

论文题目：基于工作流的油气信息协同管理与集成研究

专    业：计算机应用技术

硕 士 生：王  锐 （签名） 王  锐

指导教师：方明（签名） 方  明

## 摘    要

本文探讨了目前油气信息系统的开发模式和存在的问题，并简要介绍了工作流的技术特点，同时分析了工作流在油气信息管理系统中的研究和应用。在深入调研石油行业具体问题的基础上，根据工作流技术理论设计了一个基于组件技术的轻量级工作流引擎，并以轻量级工作流引擎为核心设计了油气信息管理系统的整体架构。在此架构之上实现了基于工作流技术的高自动化、高扩展性且具有强大的自适应能力油气信息协同管理与集成系统。该系统满足了企业的日常业务自动化运转的需求，促进了部门间的协作，提高了业务运转的效率。

该系统充分利用 J2EE 体系架构的优势，采用 JAVA、JSP、JPA 等技术，构建了一个多层体系结构的业务系统，各层之间相互独立，实现了业务逻辑和业务数据的分离，提高了系统的稳定性、可扩展性。系统在设计和实现过程中，充分运用了设计模式来优化本系统的结构、提高了系统的可扩展性、可重用性和易维护性。

关键词：工作流；信息协同；集成；油气勘探开发

论文类型：应用研究

**Subject:** Based on the work flow of oil and gas information coordination management and Integration

**Speciality:** Computer Application Technology

**Name:** Wang Kun (signature) Wang Kun

**Instructor:** Fang Ming (signature) Fang Ming

### ABSTRACT

This paper explores the current development model of oil and gas information system and the problems, makes a brief introduction on the technical characteristics of the workflow, at the same time analysis the workflow, which has been studied and applied in the oil and gas information management system. After I make a in-depth research on the specific issues of the oil industry, a component-based technology lightweight workflow engine based on the workflow theory was Designed. take lightweight workflow engine as core, the paper designed the overall structure of the oil and gas information management system. oil and gas information management and coordination integration system based on workflow technology was realized, which has a lot of advantages, such as high automation, high scalability and a strong adaptive capacity. The system met the company's day-to-day operation of the business automation needs, promote the collaboration among departments, and improve the efficiency of business operations.

The system fully utilize the advantages of J2EE architecture, with JAVA, JSP, JPA and other technologies, to construct a multi-tier architecture system, all levels in the system is independent, it achieved the separation between business logic and business data, and improved the stability and scalability of the system. In the design and implementation process, the system fully used design patterns to optimize the structure and improve the scalability, reusability, and maintain.

**Keywords:** Workflow; Information Coordination; Integration; Petroleum Exploration and Development

**Thesis Type:** Application Study

## 学位论文创新性声明

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果；也不包含为获得西安石油大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中做了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

论文作者签名： 王 磊

日期： 2008.5.20

## 学位论文使用授权的说明

本人完全了解西安石油大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属西安石油大学。学校享有以任何方法发表、复制、公开阅览、借阅以及申请专利等权利，同时授权中国科学技术信息研究所将本论文收录到《中国学位论文全文数据库》并通过网络向社会公众提供信息服务。本人离校后发表或使用学位论文或与该论文直接相关的学术论文或成果时，署名单位仍然为西安石油大学。

论文作者签名： 王 磊

日期： 2008.5.20

导 师 签 名： 刘 明

日期： 2008.5.20

• 注：如本论文涉密，请在使用授权的说明中指出（含解密年限等）。

## 第一章 绪论

### 1.1 研究的课题背景

近几年来, 计算机网络的飞速发展提供了功能强大的信息处理平台, 同时石油行业的快速发展使得信息迅速膨胀, 这要求石油行业急需一整套从信息采集、信息处理, 到信息传递与共享高度自动管理信息系统。而以分布式为核心当代信息技术特别是 Web 技术的迅速发展, 又使得传统的石油管理信息系统的体系结构、计算模式等必须与 Internet 融合。

管理信息系统<sup>[1]</sup>与 20 世纪 50 年代提出, 在 20 年代后得到快速发展。其主要活动涉及: 信息处理、数据保存等若干计算机技术, 是最广泛使用的计算机综合技术。国内石油行业较早就使用了管理信息系统, 但目前油气信息系统仍然以传统信息管理系统为主, 基本上集中于数据处理, 以及常规业务处理, 而石油行业的现状则使得当前石油行业软件不能完全满足企业的实际需求。

过去石油行业开发了许多信息系统, 这些系统大都是传统的管理信息系统, 由于当时网络技术和信息管理软件本身的局限, 这些系统存在许多缺陷难以适应现代工作的要求。如集成化程度低、资源共享不充分、缺乏韧性、对环境适应性差。特别是这些系统大多数时事务处理型系统, 仅能提供信息的存储、管理、查询、远远不能适应网络化信息时代的要求, 使其应用范围和效果很有限。当前要求石油行业需要的信息系统应该能够显著提高业务处理效率, 降低业务成本, 充分利用内部资源, 加快工作流程, 适应环境的变化等。要实现高度集成化的、灵活的、人机界面交互简单友好的、动态的网络管理信息系统。

就当前而言, 油气信息系统一般采用传统技术, 集中于油气信息数据的采集, 处理以及业务处理, 基本上是一个事务性管理信息系统。但是由于石油行业的特殊性, 其上级和下级单位众多, 且各自又有相当多的业务系统, 这些业务系统保存有大量的油气管理信息系统需要的重要数据, 形成了信息孤岛, 这使得传统以事务为主的石油信息系统, 难以整合这些需要共享的数据, 当前迫切需要集成现有的有业务系统, 已实现应用到应用的数据移动。此外传统的油气信息系统将业务流程固化在系统中, 不能适应企业的流程变化, 这使得企业业务流程的重组变得困难重重, 所有这一切都使得现有的油气信息管理系统的开发模式变得越来越不适合飞速发展的石油行业现状。

当前的石油行业管理软件面临着两个主要困难: 1) 如何保证系统满足复杂业务流程、满足业务流程将来的扩充; 2) 如何与众多现存的业务系统集成。就石油行业来说, 一方面石油行业中的许多业务过程比较复杂, 且这业务活动的发生及有顺序关系, 也有并行关系, 相互间的数据依赖关系也较为复杂, 这需要我们保证数据能够方便的在不同的业务中自动的流动, 从而实现业务自动化。另一方面, 石油行业现有的企业应用系统如

ERP、CRM、SCM 某种程度上都还是一种自动化孤岛，它们被设计用来提供对特定领域的最优解决方案，而随着企业业务的发展对这些自动化孤岛提出了集成的要求。基于工作流<sup>[2]</sup>的油气管管理信息系统正是上述需求的结果，它主要聚焦于业务过程之间的数据移动以及与现有的系统进行整合，油气信息管理系统不再是简单的处理数据，而是实现业务自动化和如何有效的与各种业务系统进行整合。石油行业迫切需要一个具有高度可重构性、可重用性、可扩展性，支持信息协同管理和集成的应用系统，以满足石油行业以较小成本、较小风险获取可满足需求的系统。

所有这些困难都意味着必须抛弃传统的石油信息系统开发方式，从根本上改变以事务处理为主的石油信息系统，进而过渡到以满足复杂业务流程和与现有业务系统集成上来，真正实现满足企业飞速发展的业务需求上来。这使得在石油行业信息系统中引入工作流管理系统成了一种有效地实现了石油信息系统的飞跃得手段。

## 1.2 课题目标，研究内容及技术思路

本课题的最终目标是：为油气信息系统提供一个具有高度可重构性、可重用性和可扩展性，支持信息协同管理和集成的分布式油气信息系统，满足石油行业以较小成本、较小风险获取可满足需求系的统。

基于以上目的，本课题研究工作主要包括以下几点：

1) 分析工作流技术与油气信息系统的关系，研究工作流技术在油气信息系统的应用及工作流技术运用于油气信息系统中的优势。

2) 对工作流技术进行深入的探讨和研究，并在此基础上结合油气信息业务特点设计并实现了一个基于组件的轻量级工作流引擎。

3) 设计一种基于工作流技术的油气信息系统的体系结构。

4) 基于油气信息系统的体系结构，结合现代企业单位在信息化过程中所遇到的具体问题和需求，设计并实现了的油气信息系统。

为了实现油气信息协同管理与集成，本文采用以下技术：

1) 利用面向对象的系统设计技术，设计结构清晰、复用性强的类结构和各种接口。

2) 采用 J2EE 平台提供的组件封装系统功能。利用 Java 组件实现基础功能组件和核心功组件，利用 Java 组件来实现应用功能组件。

3) 基于 Java 技术实现了一个轻量级工作流引擎。

4) 以 J2EE 架构和工作流技术为核心设计油气信息系统的体系结构。

5) 采用 Sqlserver 集成服务和工作流技术实现企业数据集成与业务集成。

### 1.3 论文结构

第一章：绪论。第一章要提出问题，说明目前石油信息系统开发模式和现状，存在什么问题，由此引出需要借用 workflow 技术进行软件系统的开发，强调 workflow 技术引入的重要性，同时给出即将展开的研究目标、研究内容及所要用到的技术。

第二章：workflow 技术和油气信息系统研究。本章讨论了 workflow 的相关概念，参考模型以及 workflow 管理系统的相关知识，然后根据现有 workflow 产品分析了 workflow 技术在信息系统开发中的应用现状以及在石油信息管理系统中应用现状，存在的问题。

第三章：基于组件地轻量级 workflow 引擎。以 workflow 理论为基础，结合油气信息系统特点，设计并实现了一个基于组件地轻量级 workflow 引擎。

第四章：基于 workflow 的油气信息系统总体设计。本章以 workflow 为核心进行了油气信息系统的系统分析，并给出了系统体系结构。

第五章：油气田开发系统详细设计与开发。本章以油气开发中一个业务子模块进行分析，并给出了其设计和实现系统实现中的一些技术问题。

第六章：结论与展望。本章对全文的研究工作进行了总结，对 workflow 技术发展及其在油气田开发中的应用作了展望。

## 第二章 工作流与油气信息系统

### 2.1 工作流技术简介

#### 2.1.1 工作流技术相关概念

工作流 (Workflow) <sup>[3,4,5,6]</sup> 的严格定义如下: 为了实现组织目标, 有关业务活动依时序或逻辑关系相互连接构成业务流程。在业务开展过程中, 文档、信息或任务, 依据组织规范在参与者之间传递、处理或执行。业务流程中, 实现了基于计算机自动化的全部或部分称为工作流。简单地讲, 工作流是指整个或部分经营过程在计算机支持下的全自动或半自动化。在实际情况中可以更广泛地把凡是由计算机软件系统 (工作流管理系统) 控制其执行的过程都称为工作流。一个工作流程包括一组活动及它们的相互顺序关系, 还包括过程及活动的启动和终止条件, 以及对每个活动的描述。

工作流是针对工作中具有固定程序的常规活动而提出的一个概念。通过将工作活动分解成定义良好的任务、角色、规则和过程来进行执行和监控, 达到提高生产组织水平和工作效率的目的。工作流技术为企业更好地实现经营目标提供了先进的手段。工作流管理系统 ( workflow management systems , WFMS ) <sup>[7]</sup> 是以规格化的流程描述作为输入的软件组件, 它维护流程的运行状态, 并在人和应用之间分派活动。在此, 我们先定义一些基本的术语: 流程定义 ( process definition ) 和流程实例 ( process instance )。一个流程定义是一个业务流程或过程的规格化描述。一个流程实例是流程定义的一个运行实体。工作流管理系统还处于技术发展曲线上的初级阶段。目前, 工作流中使用了过多的概念。在这个领域中的大量规范和工具没有一个是相似的, 他们之间主要的分歧在于如何阐述流程中的步骤。

在介绍工作流时有一个话题必须包括, 那就是工作流和业务流程管理 ( BPM ) 的关系。术语 “ 工作流 ” 通常描述人与计算机系统的一系列相关交互。在开发人员中, 工作流经常被提及。有时, 工作流的意思是指一些不同的 UI 界面。业务流程管理的范围比较广, 相比之下工作流多半局限于技术领域。业务流程管理还从管理人员的角度涉及了非技术问题, 比如分析、组织的效率。

工作流管理系统 ( Workflow Management System, WFMS ) 是定义、创建、执行工作流的系统。在最高层上, WFMS 应能提供以下三个方面的功能支持: 建造功能: 对工作流过程及其组成活动定义和建模; 运行控制功能: 在运行环境中管理工作流过程, 对工作流过程中的活动进行调度; 运行交互功能: 指在工作流运行中, WFMS 与用户 ( 业务工作的参与者或控制者 ) 及外部应用程序工具交互的功能。

