

(1) 模块的主要组成机制：流程定义管理机制、流程模型设计器机制、流程实例化机制、流程监控机制。本系统提供了对 XML 格式的流程定义文件的增删改查，实现对流程定义文件的灵活有效管理。流程设计器是一个所见即所得的可视化界面，简单易用。流程实例化是过程管理模块中最重要的部分，柔性工作流系统实现灵活的流程定制，生成各不一样的流程实例，能更好的满足实际需求。流程的监控是符合实际工作需要的一项重要功能机制，随着流程执行情况的发展而变动。

(2) 系统流程的分析与模型构建：依据本文上一部分分析给出的试油试采分生产管理流程泳道图和工作流程图，构建出了实际业务过程十七个基本的流程模型，本系统采用 Eclipse3.2 建模工具建构模型，根据各模型不同的需求设计其约束条件，并通过模拟器进行测试，保证流程顺畅运行，满足设计目标。

(3) 系统实例中柔性的具体体现：

a) 柔性的流程定制

流程的定制是指以流程模型为基本参照，一个流程模型通常要求尽可能全的包含业务过程的所有环节。而在实际工作中，启动者依据业务需求，决定该流程实例应选取执行的流程步骤，这些环节可以不必连续，但要求至少为一个、至多为模型总环节数。这样每次启动的流程实例都不尽相同，但都是流程模型完整实例的一个子集，灵活的流程定制是本系统柔性的一个体现，不需要的流程步骤可以被略过，简化了工作的复杂性，提高了工作流系统对实际业务需求的支持。

b) 流程流转方向的柔性控制

流程实例启动后，在其运行过程中存在着一些不确定的因素，容易引起业务过程的变动，从而会对正在执行的工作流程有动态调整的需求。本系统在每个流程执行人完成流程基本任务（如签收、审核、提出意见等操作）之后，还设置了一个这样的接口，即选择流程下一个执行环节。默认值为不设置，则流程按照定制时的活动先后关系流转到

下一环节；若此时系统有动态调整的需求（如略过某一些环节，或是新增某些执行环节，以及可以跳回到之前的环节）则用户可以根据需要灵活的设置下一个执行环节，从而能改变原有流程实例的流转路径。

(4) 流程监控的实现

在流程动态管理方面，还有一个重要的模块就是流程的监控。必需要对流程实例的运行情况进行监控、分析，才能知道流程任务是否按要求被合法用户完成，从而保证业务目标的顺利实现。新的流程实例启动后，系统提供了流程进度查询功能，实现对流程进度的实时监控。在本系统中监控主要分为分两个粒度：一是以井为单位进行的监控，监控与某一口井相关的所有流程的执行情况；二是以流程为单位的监控，这是更细一级别的监控，能及时的反应出具体的某一流程实例执行的具体情况。

(5) 与流程相关的主要数据对象与相关操作：主要数据对象包括流程定义、流程实例、流程任务实例、流程工单、流程的节点、流程的边（路径）。主要操作包括对这些数据的增删改查等基本操作，流程实例、任务实例、工单等运行相关的一些操作（如创建、挂起、终止、恢复等），流程柔性执行的一些基本操作（如完成工单并指定跳转环节、完成并指定跳转环节及跳转人等）。

2. 用户权限管理模块的设计与实现

在本系统中采用以角色为基础的访问控制模型，设计与实现了用户权限管理模块。对权限的授予具体可分为三类：一是对职员的岗位授权，指定岗位所具有的权限，然后指定职员所属岗位。在本系统中，根据各部门及岗位设置了相应的角色实现授权。二是对职员直接授权，该授权是根据职员本身的属性或状态授权，在系统中这一类授权在采用推式策略时体现。三是代理权限，即拥有权限的角色或用户能将自己的操作权限委派给另一角色或者用户去完成相应操作，在本系统中此类授权方法已实现，但暂未被使用。

用户权限的管理即用户拥有一定的角色去完成相应的任务，是伴随着流程过程执行而发生的。通过第四章的分析与设计，在本系统中对用户权限的柔性管理在流程定制与流程路径更改都提供了动态指派用户的功能。在流程定制时，发起人能对定制好的执行环节对应的合法执行人进行选择，而其它未被定制的环节对应的执行人选项是灰色即处于不可用状态。在流程运行过程中，与制定下一环节相对应，本系统提供了指定环节执行人的功能，合法用户列表与选择的环节保持一致。

本系统综合采用了推式（PUSH）与拉式（PULL）两种策略，以流程模型的复杂性以及实际业务过程的需求为依据，很好的实现了用户权限的柔性分配。策略选取的影响因子很多，本系统主要以流程实例的环节总数和流程实例某环节的合法用户数为依据，分别对二者给定对应参考标识，用以决策如何恰当的选用分配策略。

(1) 推式策略的实现：当流程实例环节不多，各环节的执行人较少且通常为固定的几个人时，就采用 PUSH 策略，这样能减少引擎的工作负担和系统的出错率，从而提高工作效率。其主要步骤包括：引擎分配任务、用户签收任务、用户完成任务。

(2) 拉式策略的实现：面对流程过程较复杂，执行人变动性较大的流程实例，则采

用 PULL 策略，更好的保证用户权限安全灵活的满足系统的实际需求。本系统中拉式策略的实现步骤有：引擎产生任务分配至共享工作列表、合法用户申请任务、引擎通过评分决策任务的分配、申请成功的用户签收并完成任务。

(3) 与用户权限管理相关的主要数据对象与相关操作：主要数据对象包括：角色、用户、权限、角色权限、用户激活角色等。主要操作包括：用户、角色、权限、部门等都有基本的增删改操作；一些其他的重要操作（获取角色的全部用户、用户的全部角色、角色的相应权限、角色的基本分配等）。

三、总结

本文从流程过程管理与用户权限管理两个方面出发，探讨如何实现工作流系统的柔性。以用户授权为结合点，将 RBAC 技术引入到工作流管理系统中，构建了基于 RBAC 的柔性工作流模型，提高了系统访问控制方面的灵活性与安全可靠性。分别从节点与路径考虑，实现了柔性的流程定制与流程路径的灵活选择。同时，完成了基于 RBAC 的柔性工作流系统的开发，并在试油试采综合信息系统应用中验证了其可行性、有效性和实用性。

艰苦创业　严谨治学

严谨朴实　勤奋创新

招生办：0459-6503721

培养办：0459-6504792

学位办：0459-6503938

学校网址：http://www.nepu.edu.cn