根据 WfMC 的定义, 工作流管理系统 ( Workflow Management System, WFMS ) 通

过软件定义、创建工作流程并管理其执行。它运行在一个或多个 workflow 引擎上，这些引擎解释对过程的定义，与 workflow 的参与者（包括人或软件）相互作用，并根据需要调用其他的 IT 工具或应用。

workflow 管理系统是以规格化的流程描述作为输入的软件组件，它维护流程的运行状态，并在人和应用之间分派活动，推进 workflow 实例的执行，并监控 workflow 的运行状态。workflow 管理系统可以描述不同覆盖范围和不同时间跨度的经营过程，根据经营过程以及组成活动的复杂程度，workflow 管理系统可以采取多种实施方式，在不同实施方式中，所应用的信息技术、通信技术和支撑系统结构会有很大的差别，workflow 管理系统的实际运行环境也可以在一个工作组内部，也可以在全企业所有业务部门。

workflow 管理系统在实际系统中的应用一般分为三个阶段：即模型建立阶段、模型实例化阶段和模型执行阶段。在模型建立阶段，通过利用 workflow 建模工具，完成企业经营过程模型的建立，将企业的实际经营过程转化为计算机可处理的 workflow 模型。模型实例化阶段完成为每个过程设定运行所需的参数，并分配每个活动执行所需要的资源，模型执行阶段完成经营过程的执行，在这一过程中，重要的任务是完成人机交互和应用的执行。

下图为 workflow 的体系结构。

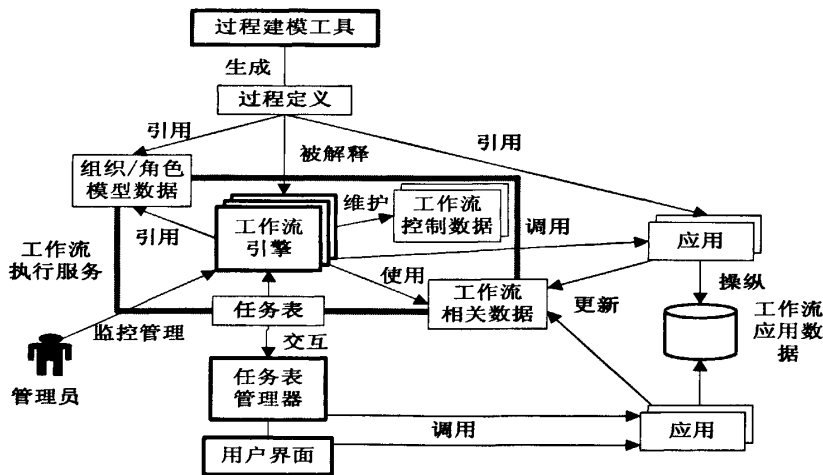


图 2-1：workflow 体系结构

### 2.1.2 workflow 管理系统参考模型.

workflow 参考模型<sup>[8]</sup>来源于对普通工作流程序结构的分析，确定结构中的接口，这些接口可以使不同产品在不同的结构层次上协同工作。所有 workflow 系统都包含一系列的公共组件，组件间采用一套被定义好的方法进行协作；不同的产品在这些公共的组件中，



会表现出不同的处理能力。为了实现不同 workflow 产品间的协同工作，需要在这些组件间制定一套标准的接口和数据交换格式。通过实现这些标准接口，可以达到产品间的协同工作。下图是 WFMC 定义的工作流参考模型，图中的 workflow 引擎为 workflow 实例提供运行时期执行环境的软件服务器或引擎。workflow 应用程序中的数据交换主要通过 3 个接口来处理：客户端应用程序接口、应用程序调用接口和 workflow 引擎协作接口。workflow 的过程定义与运行环境应该分离，过程定义工具与 workflow 引擎的交互接口被称为过程定义导 N 导出接口，它支持过程定义信息间的相互转换或已完成的过程定义间的相互转换。管理和监视工具可以提供 workflow 运行状态的完整视图，以及系统管理、安全和权限控制等功能。

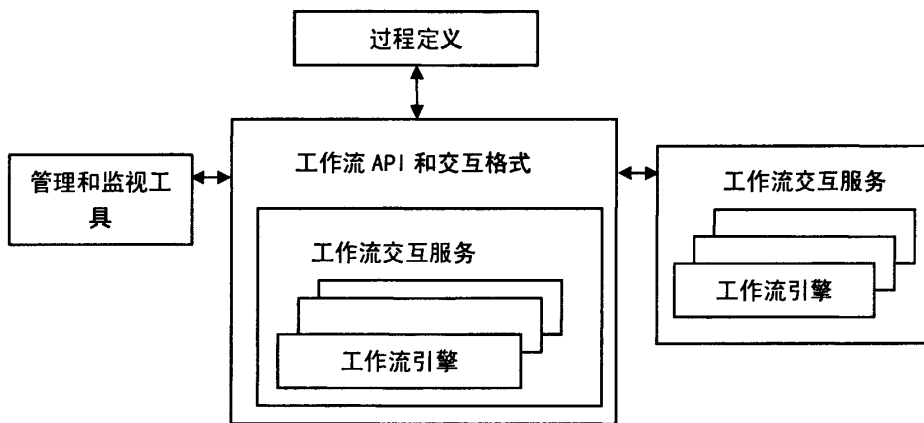


图 2-2 工作流参考模型

workflow 模型是对 workflow 的抽象表示，也就是对经营过程的抽象表示。包含了描述一个能够由 workflow 执行服务软件系统执行的过程所需要的所有信息。这些信息包括过程的开始和完成条件、构成过程的活动以及进行活动间导航的规则、用户所需要完成的任务、可能被调用的应用、workflow 机的引用关系、以及所有与 workflow 相关数据的定义。由于需要在计算机环境下运行，所以 workflow 模型不仅要让人读懂，更要让计算机理解所定义的 workflow 模型。因此，简单的业务过程通过语言或文字就可以完全表达，无需建立明确的模型，而描述企业经营过程的 workflow 必须建立相应的工作流模型，才能实现企业业务流程的 workflow 管理。尤其是实现复杂并行执行流程的业务过程，只有建立模型才能对流程执行情况进行有效的监控。所以 workflow 模型是一个概念模型，是对业务流程的一种抽象。

workflow 执行服务器使用一个或多个 workflow 机，为过程实例和活动提供运行环境，负责解释和激活过程定义，与过程所需的外部资源进行交互。在模型中，过程与活动控制逻辑间有一个逻辑上的分离，活动控制逻辑构成 workflow 执行服务器；过程与应用工具间、与终端用户任务间也有一个逻辑上的分离，应用工具和任务建立起对每个相关活动的处

理。这种逻辑上的分离，为制定更多的行业标准提供了机会，也为在工作流程序中集成用户具体的应用工具提供了机会。在分布式的工作流执行服务器中，每个工作流机控制过程执行的一部分，并与这部分过程中的活动所要用到的用户、应用工具进行交互。在分布式的执行服务器中有公共的名称空间与管理范围的，从而过程定义、用户/应用程序的名称在一致的标准下被处理。分布式工作流系统，在工作流机间采用特殊的协议和信息转换格式，来同步工作流机的操作、过程交换和活动控制信息。也许工作流相关数据也要在工作流机间进行传递。

有许多不同的工具可以用来分析、建模、描述业务过程；这样的工具有很大的不同从非正式的（铅笔和纸）到成熟的、十分专业的。工作流模型不关系这些工具的特性，也不关心在过程建立时期他们是如何交互的。有的工作流产品提供了其自己的过程定义工具，从而过程定义一般是保留在工作流产品范围内的，并且可能或者不能被，读/写信息的编程接口所访问。而使用单独的过程定义和执行服务器产品，过程定义能够在不同的产品间进行转换，并可以被其他产品访问。设计活动和最后的过程模型输出，称为过程定义。在运行时期过程定义可以被工作流机解释。过程分析工具、建模工具和定义工具，都要有在一个组织结构中模拟过程的能力（尽管这不是工作流参考模型规定必须有的）。如果组织模型集成到了这些工具中，那么过程定义将包含组织相关对象，例如角色。这些都是与系统相关的控制数据，例如角色：活动者间的关系，可能会在过程执行期间被引用。

任务表处理器是在需要调用人类资源的活动中，用来与终端用户进行交互的软件。任务表处理器可以作为工作流产品的一部分提供给用户，也可以由用户自己开发。在其他情况中，工作流可能要普通的办公系统进行集成，例如Email，来为终端用户提供一个统一的任务管理系统。这就要求在工作流执行服务器与工作流客户端应用程序间有一个非常灵活的通信机制，来构建各种可能遇到的运行系统。在工作流模型中，通过客户端应用程序与工作流机间的定义良好的接口进行交互。在这个接口中包含任务表——由工作流机分配给用户的任务序列。最简单的情况是，工作流机访问任务表，来把任务分配给用户；任务表处理器访问任务表，向任务表中添加任务项。有许多不同的产品来实现任务表的交互。任务表中任务项的激活（例如，启动应用程序，连接工作流相关数据），可能是由工作流客户端应用程序或者终端用户控制的。在工作流客户端应用程序与工作流执行服务器间定义了一系列的方法，用来向任务表中添加任务项、从任务表中删除完成的活动、激活临时挂起的活动，等。任务表处理器也可以调用应用程序，或者直接调用，或者由终端用户调用。通常希望，任务表处理器的应用程序调用范围能够受到运行环境的限制，尽管这样会给模型带来通用性的限制，但这种情况是一直存在的。与任务表相关的部分活动的的数据，是任务表处理器用来调用应用程序所必须的信息。当应用程序数据是强类型时，在任务表处理器中要存放一个联接，用来实现程序的调用。在其他情况中，在任务表处理器与工作流机间要进行完全的应用程序名称和地址信息的交换；

这时， workflow客户端应用程序也可能实现一些应用程序调用接口中的功能，来获得必要的信息。任务表中可能要包含一个过程中的几个不同实例的相关任务，或者包含几个不同过程中的一个共同活动项。一个任务表处理器可能要与几个不同的 workflow机、几个不同的 workflow执行服务器进行交互。（按照每个产品的实现，为每个过程单独维护一个物理上分开的任务表，或者任务表处理器把几个不同的任务表联合到一起，呈现给终端用户所有的 WFM产品都没有足够的逻辑单元，知道如何调用所有的应用程序，这些应用程序存在异种的产品环境中。这就需要，能够处理在所有平台下和网络环境中进行调用的逻辑，并需要能使用公共格式和编码进行应用数据或相关数据传递的方法。然而，许多 workflow系统能够使用了更多受限制的应用程序，特别是那些采用强制数据类型和直接与应用程序相连的系统。在其他情况中，应用程序对操作的调用，可能是通过边准的交换机制来实现的。这些 API 可以被应用工具直接调用；也可以被应用程序代理过过程调用，作为与其他应用程序（不包含任何 workflow技术的程序）交互的前端。

WFMC的一个主要目标是，为不同开发商的 workflow系统产品，相互间能够进行无缝传递任务项，定义标准。workflow产品的特性变化多样。在 WFMC的协同工作标准中，没有强迫开发商必须在①提供一个只面向用户需求的产品②只考虑协同工作，二者中作选择。WFMC把焦点聚集到，开发多种不同的协同工作框架，这些框架可以操作一系列标准的协调工作，从简单的任务传递到整个 workflow系统的协同工作（包括过程定义转换、 workflow相关数据交换、通用的界面等）。简单的协同工作，WFMC的协同工作定义将在最初就能支持；而复杂的协同工作，还需要进一步的研究。尽管可以开发一个非常复杂的协同工作框架，由许多个工作流机构成个执行服务器，但是这种框架不会在近期实现，因为这需要所有的工作流机都可以解释一个公共的过程定义和共享公共的 workflow控制数据集，事实上是维护异种 workflow机间的一个共享过程视图。现阶段更现实的目标是，能够在运行时期传递过程的某些部分，来支持不同的执行服务器运行。

WFMC规范的最后关注的是，为管理和监视功能开发公共的接口标准，这样一个开发商的产品就可以用来管理其他 workflow机的运行。通过公共的接口，几个不同的 workflow执行服务器可以共享，管理和监视功能。尽管，过程状态命令在接口定义中已经描述了，但一致认为，在某些行业中需要，进行全部状态监视和提取信息的功能。WFMC提出的接口，是要让用户能够得到 workflow运行状态的完整视图，无论是什么样的 workflow系统；同时，也希望能提供一套全面的功能集，进行系统管理，包括安全性、控制和权限。接口中包含 WAPI集中的一些具体命令，来操作管理和监视功能。另外，进一步的讨论，期望能够确定在什么范围内，这个接口可以使用现有的协议（如 CMIP、SNMP），来设置、恢复管理状态和统计信息。

### 2.1.3 workflow技术应用目标领域

使用 workflow 管理系统的目的之一是作为企业应用系统集成的平台。在当前大部分企业级 IT 架构中, 各种各样的异构应用和数据库运行在企业内网中。在这些系统被应用到组织时, 都有一个清晰的目标。例如, 客户管理、文档管理、供应链、订单、支付、资源计划等等。让我们称这些系统为专门应用。每一个专门应用都包含它们所支持业务流程的领域知识。这些专门应用中的自动化流程, 被拼装到企业中更大的非自动化流程中。每当一个这样的专门应用安装并投入使用, 都会带来涉及其他多个应用的新功能需求。企业应用系统集成就是通过使用多个专门应用满足软件新需求的方法。有时, 这只需要在两个应用之间提供数据通讯的通道。专门应用将很多业务流程硬编码在软件中。而 workflow 管理系统是不必事先知道问题域的相关信息的。workflow 管理系统将业务流程描述作为输入并管理流程实例的执行, 这使得它比专门应用更灵活。workflow 管理系统可以用来管理全局的业务流程。如果专门应用支持你所需要的业务流程, 那么使用专门应用。在此讨论的 workflow 管理系统的第一种使用方式就是: 结合所有的专门应用, 使用 workflow 管理系统构建一个集成平台。

workflow 管理系统能够发挥很大价值的第二个使用方式是: 协助涉及多人相关任务 workflow 软件的开发。为了达到这个目的, 大部分 workflow 管理系统都有一个方便的机制, 来生成执行任务的表单。采用这种方式使用 workflow 管理系统能够显著提高生产率。不用将过程用文字的形式写在纸上, workflow 管理系统使你通过流程定义建模实现过程的自动化。

workflow 管理系统的第三种使用方式是: 将 workflow 引擎嵌入到其他应用中。在前面我们谈到, 专门应用将指定问题域相关的业务流程固化在软件中。开发专门应用的公司也可以将 workflow 引擎嵌入到他们的软件中。在这里, workflow 引擎只是作为一个软件组件, 对于应用的最终用户是不可见的。将 workflow 引擎嵌入到应用中的主要原因是为了重用和应用软件的可维护性。

### 2.1.4 workflow技术在信息系统中的应用现状

当前, workflow 技术取得一定的成果, 但应用面还比较狭窄, 它被有限应用于几个方面。

**办公自动化:** 系统发展已经经历了从个体工作自动化、工作流程自动化到已知识管理和信息服务为核心的三代发展, 从简单电子邮件、群件, 到构建 Web 应用等各种方式来获取、存储、提炼和利用知识等。目前越来越多的企业将办公自动化系统作为企业知识管理和信息集成的综合门户。<sup>[9,10,11]</sup>

**电子商务模型:** 在电子商务模式下, 企业可以采用电子交易的方式, 企业之间多个有商务关系的企业之间利用网络手段进行有效的信息交换, 实现电子协作。由于电子商

务的分布式计算模型，一个业务流程需要多个异构系统的参与，电子商务活动中每个过程是在计算机支持下的多方协同工作，这使得 workflow 成为最适合的业务集成平台，因而电子商务系统中 workflow 被广泛使用，下图为电子商务模型。

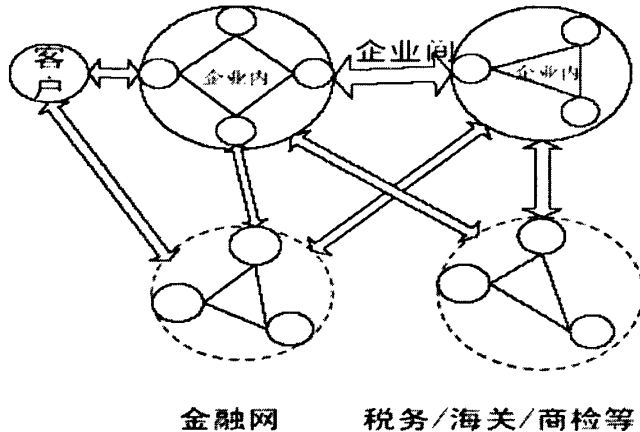


图 2-3 电子商务模型

商业流程：商业流程是一系列结构性的工作任务，或者按序排列，或者平行排列，一般是同一目标下由两个或更多的人来完成；而 workflow 系统则是指商业流程中的人工内容自动化。所以，workflow 是商业流程的一部分，且 workflow 系统是商业流程管理的主要特征简言之，采用作业流构建商务流程系统正是 workflow 管理系统的强项，因而在商务流程系统中 workflow 已被广泛采用。

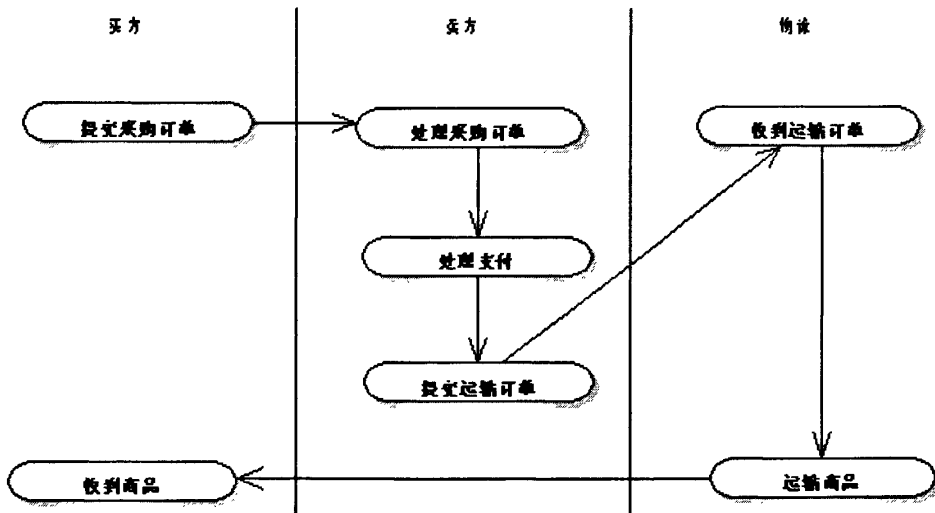


图 2-4 商业流程

但是从workflow在信息系统中的实际应用状况来看，目前仅有部分企业在经营过程中采用workflow，而且系统的应用范围也很有限，并不能全方位支持企业的关键流程。究其原因有：各个workflow产品彼此都不兼容，能力有限，对大规模业务的负荷无法胜任；鲁棒性与实用性差。在workflow管理系统的开发的最初阶段缺乏统一的标准以及限制于当时的软件支撑技术的水平，导致不同公司的workflow管理系统在功能上，所采用支撑技术上、开发技术上和接口上都呈现出非常大的不同。

目前，workflow技术在管理信息系统开发中表现得不成熟，较为根本的原应主要表现在以下几个方面：首先，缺乏一种能够支持过程定义、过程演进以及过程分析的形式化数学模型，这使得workflow在信息系统的许多关键特征无法得到保证，包括过程模型的柔性定义，过程重用、事务管理、异常处理等，这都大大限制了workflow在企业信息系统中的普及和推广。其次，缺乏一个标准化的集成框架来支持对企业常用的分布式应用集成。企业在应用workflow进行业务流程的运作时，最为关心的就是workflow能否与企业原有的各个应用系统紧密集成起来，使它们成为一个完整业务流程当中的有机组成部分，但是就目前来看，制约workflow应用中的瓶颈问题就是workflow管理系统所能支持的企业应用太少。最后，在缺乏仿真方法和仿真工具支持的情况下，贸然将workflow引入到信息系统中，将会使人们难以预测所部署的workflow过程将可能出现怎样的结果，它有哪些不合理的地方，它的性能指标如何，这一切都必须等到实际运行以后才能由workflow管理系统所记录的数据中获得。

以上的远因使得当前管理信息系统开发中使用workflow还远远未达到成熟的地步，需要进一步研究如何将两者有机的结合起来。

## 2.2 基于workflow的油气信息系统

### 2.2.1 workflow技术在石油信息系统中的应用现状

workflow与石油行业信息管理系统的概念，已经被一些研究者联系在一起研究和应用。当前workflow在石油行业中从已知道的报告来看，还处于初级应用阶段，一般应用在石油行业的办公自动自动化系统中，而对于石油行业迫切需要的数据流动自动化，数据处理自动化、数据集成自动化以及业务集成自动化方面仍然很缺乏。究其原因，除了workflow自身不成熟导致的问题外，还存在以下几方面问题：首先，油气信息系统本身较为复杂，流程多以数据自动化为主，这使得目前以面向办公自动化为主的大多数workflow系统难以适应。其次，企业业务集成本身就是一个很复杂的问题，如实现组织中不同部门软件系统的整合，从而实现跨部门的应用共享，对于组织中那些原先不能实现有效信息通信和共享的软件系统，使组织的操作过程流水线化这是一个十分棘手的难题，现有的技术能力难以保证集成紧密性。最后，除了技术问题外，是由行业本身也存在问题，即企业本

身对于油气信息系统的重视程度，员工对于软件的接受程度，和国内石油行业业务流程的非标准化都造成了 workflow 系统在石油行业的应用困境。

## 2.2.2 现有产品的不足

在石油行业中的许多业务过程比较复杂，且这业务活动的发生及有顺序关系，也有并行关系，相互间的数据依赖关系也较为复杂，这需要我们保证数据能够方便的在不同的业务中自动的流动，得到所需要的数据，数据管理不再是简单的处理数据，而是如何有效的处理和整合数据。目前应用于油田的信息系统，大多数实际上还是一套 MIS 系统，侧重于企业基本业务数据的维护，以及基本的业务流程处理，这些流程大多数是一些比较简单的，固定的流程，都是比较简单的、固定的流程，无法满足业务流程将来的扩充和对其他业务系统的集成。

显然，传统的油气信息系统在解决上诉问题时都存在这样那样的缺陷，比如固定的业务流程阻碍了企业低成本业务重组的需求，另外也造成了与其他业务系统的集成困难。建立开放的软件体系结构是解决上述问题的一个关键。基于组件和面向对象的多种复用机制来构建企业的业务功能模块，将 workflow 引入到油气信息系统中是一个能够很好解决上述问题的方案。

## 2.2.3 workflow 与油气管理信息整合的优点

正是由于现有的系统有这样那样的不足，在油气信息系统中引入 workflow 使得具有流程的业务活动可以使用 workflow 模型来描述，然后再在 workflow 引擎的驱动下自动按照预先定义的业务流程执行，包括油气信息系统中的复杂业务的执行、与第三方系统的集成<sup>[12,13,14,15]</sup>等。我们可以描述如下场景。

在应用之前，首先根据企业对有关业务流程制定的规则，运用 workflow 建模工具进行流程建模。经过定义的业务流程被载入 workflow 引擎，workflow 引擎按照预先定义好的业务流程来确定各业务的下一步走向，各项业务任务会在准确地时间发往准确的执行人的业务桌面。业务人员只需按照业务桌面的任务清单执行就可以。同时任务清单还提供任务的处理时间限制等要求，这样业务人员可以有选择地优先处理时间紧迫的任务，从而可以避免传统手工处理过程中由于大量业务堆积而造成的一些重要或紧迫的任务因没有及时发现而被耽误的现象。

workflow 管理系统运行于系统的底层，可以十分柔性的处理业务系统。通过定义适当粒度的业务组件，可以使得一些节点具有业务数据的录入、查询功能，甚至在流程运转的某些时间点执行与业务相关的计算或自动过程，而另外一些节点可以调用 Webservice 服务，执行第三方的应用系统譬如：处理数据集成的 SQLServer 集成服务<sup>[16]</sup>。以这种方式，在流程的任意节点都可以完成与系统的交互，这样不仅使得业务流程处于严格的业

务规则监控之下，而且有助于形成更加人性化的人机界面。因为所有需要人员参与的活动都可以通过定义业务流程以工作项目的方式发往执行人的任务列表，系统的参与人只需要与工作项目管理器交互，就可以便捷的得到需要自完成的工作，而不用再去通过菜单等方式进入各个子系统去查看是否有特定的业务需要自己处理。

石油企业组织的运营都包含了许多业务流程，建设高效实用的软件系统的重要工作之一就是对用户工作流程的分析、建模和实施。工作流是支持企业业务过程重组和油气信息系统动态重构的关键技术，工作流技术支持应用程序按用户定义的流程或路线进行运行。传统的油气信息系统将业务流程固化在系统中，不能适应企业的流程变化。基于工作流的油气信息系统，可以通过流程的再定义，灵活地将应用系统的功能连接在一起，快速完成油气信息系统的搭建。

综上所述，工作流管理技术是解决业务过程集成的重要手段，它管理信息系统的集成，实现业务流程的管理、控制和过程的自动化，使业务系统真正协同，实现企业业务流程的重构。所以工作流管理技术和将工作流技术与石油行业管理信息系统结合受到人们的高度重视并得到快速的发展。

### 2.2.4 基于工作流的油气信息开发流程

本节综合分析比较各种软件开发方法、结合网络化信息时代的特点及要求确立了适合开发分布式油气信息系统的工作流的概念，进而引用了基于工作流的油气信息系统软件设计方法及软件开发方法的具体步骤。

#### 1. 系统分析

首先就要对系统作深入的需求分析，明确用户单位的组织结构，各个组织机构之间及其内部的具体业务处理过程、处理过程中的数据来源及流向。

#### 2. 工作流建模

在系统分析的基础上，按照工作的相关性将系统划分为工作流的集合。首先分析现实系统的各个组织机构，将用户提出的特殊功能要求也视作一个组织机构。然后对每个组织机构的各个工作进行细化，即定义每个工作的工作过程、活动内容、数据来源、数据去向及约束性，并且为各部门拟定工作列表。最后对各个组织机构的工作进行归纳，提取工作流。

#### 3. 工作流优化

上述步骤中提取的工作流模型称作具体工作流模型，它还与具体的组织机构有着一定的联系，必须加以优化才能得到与具体的组织机构无关的逻辑模型，此外从实际系统中直接提取的工作流模型不一定是最优系统，有必要进行优化。优化的另一个目的是解决工作流程中的瓶颈问题，提高工作效率。

#### 4. 工作流节点组件的构建



定义企业业务逻辑的过程，可以根据每个企业业务需求组合这些原子活动和组件，以建立一个可实际运行的工作流系统。这些原子活动和组件代表着工作流程中的各个节点的具体功能，因而需要开发这些节点组件以满足业务的实际需求。

#### 5. 工作流调试

使用仿真数据测试业务流程原型、并对原型提出修改意见，采用迭代方式，设计人员根据意见不断修改、测试并完善工作流原型，直到所有的工作流程都满足业务需求为止。<sup>[17]</sup>

#### 6. 系统集成

系统集成也就是工作流集成，完善用户界面。因为工作流基本是相对独立的，所以在集成时工作流的运行机制基本不需要改变，这时的重点是为用户提供一个集成的用户界面。<sup>[18,19]</sup>

## 第三章 基于组件的工作流引擎设计与开发

### 3.1 工作流引擎分析

#### 3.1.1 基于组件轻量级工作流

本章以一个油气信息系统为实例，同时分析了石油行业的关键业务的基本特征，针对关键业务的开发需求基础上，讨论一个基于组件技术的轻量级工作流引擎的具体的设计原理与实现方法。它充分考虑了关键业务开发过程中对工作流功能的需求，利用此工作流引擎，构造出具有工作流特征的油气管理信息系统。本章的侧重点不在于完全实现一种功能齐全的工作流管理系统，而仅仅设计和实现工作流管理系统的核心--工作流引擎。

基于组件的轻量级工作流引擎指的是从够用、灵活和低成本的设计原则出发，不追求工作流引擎的功能的完备和复杂，只是实现其中必不可少的功能和特征。在设计工作流引擎时主要考虑对其组件模型的定义和解释、活动之间的协调等功能提供支持，而不支持诸如提供内建的组织模型、对应用数据的定义和完整性维护、完善的异常处理以及长事务控制等功能。石油行业的业务任务的发起和结束是一项过程化的任务，任务又可以分解成一个一个环节任务，而任务是带有目的性的，由这个目的去拉动这个过程中的一个一个的环节任务，促使环节任务的推进，最终达到任务完成的目的。这些业务的过程化不是随机的，而是已经严格规定好的，只有遵循这些过程化的规则和流程环节才能完成整个业务。基于组件的轻量级的工作流引擎就组合了以上这些特点，不追求工作流引擎的功能的完备和复杂，以满足一般性业务为目的，为石油行业快速开发出适合他们业务的工作流管理系统。

基于组件的轻量级工作流引擎它利用组件技术实现工作流引擎的控制逻辑；它提供了一系列比较完备的应用程序编程接口，应用的开发者可以将这些应用程序编程接口嵌入到自己的应用系统中从而实现具有工作流性质的信息系统。基于组件的轻量级工作流引擎的适用对象并非应用系统的最终用户，而是利用专用开发工具构造相应应用系统的专业开发人员。它为开发人员提供了驱动工作流的机制的支持，从而构造出各种灵活的具有工作流特征的应用系统。

#### 3.1.2 工作流引擎设计思路

本章所设计的工作流是基于组件的轻量级工作流系统，它以活动为工作程序的基本构造块。工作程序中的一个活动代表了一个可恢复语句组件。工作流编程模型的活动分为两类：简单活动和复合活动。我们把包含其它活动的活动称为复合活动，简单活

动则是一个不包含其它活动的活动。基本活动代表一组特定的表达式，而复合活动用作工作程序的控制流，复合活动管理了活动的执行。工作程序实际上就是一个活动的层次结构，准确地说，工作程序是由复合活动和简单活动构造出来的一个活动树，活动树本身也是一个活动。当工作程序执行时，工作流运行时加载活动树，并把它驻留在 java 虚拟机中，并在程序间断执行中，提供对工作程序实例处于非活动状态的钝化。这使得在适当的时候，工作程序实例可以被激活并重新载入执行。以下为工作程序的执行过程。

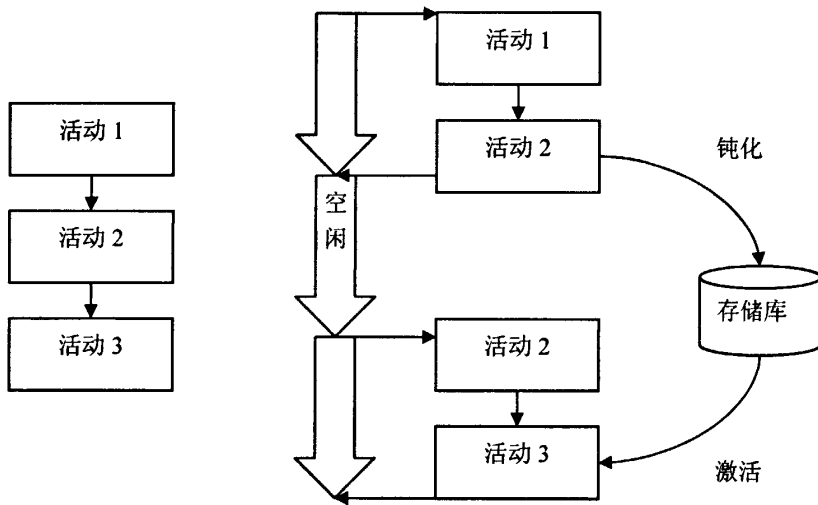


图 3-1 工作流执行过程

基于组件的轻量级工作流编程模型是根据活动自动机来实现工作流运行时和活动之间的交互的，在工作程序执行过程中，活动机掌握着所有活动的生命周期，并控制着状态的转移。工作流运行时处理从一个状态到另一个状态的转移过程，以保证工作程序执行的正确性。存在于工作流运行时和活动之间的契约是，工作流运行时的职责是确保活动的执行严格地遵守活动自动机的转移规则。活动的职责是决定状态转移的路线。

基于组件的轻量级工作流中包括两类工作流：顺序工作流和状态机工作流。顺序工作流，顺序模式是一直向前执行，它在重复和、可预见的操作（这些操作通常相同）中非常有用，例如，一系列的活动以规定的、通常相同的顺序来运行。顺序工作流以按顺序执行的方式来执行活动，直到最后一个活动执行完成。状态机工作流，状态机模式由一系列事件驱动的状态组成，数据驱动模式依赖于数据来决定某个活动是否执行，在状态机模型中，工作流就是一个状态机，它由一系列的状态组成。其中一个状态指示了开始状态，每个状态都可以接收一系列的事件（客户端调用）触发，工作流从一个状态迁移到另一个状态，状态机工作流可以有一个终结状态，当迁移到终结状态时，工作流就

完成了。本文的设计的轻量级工作流以状态机工作流为主，每一个状态由一或多个状态活动组件构成，状态活动组件内部由顺序工作流构成每一个状态的内部业务活动。

## 3.2 工作流引擎设计

### 3.2.1 工作流引擎总体结构

整个工作流引擎包含流程描述文件，工作流运行时，工作流实例和工作流库。流程描述文档描述了工作流程的构成，工作流运行时载入描述文件并根据它生成由活动组成的工作流实例，然后通过内部调度执行工作流实例，如果需要则持久化工作流实例和上下文参数到数据库中。

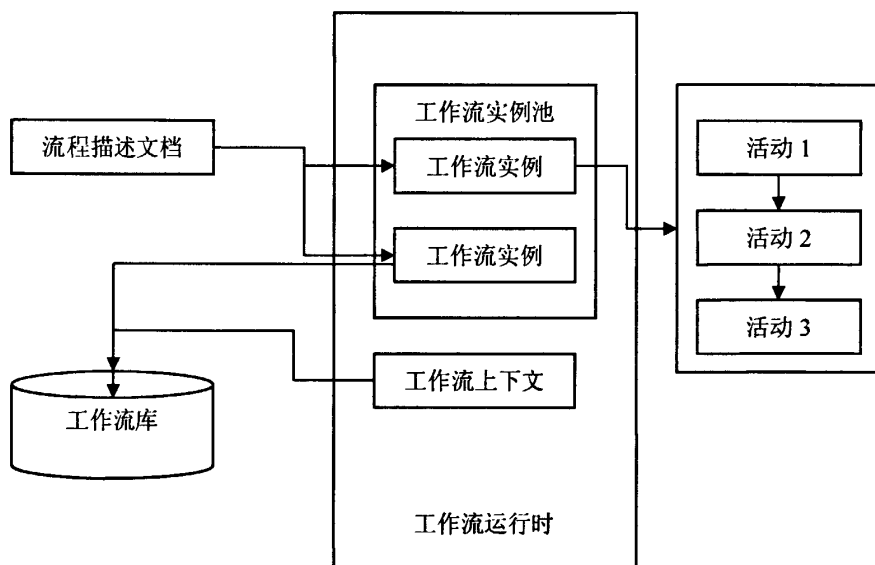


图 3-2 工作流引擎总体结构图

### 3.2.2 工作流引擎设计静态模型

引擎主要处理的是实现流程实例、活动和工作项三类逻辑对象之间的逻辑关系和相互的约束。其中流程实例以流程定义为参照，活动以活动定义为参照，工作项以活动为参照。它们共同协作完成整个系统的逻辑功能。

整个工作流引擎包含四个部分：工作流运行时、流程实例、活动和服务。工作流运行时负责管理所有工作流引擎的生命周期，流程例是流程定义的一次动态执行过程。活动负责构成流程的具体内容，服务则提供活动对外部资源的访问。

工作流引擎的类设计可以对应以下几部分：

运行时：**WorkflowRuntime**，这是一个具体的类，它负责管理整个引擎的生命周期，包括启动流程，查找流程，持久化流程，注册服务等。

流程实例：**Instance**，这是一个具体的类，它负责管理流程的生命周期，维护活动树，执行活动树等。

活动：根据流程定义的划分，活动包括流程控制活动，功能提供活动两大类，这里活动被设计为一组继承结构，所有活动继承自 **Activity** 接口，其中个体活动从 **Activity** 派生，组合活动从 **CompositeActivity** 派生。

上下：**WorkflowContext**，代表流程执行过程中的相关信息。

活动构造服务：**ActivityBuilder**，负责从流程描述文件中构造 workflow 活动树。

持久化服务：**SqlPersistenceService**，负责将流程示例持久化到关系数据库中，和从关系数据库中恢复实例。

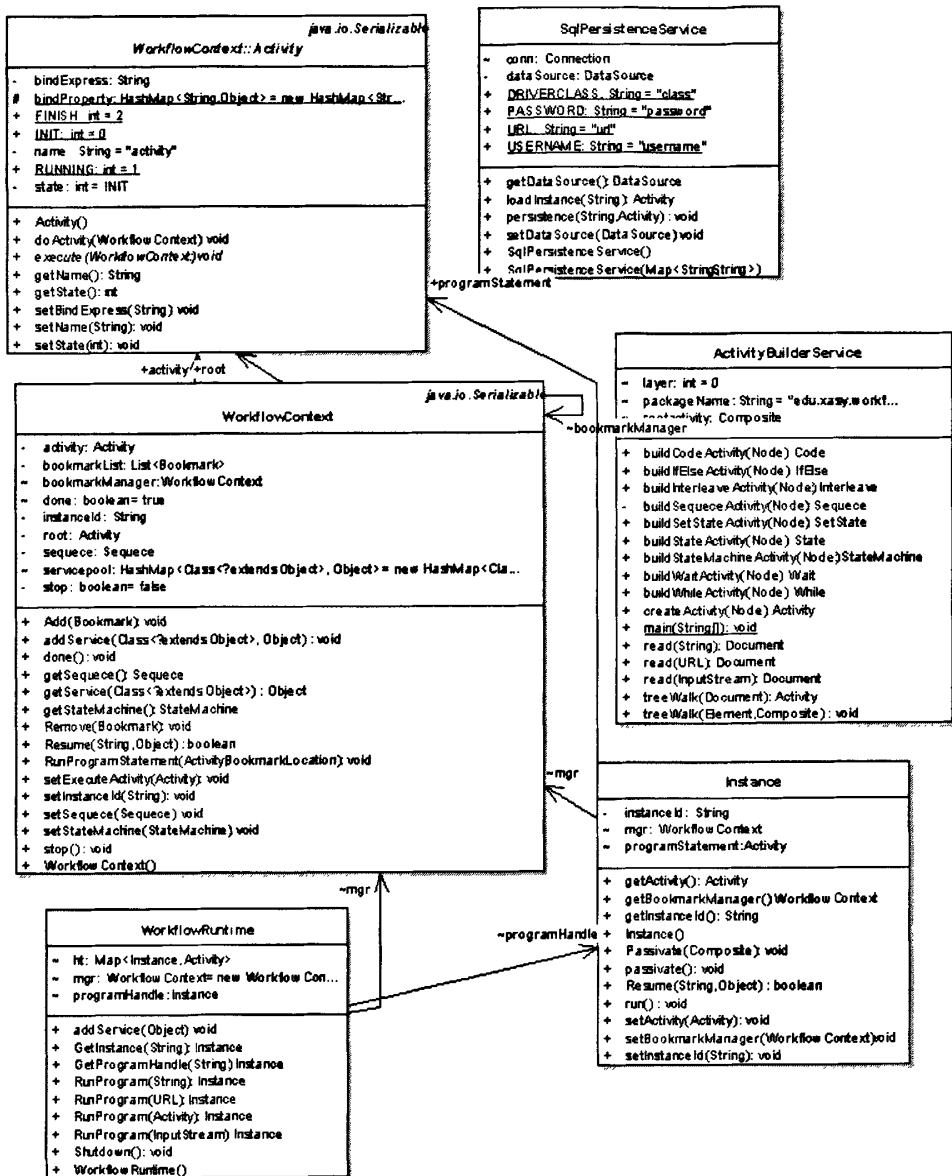


图 3-3 引擎主要类结构图

### 3.2.3 工作流实例状态转换规则

- 1)初始态: 一个过程实例已被创建, 但还没有满足启动执行的条件并收到和响应一个启动过程事件。
- 2)运行态: 过程实例已经启动, 但仍没有一个状态活动处于激活状态。
- 3)挂起态: 过程实例处于一种不活动的状态, 在该状态下, 不能启动和操作任何节点实例和工作项, 除非通过一个恢复过程实例的事件返回到运行状态。

4)终止态：在正常结束之前，该过程实例的执行被事件强行结束。终止态也是状态转换中的最后一个状态，之后不会再有状态转换行为发生。

5)激活态：该过程的一个或多个活动实例满足了激活条件并已经被启动且被激活。

6)完成态：过程实例已满足结束条件，被正常结束。完成态势状态转换中的最后一个状态，之后不会再有状态转换行为发生。该结束不是任何强迫行为导致的结果，而是工作项按预定的方案进行正常的状态转换所形成的结果。

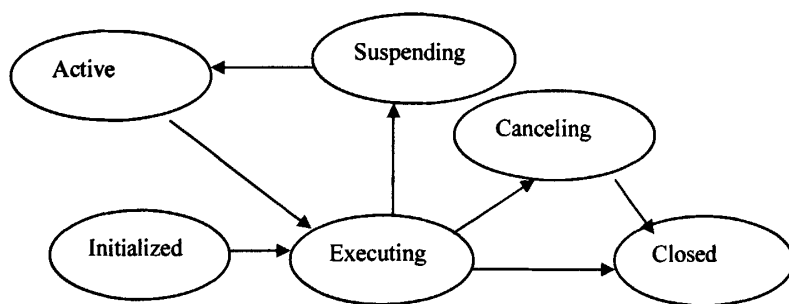


图 3-4 实例状态转换图

### 3.2.4 workflow 活动状态转换规则

(1)初始态：一个活动实例已被创建，但还没有满足启动执行的条件并收到和响应一个启动过程事件。

(2)运行态：活动实例已被创建，但仍没有处于激活状态。

(3)终止态：在活动正常结束之前，该活动实例的执行被事件强行结束。

(4)完成态：过程实例已满足结束条件，被正常结束。完成态势状态转换中的最后一个状态，之后不会再有状态转换行为发生。该结束不是任何强迫行为导致的结果，而是工作项按预定的方案进行正常的状态转换所形成的结果。

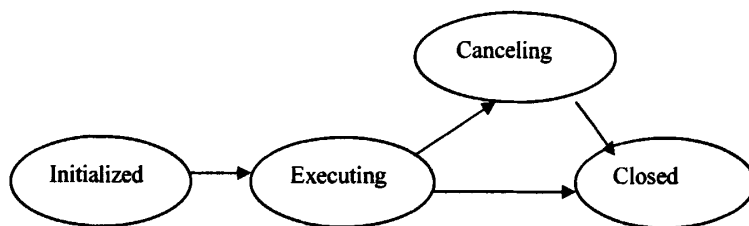


图 3-5 活动状态转换图

### 3.2.5 workflow 组件的构建

一个流程就是将一系列的活动按照某种预定的方式，组合起来的。因此我们可以通

过制定流程的起始条件，流程的约束规则来建立新的功能强大的组件。活动（Activity）是 WMFC 的标准模型元素，描述的是工作流中的一个活动（Activity 是描述流程运转的最小单元）。每个业务过程由若干业务活动组成，不同的业务活动通过各不相同的活动（Activity）组合来构造。同一个业务活动在工作流运行时可能具有多个实例（instance）。我们将活动的实例称为任务，将属于同一业务过程的任务称为属于同一批次的任务。有的业务活动可能针对具体的业务环节，即在前台（后台）对应实际的应用逻辑；有的业务活动则不针对具体的业务环节。

基本活动组件包含两类：流程控制活动组件和功能活动组件。流程控制活动组件负责控制流程，包括状态组件、串行组件、循环组件，并行组件、选择组件、状态转换组件等；功能组件则负责实现功能模块，比如邮件发送组件、webservice 服务调用组件。这里流程活动组件是工作流组件的基础。

下面首先分别讨论如何构建基本的流程控制活动组件。

串行组件：串行组件按照预定的组件顺序，有序的执行各个内部活动。活动B必须在活动A执行完之后才能执行；活动B的输入是活动A的输出。

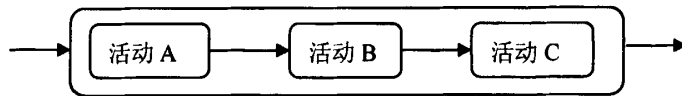


图 3-6 串行组件

循环组件：循环组件主要用于表示，同一个任务节点，根据一定的条件重复的执行多次。活动A根据执行条件重复执行活动A多次。

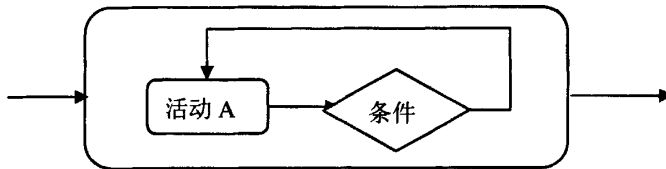


图 3-7 循环组件

并行组件：在流程运行过程中，因为不同的条件或情况，或者处理的业务需要多部门（多任务）分开处理，而产生了流程分支。流程在执行完活动A后，因为需要产生了两个并发执行的分支（A---B 和A---C），这两个分支之间是对等的，也是并行执行的

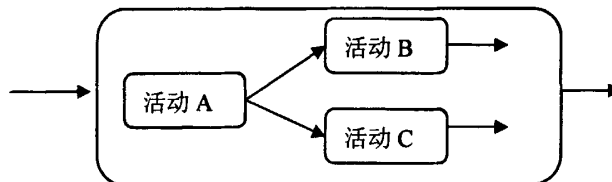


图 3-8 并行组件



选择组件：当一个任务处理完后，当任务达到这个鉴别器的时候，鉴别器会根据当前流程所处的状态，对比预先设定的一些选择规则，自动判别接下来流程的流向，也就是自动根据条件，选择一个满足条件的分支运行，只允许选择其中某一个分支运行。虽然在任务活动 A 后预订了两个不同的任务，但是仅活动 C 满足条件，造成后续的流程中，走了 A——C 分支，而另外的分支被抛弃。

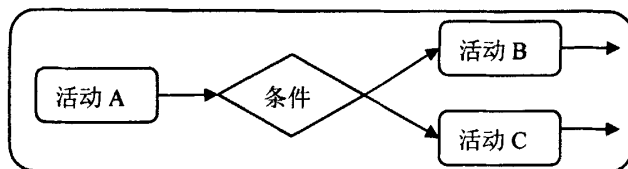


图 3-9 选择组件

状态组件：业务活动的各个环节的由状态组件构成，业务活动通过状态之间的转换构成整个业务流程，状态组件包含一个自身状态，内部由多个流程控制活动组件构成，并通过状态转化活动提供状态转换。

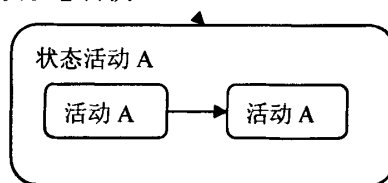


图 3-10 状态组件

状态转换组件：当一个状态组件中的一系列活动执行完毕之后，状态转换组件可使设置工作流程流转的下一个状态，从而使 workflow 从一个状态迁移到另一个状态。活动 A 执行完毕后，根据选择条件，活动 B 使得状态 A 迁移到状态 B 或者活动 A 使得状态 A 迁移到状态 B

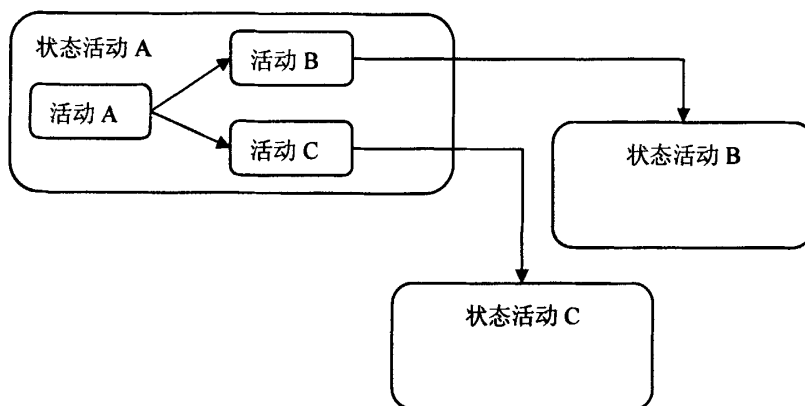


图 3-11 状态转换组件

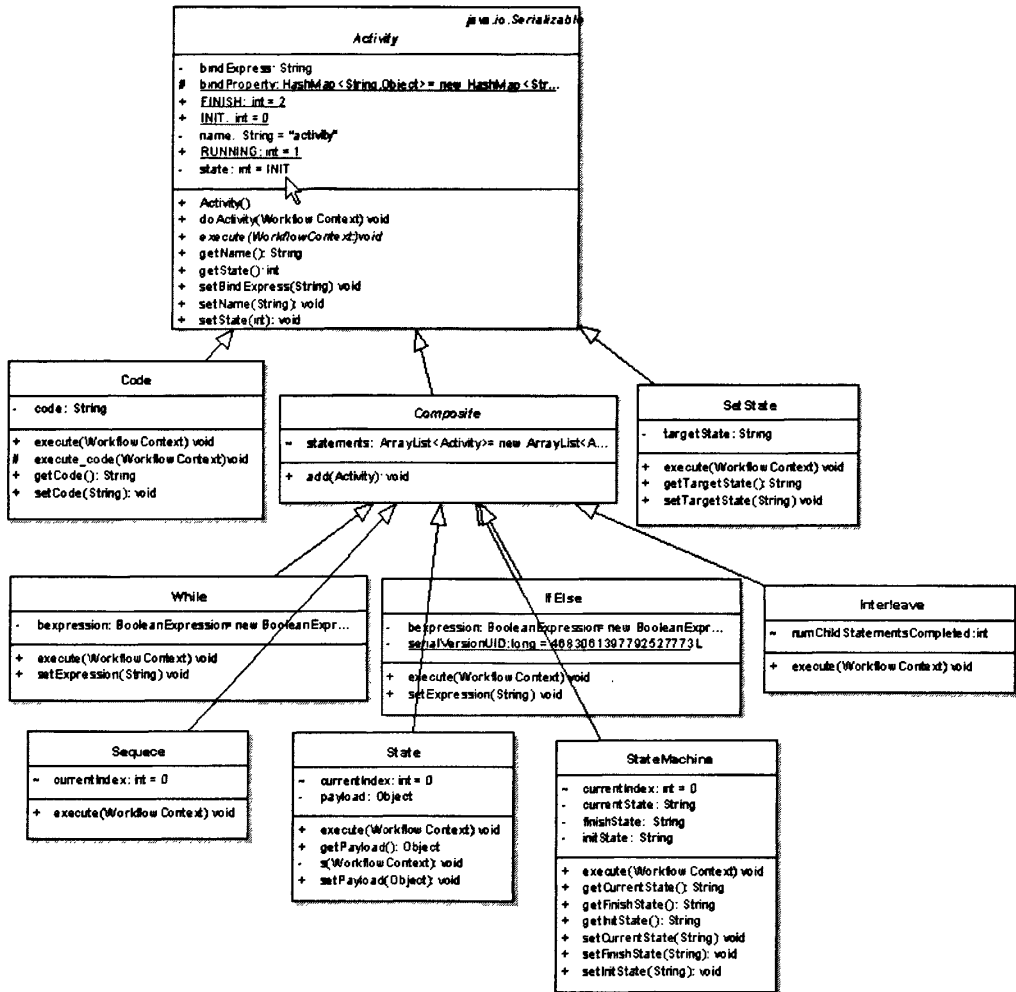


图 3-12 基本活动类结构图

### 3.2.6 工作流持久化数据模型

在部署一个流程后，工作流运行时加载活动树，并把它驻留在就 java 虚拟机中，然后执行，当用户终止引擎或是工作流实例长时间的处于空闲状态，则对工作流实例进行钝化，在适当的时候工作流实例可以被激活并重新载入执行。钝化时实例相关信息需要写入数据库，同时工作流程中的流转数据也需要被存贮到数据库中。其表结构如下。

工作流实例表：纪录了工作流实例被持久化后的相关信息。包括实例 ID、工作流名称、工作流实例、工作流当前状态、创建时间，完成时间。

表 3-1 工作流实例表

INSTANCE	
INSTANCEID	Varchar (20)
NAME	Varchar (20)
STATE	image
STATUS	Varchar (20)
CREATETIME	datetime
FINISHTIME	datetime

工作流实例参数表：纪录了工作流实例中相关的参数信息。包括实例 ID、参数名称，参数类型以及以不同类型存贮的参数值。

表 3-2 工作流参数表

INSTANCE	
INSTANCEID	Varchar (20)
KEY	Varchar (20)
TYPE	int
NUMBER_VALUE	bigint
STRING_VALUE	Varchar (20)
FLOAT_VALUE	float
DATE_VALUE	datetime
DATA_VALUE	image

3.2.7 工作流引擎关键类分析

下面分析工作流引擎设计中主要类的重要方法。

表 3-3 WorkflowRuntime 类方法说明

Instance creatWorkflow(String workName)	根据参数加载流程描述文件创建一个工作流实例。
Instance getWrokflow(String instanceid)	接收工作流实例编号，从工作流实例池中查找实例，如果实例池中并没有相应的实例则从工作流实例库中查找并加入实例池。
Instance loadInstance(String instanceid)	从工作流实例库中查找实例。
void persistenceWorkflow()	将工作流实例池中的所有实例持久化到工作流实例库，同时持久化工作流上下文中的参数。
void stopRuntime()	停止工作流运行时，并调用 persistenceWorkflow()。
void addService(Service service)	注册本地服务，以便于为工作流运行时和流程中的活动提供业务服务。

表 3-4 Instance 类方法说明

void execute(Map args)	接收上下文参数并执行工作流实例。
String getCurrentState()	获取工作流实例当前状态。

表 3-5 WorkflowContext 类方法说明

void add(Bookmark bookmark)	将活动可恢复标记加入。
Void remove(Bookmark bookmark)	删除活动可恢复标记。
boolean Resume(String bookmarkName, Object payload)	恢复执行标记活动并提供补偿数据
done()	活动从执行态迁移到完成态
void runProgramStatement(final Activity activity, final BookmarkLocation continueAt)	执行活动

表 3-6 Activity 类方法说明

void execute(WorkflowContext context)	接收上下文参数并执行活动。
---------------------------------------	---------------

## 第四章 基于工作流的油气信息系统总体设计

### 4.1 系统设计目标

随着石油行业不断深入发展,现有的油气信息管理系统已不能满足企业对系统日益复杂业务的需求,企业需要的系统是具有高扩展性、高集成性和能柔性适应复杂且变化迅速的工作流程,需要处理大量的复杂业务,并能与企业现存的其他系统良好集成。在设计的过程中,综合考虑各方面因素,系统最终实现以下目标:

首先,要使原来部门协作时工作效率低下的情况得以逐步改变,为部门协同工作提供自动化,促进工作及时进行。通过系统本身的自动化功能使部门间的工作由原先的一环接一环的比较松散的结构向紧凑的工作流程结构转变,使部门间的协同工作真正成为协作,摆脱以往多个部门协作时出现的互相等待状况。

其次,提高企业业务处理的自动化程度,传统的油气信息系统在数据业务处理中,能以满足油气信息数据处理自动化的需求。采用基于 workflow 开发油气信息系统,可以通过流程的定义,灵活地实现企业业务数据的自动化过程。<sup>[20,21]</sup>

最后,支持对现有企业业务系统(财务,人力等)的集成,使原本互相独立的系统紧密协作,最大限度的利用已有企业资源,另外支持数据集成以提供对多源头数据的整合及数据仓库的支持。<sup>[22]</sup>

### 4.2 软件体系结构设计

#### 4.2.1 系统总体设计

油气信息系统的特点是既存在复杂的业务流程的运行需求,又存在于现有业务集成的功能需求。在这种情况下,采用以 workflow 引擎为核心的系统设计为最佳的方案,既能应对复杂流程的运行,又能满足业务集成需求,同时又能支持一般业务需求。油气信息系统体系结构设计如下。

针对系统的目标做出的主要构架决策是在 J2EE 平台采用基于组件的轻量级 workflow 引擎构架油气信息系统。该系统属于分层架构<sup>[23,24]</sup>,由表示层、应用层、工作流层、活动、服务层、持久层等多个层次组成,各层之间相互解耦,workflow 管理系统是系统的核心。从系统的设计上来看,由于整个系统采用了分层模式,各层之间相互独立,便于各自独立实现。下图说明了油气信息系统的体系结构。

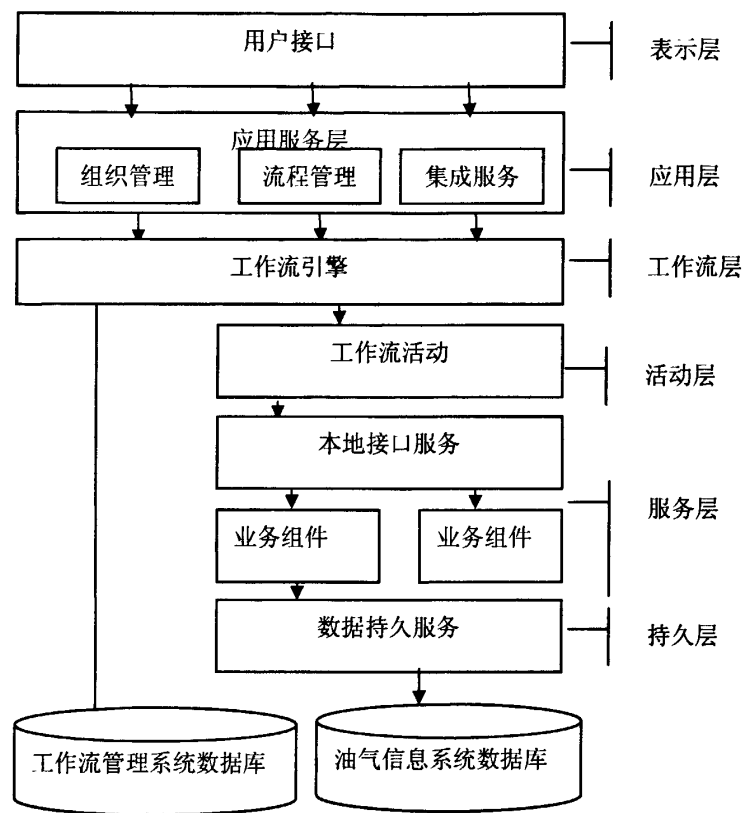


图 4-1 基于工作流的油气信息管理系统的体系结构

数据存储层提供基本的数据存储接口；数据访问层向服务层提供访问数据的功能接口；服务层调用数据访问层提供的功能实现业务流程中的行为；活动层（系统中所有的实体行为都是继承自基于组件的轻量级工作流引擎的 Activity）使用服务层提供的功能；工作流服务层调用活动层提供的业务接口进行流程的运转；应用层提供用户的业务操作接口；表示层提供应用表示。在整个体系设计中，工作流层在整个体现机构中处于承上启下的核心地位，基于组件的轻量级工作流引擎则是该层的核心。整个系统的设计开发都是围绕工作流引擎来进行的。

**表示层：**表示层是系统的用户接口部分，是用户与系统间交互信息的窗口。它包括用户图形界面和与控制层之间通信的逻辑部件，主要职责是向用户显示信息并把从用户那里获取的信息解释成服务层或数据源层上的各种动作。表示层程序的修改不影响其他层次，其功能只限于用户交互，不包括有关业务本身的处理逻辑。

**活动层：**构建工作流程的核心，它通过一系列活动的组合实现一个流程活动树，规定了业务流程的执行过程。

服务层：它是应用的所有领域相关的工作：包括根据输入数据或根据已有数据进行计算，以及根据从表示层接受的命令来确定调度哪些数据源逻辑，在应用设计中，表示层和领域层之间完全解耦，业务请求通过工作流调用序列中的工作流子活动（Activity），在通过本地服务委派给适当的业务组件方法。对于一个系统，业务逻辑可能非常复杂，此时，规则和逻辑描述了许多不同的用例和行为的变化，这里业务实现采用领域模型<sup>[25,26]</sup>，它创建了一张由对象组成的网，其中每一个对象都代表一个有意义的个体，在应用程序中使用领域模型需要建立一个完整的由对象组成的层，这些对象模拟业务活动中的数据或是捕捉业务使用的规则，数据和处理一般整合到一起，从而使得数据和数据之上的操作紧密结合。同时企业应用通常需要提供不同种类的接口来访问内部存储的数据和所实现的逻辑：例如数据加载器、用户界面和系统集成入口。尽管这些接口各异，但他们一般都有一些与应用出程序共有的交互，用来存取和处理数据并调用业务逻辑。这些交互可能很复杂，涉及到跨越多个资源的事物或者需要协同动作的多个响应。本地服务接口定义了应用的边界和工作流的活动（Activity）角度能看到的可操作集合，他封装了应用的业务规则、事务控制及其操作时吸纳给中的响应协调。

持久层：数据持久层是包含数据库管理系统(DBMS)<sup>[27]</sup>，文件管理系统，外部环境资源管理系统等，负责管理对数据和资源的读写、获取与处理。该层包括数据与资源管理的逻辑方法，数据储备和与业务层之间的接口，只有该层可以直接通过逻辑接口同应用程序稳定的数据源打交道。对于大多数企业应用来说，最主要的数据源逻辑就是数据库，它的主要职责是存储持久数据。企业应用中常常需要访问关系数据库或是外部资源如何 XML，活动目录等，为了在业务逻辑中屏蔽这些不同资源的访问，可以采用数据访问对象，它可以屏蔽对不同资源访问，便于程序集中于业务逻辑而不是不同资源的访问。

在整个体系设计中，工作流执行服务在整个体系结构中处于承上启下的核心地位，而工作流执行服务层中的工作流引擎则是该层的核心。整个系统的设计开发都是围绕基于组件的轻量级工作流引擎来进行的。

#### 4.2.2 系统总体接口

系统的总体接口设计如图：

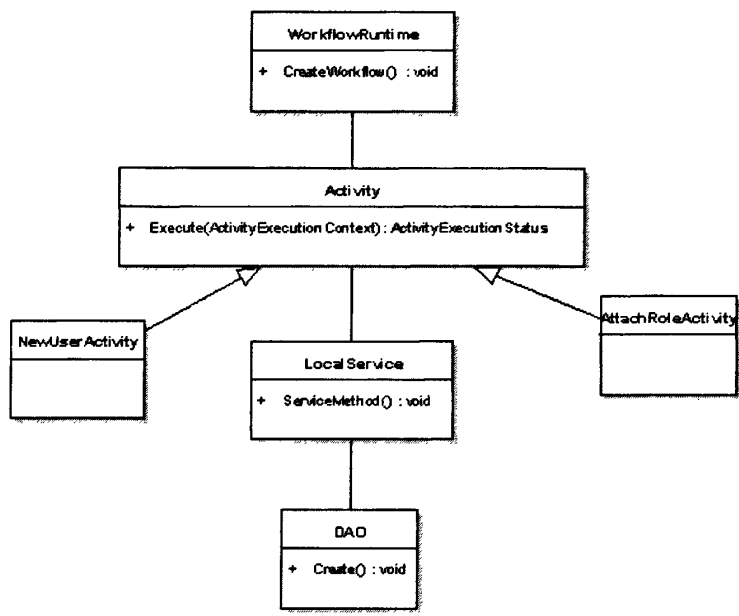


图 4-2 系统总体接口

4.2.3 数据持久层设计

典型的基于工作流的油气信息系统中，对于数据的访问，我们采用具备可持久化的实体来携带需要存取的数据，并通过数据访问对象来访问。流程中，活动将需要保存的数据通过实体对象携带，传递到数据访问对象，并通过数据访问对象持久化。

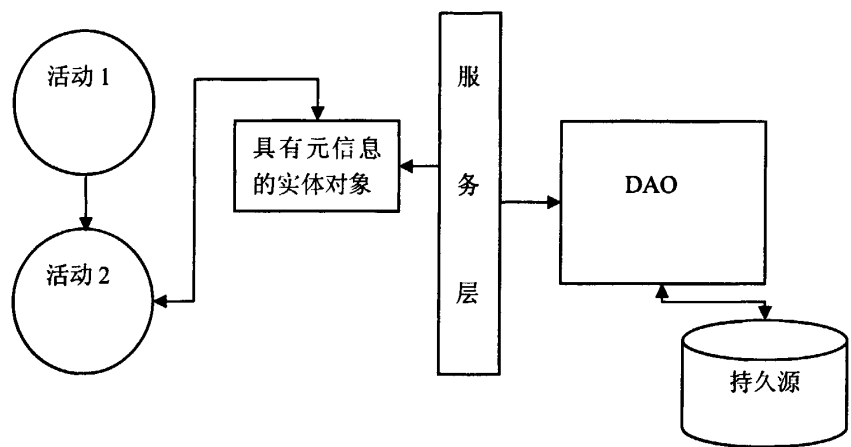


图 4-3 活动层与数据访问层交互

在数据访问层，系统使用 java 持久化 API（JPA），它是专门为 J2EE<sup>[28]</sup>提供的对数据库的访问接口，通过 JPA 可以以面向对象的方式自动处理数据库访问和同步。通过使用



JPA 可以在 J2EE 应用中同大量关系数据库管理系统之间建立起独立于数据库的对象关系映射,从而可以以面向对象的方式访问任何关系数据库,而不必为不同的数据库重写代码。此外通过 JQL 查询语言可以采用相同的方式针对各种数据源查询:SQL 数据库、XML 文档、各种 Web 服务等。同时系统使用了数据访问对象(DAO),同过数据访问对象,我们在业务逻辑层屏蔽了一切关于持久化数据(RDBMS、XML、活动目录等)相关的问题,使得业务逻辑只专注于业务逻辑处理,同样的由于所有的业务规则都集中在中间的业务逻辑层,数据持久层内不需要编写任何业务逻辑代码,因而数据持久层的任务单一,实现起来非常简便。采用 JPA 和数据访问对象使得系统在不同的数据库系统上变得更容易移植。

在面向对象系统中,我们常常希望以面向对象的方式来存储对象所携带的数据,数据持久层的实体/关系映射正是作为对象的永久存储系统而建立的,所以数据库的库表结构由需要永久存储的对象(对应对象)和对象的属性(对应表的列)直接映射而来,类对应表的结构,对象对应数据行,对象和对象间的联系对应表表之间关系。采用实体/关系映射则表的设计变得相对简单,因为类的属性和类的关系在业务层的设计阶段就已经确定了。同其它组件模型一样,J2EE 中 JPA 组件模型通过提供可调用的方法实现从持久对象属性到数据库中对应表的字段的值转换。J2EE 中 JPA 中采用 Annotation 元信息标注方式实现了 JAVA 类映射到数据库中存储方式定义元素的表。JAVA 类中的属性通过元信息标注方式非常容易、直接地映射到数据库表中的列上,并且 JAVA 类自身也可以看成实际的数据库表格,每一个类的实例对象被看作为表中的一个数据行。这样,JAVA 类之间所存在的关系就可以用数据库表之间的主、外键关系来模拟。只要将数据表中的列映射到实体 JAVA 类类的属性上,就可以把到数据库中的视图和表看成是一个 JAVA 类类,JAVA 类类同过持久管理器实现 JAVA 类类实例和与之对应的数据库中对应数据状态之间的同步,这样数据持久层的设计工作变得相当的单纯,仅仅把对象及属性映射到数据库的表结构设计而已。

设计实体/关系映射时应该遵循以下几个原则:

- (1)为每个实体对象在数据库中建立一张对应的表。
- (2)把这些类的属性作为该类所对应的表的列。
- (3)为每个表建立主关键字。
- (4)建立表间的引用约束。

(5)对于实体对象间存在关系的,需要详细分析对象间的联系类型,以便于和数据库表象对应,对象间的关系分为三类:包含、聚合、继承,依赖等,都会在表中有不同的映射表现,具体映射时需要按照不同策略操作。

(6)建立数据库的数据类型和实体对象的数据类型间的数据映射,以便于在由属性到字段的转换工作完成数据类型的转换。

数据访问对象(DAO)是标准数据访问方法,数据访问对象将底层数据访问操作与

高层业务逻辑分离开，它负责和应用中的各种数据源打交道，诸如 SQLSERVER、DB2、Oracle 等关系型数据源、XML 数据以及其它种类的非关系型数据、Web 服务、各种特别的遗留系统等等，将它们整合起来，为业务逻辑层提供统一的数据服务。数一个典型的 DAO 实现通常有以下组件：

- 1) 一个 DAO 工厂类
- 2) 一个 DAO 接口
- 3) 一个实现了 DAO 接口的具体类
- 4) 数据传输对象

这当中具体的 DAO 类包含访问特定数据源的数据的逻辑。下图为油气信息系统中设计的数据访问独象类结构模型。首先设计了 DAOFactory（数据访问对象工厂）它被设计为一个抽象类，它的两个子类分别用了创建持久化到数据库和 XML 文档中的数据访问对象工厂，而在关系数据库数据访问对象工厂里有，则实现创建具体的访问何种数据库的数据访问对象，而具体数据访问对象则通过实现数据问对象接口来分别实现访问不同的数据源。此外，数据访问对象通过数据传输对象，在这里是上面所提到的实体/关系映射中的实体。

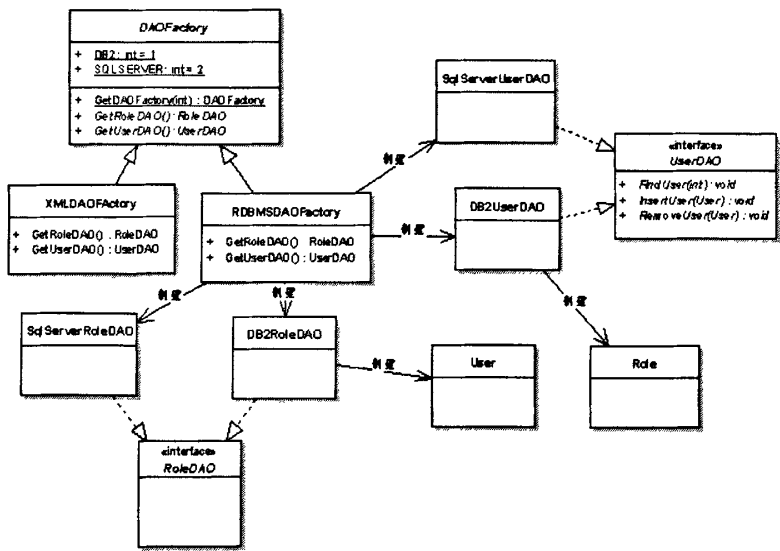


图 4-4 数据访问对象类结构

#### 4.2.4 服务层设计

本文中，服务层的目标是为 workflow 活动提供功能服务，workflow 活动（Activity）本身只负责组织流程，活动（Activity）所需实现的功能由本地接口服务（LocalService）形式提供，这样 workflow 活动（Activity）的业务功能通过本地接口服务（LocalService）委托给普通的领域模型对象来实现，在这种方式，活动（Activity）和领域模型对象各自负责的各自的任务，且使得 workflow 活动（Activity）和提供实际功能的领域模型对象最大限度的解耦和，有利于系统的稳定性和可扩展性。

领域模型对象完成业务实际处理功能，这些领域模型对象可以被发布为分布式服务，以便于跨域调用，实现了面向服务的架构体系。分布式通信技术是分布式企业系统的核心技术。J2EE 框架为 Web 应用和桌面应用提供多种通信模式，使运行于某一机器上的对象能够调用另一台机器上的对象，本文采用 RMI 技术及一种分布式通讯技术。分布式应用程序以其高度的可扩展性和可伸缩性，同时由于资源共享提高了系统的性价比，已经得到广泛的研究和应用，各种分布式计算对象和平台也得到了延伸和拓展。RMI 是 J2EE 框架提供的一项强大的技术，利用它可以使位于任何位置的应用程序互相通信，这些应用程序可能运行在同一台计算机上，或者局域网内的不同计算机上，也可能位于不同地域、有巨大差异的网络中。

被发布为服务的领域模型对象，通过本地服务（LocalService）把领域模型对象的功能发布给 workflow 活动（Activity）来使用，本地服务（LocalService）屏蔽了一些复杂的底层操作如：事务控制、远程访问、同步控制等，他提供给客户的是经过包装的功能接口，使得客户只关心是否有满足要求的功能接口，这极大地简化了 workflow 中业务处理方式。一般而言，我们需要本地服务（LocalService）服务类，它来包装领域模型对象中的功能，另外我们还需要一个或多个领域对象，来真正实现业务功能。

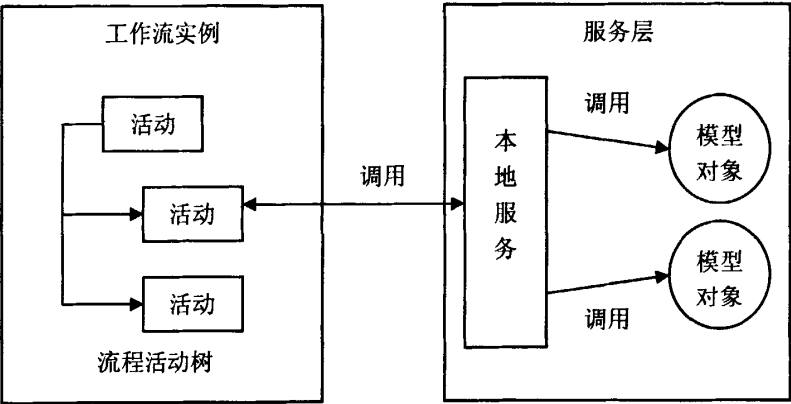


图 4-5 活动层与服务层交互

### 4.2.5 工作流活动层的设计

工作流的流程逻辑是通过组合 Activity（活动）来实现的，系统中所有的业务流程逻辑都是由继承自工作流的 Activity（活动）和 CompositeActivity（复合活动）这个两抽象类来实现，其中 Activity 为独立活动，它不包括子活动，而 CompositeActivity 允许包含子活动，以便在一个活动中实现工作流程，一般来说采用继承 CompositeActivity 的子类构建业务流程逻辑，便于实现。本系统中的所有流程业务逻辑都由工作流提供的 Activity（活动）组件和用户实现的活动组件组合构成。

以下行为层的设计如图：

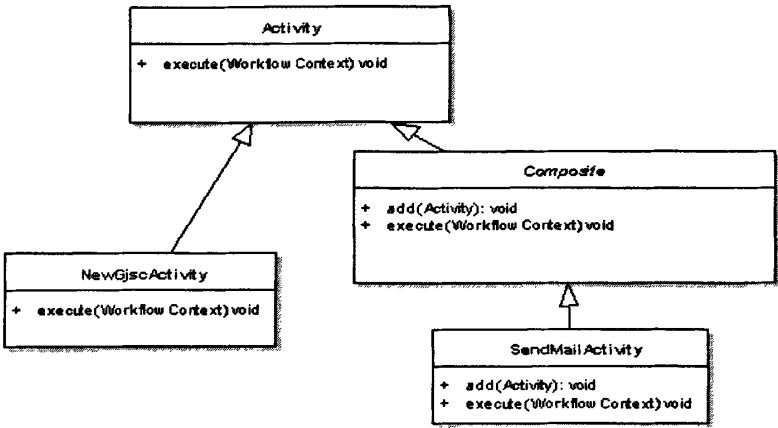


图 4-6 活动层接口设计

## 4.2.6 表示层设计

系统的 WEB 表示层包括 JSP 页面和少量的静态 HTML 文件。JSP 用来动态显示那些系统提供给用户服务的表示内容，而 HTML 则可以显示一些不需要动态内容的静态页面。<sup>[29]</sup>系统菜单作为系统和用户的接口，用户界面分布坚持对用户友好、简便、易于操作和便于功能控制的原则。系统有许多具体功能菜单，如果控制不当被非法者使用将产生重大的安全隐患。在本系统中采取了按用户角色设定页面模板的办法，角色不同模板也不同，一般而言用户桌面只放置本角色拥有的功能菜单，然后基于模板建立网页 JSP 文件。

由于 HTTP 的无状态，实现时使用了 JSP 技术中的内置 Session 对象保存角色信息（要求用户浏览器支持 cookie 功能），即在用户登录系统时建立 Session，保存用户身份信息在 Session 中，如用户名称、角色标识等，该信息在整个会话期间有效。然后在该用户进入基于角色模板的网页前进行判断，用户的角色和目标页面允许的角色相符方可以访问，否则拒绝并返回警告信息。在用户浏览器不支持 cookie 功能也可采用会话重写方式跟踪会话，即在用户登录系统时建立一个会话跟踪字符串，保存会话跟踪字符串、用户身份信息在数据库中，如会话跟踪字符串、用户名称、角色标识等，该信息在整个会话期间有效。然后在该用户进入基于角色模板的网页前进行判断，用户的角色和目标页面允许的角色相符方可以访问，否则拒绝并返回警告信息，并将会话跟踪字符串传递到目标页面，下一次请求服务时，将会话跟踪字符串提交到服务器，并于数据库中保存的会话跟踪字符串比对，已确认身份。注销用户时从数据库中删除保存的会话信息和身份信息。

## 4.2.7 系统总体运行时序图

在用户请求流程服务之后，首先调用启动并载入工作流程，并根据流程创建一个流程实例，然后运行实例，并执行流程定义执行流程定义中 Activity（活动）中的的 Execute() 方法，Activity 的 execute() 方法使用 LocalService 相应的业务方法，LocalService 业务方法调用数据访问对象访问持久化资源，并将操作结果通过 LocalService（本地服务）返回给 Activity（活动），Activity（活动）将数据通过 WorkflowRuntime 返回给客户，同时可以传递给下一个活动，Activity（活动）执行完毕后返回执行状态给 workflow 引擎，workflow 引擎根据执行状态调度执行下一个活动。

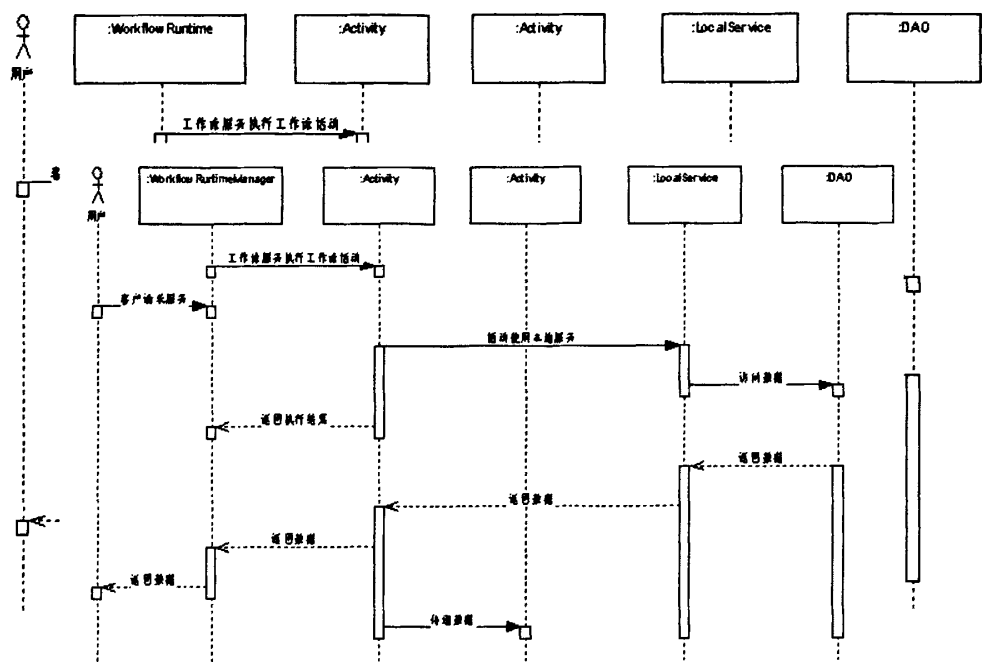


图 4-7 系统运行时序图

4.3 主要模块的设计

4.3.1 组织机构管理模块的设计

组织机构模型是以工作流为中心的协同工作系统必要的概念模型，在协同工作系统中，组织机构模型是权限、职责、信息交换以及业务活动的抽象概念集合，其中包括单位部门模型、角色模型、成员模型、权限模型等部分。业务流程中涉及部门，每个部门中有若干员工，部门与员工的关系是从属的关系，同时一个员工只能属于一个部门，一个部门可以包含多个员工。当业务员工登陆系统时，对应的是系统内部的用户，即员工与用户之间存在一对一的对应关系。用户可以具有多种角色，每一个角色都具有若干权限，同时用户也可以具有若干权限。组织机构管理模块实现了油气信息系统中的组织模型，主要涉及部门管理，角色管理，人员管理等几个方面，包括多项功能操作。[30,31,32]

(1)部门管理：完成对职能部门信息的添加，删除，修改和查询操作，以及部门与员工的关联。各部门信息中，存在部门负责人，部门领导等与业务处理流程相关的数据项，是专门为了在业务过程中实现任务指派以组为单位(部门)而设置的。

(2)角色管理：完成对系统角色信息的添加，删除，修改、查询操作。角色在被设定

权限之前，不具备任何的实用功能。

(3)用户管理：用户是油气信息系统的直接操作者和工作流程的直接参与者，其相关信息是整个系统信息数据的基础之一，用户管理包括对全部系统用户信息的添加，删除，修改和查询以及分配用户角色操作。

(4)权限管理：访问控制是系统功能操作和工作流程参与者管理的基础，它完成了油气信息管理系统中的基于角色的访问控制模型。系统访问控制权限的划分完全根据系统的功能划分，因此，能够对系统用户使用权限和操作范围有一个很好的控制。实际业务流程中，在访问控制方面可能会出现一类特殊情况，某一用户由于需要完成相应的工作，必须拥有某一特定的工作流程参与权限或系统功能操作权限，但拥有该权限的角色权限又不适于该用户，为避免在工作中越权情况的发生，只能将权限直接授予该用户。

根据以上的需求分析访问控制包括：用户角色分配、角色权限授予和用户权限授予三个部分，同时从分析结果中抽取出现在权限系统模型，其中包括:部门、用户、角色等(组织模型)以及权限、模块等(权限模型)的数据模型。以下为组织机构管理数据模型。

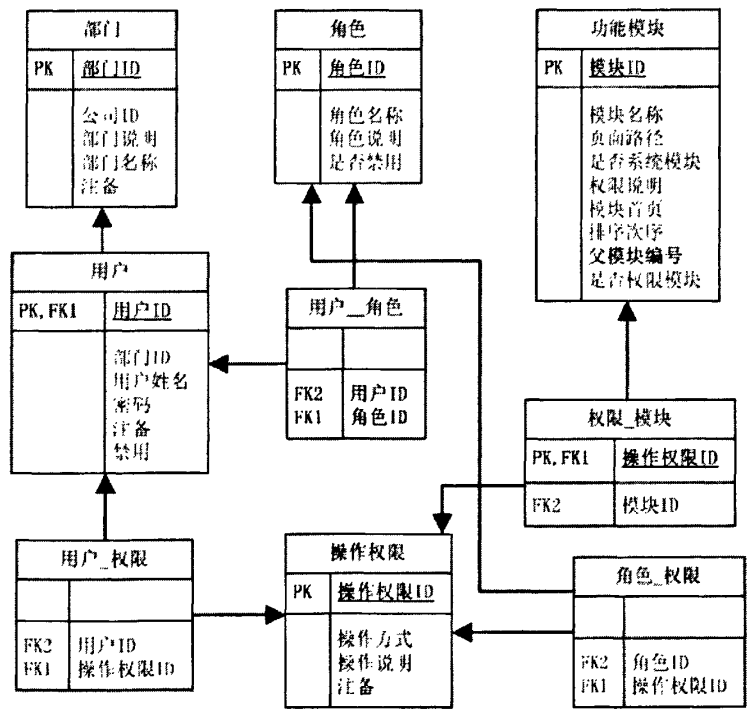


图 4-8 组织、权限管理数据模型

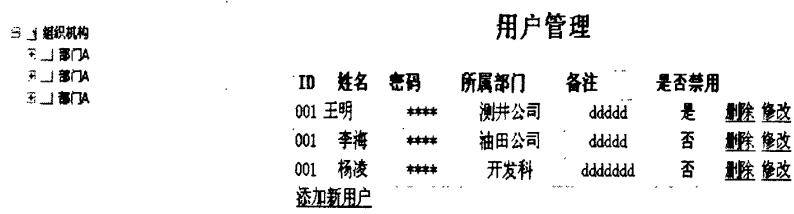


图 4-9 用户管理实现界面

4.3.2 集成服务模块的设计

解决系统集成的问题和遗留系统集成的问题采用基于 workflow 技术非常必要，它应该允许相关人员快速开发、部署新的管理信息系统、方便集成原有的遗留系统、能够适应单位不断改造业务流程的情况。基于 workflow 集成子系统分为业务集成和数据集成两部分。业务集成模块负责将现有的业务系统与油气信息系统紧密集成起来，石油行业中不同部门软件系统的整合，从而实现跨部门的应用共享。业务集成模块的主要功能是：首先对互相通讯异构系统进行数据规格化，接下来客户端与业务集成服务在转换后的数据建立通讯协议，最后与异构系统通讯协议和异构系统进行通讯。下图为基本结构图，从中可以看出油气信息系统中的某一特定流程中，某一个特定节点可以和第三方异构系统进通讯，以便于应用共享。

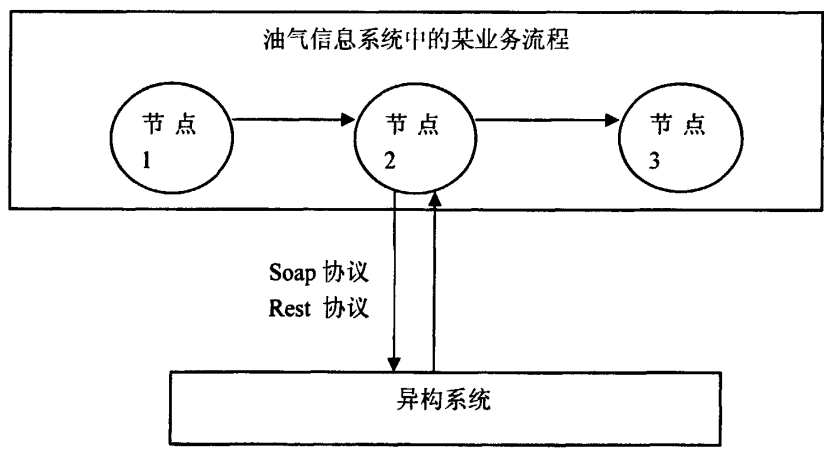


图 4-10 业务集成

业务集成模块首先对互相通讯的异构系统进行数据编码、转换以满足不同系统的数据要求，接下来转换后的数据建立通讯协议，最后在根据协议和异构系统进行通讯。在这里通讯协议可以采用 soap 或是 rest，也可以自定义通讯协议。



数据集成模块的功能就是实现企业数据从操作型数据到分析型数据的转换和多个数据源的数据集成。数据集成模块主要负责：1) 将操作型数据经过适当的筛选、转化、清洗和规格化，最后将其导入到数据仓库中；2) 对来自多个数据源头的数据进行筛选、转化、清洗和规格化最后被系统所接受。下面给出油气信息系统中一个数据集成典型流程过程，来自不同数据源的数据首先经过列筛选选取需要列，在将筛选的数据转换为适当的数据格式，第三步再将转换后的数据进行数据聚合，生成满足需求的数据，第四步则来自不同数据源处理后的数据进行数据合并，最后将合并后的数据存入合适的数据仓库中。

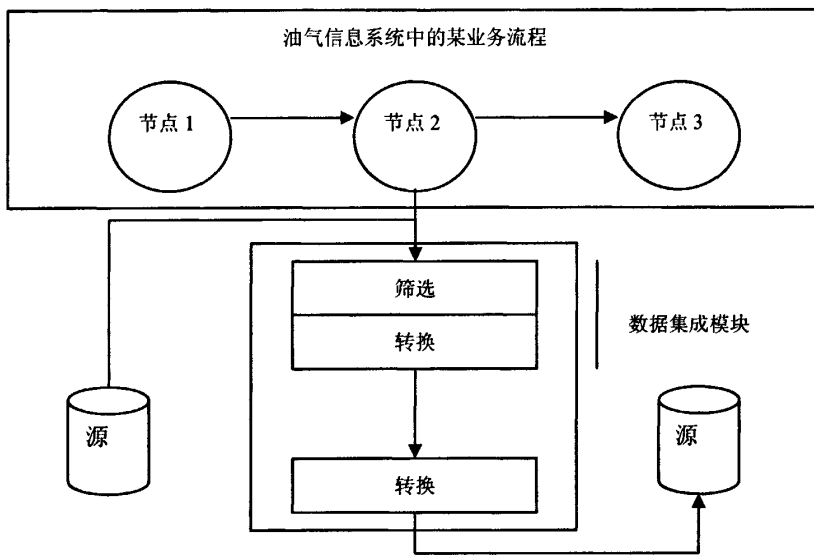


图 4-11 数据集成

数据集成模块主要负责，集成来自不同数据源的第三方数据，它主要依靠 SQLServer 集成服务提供数据集成功能，集成服务有一个配套的数据流机制和控制流机制，并且可以为数据分析服务提供必要的 ETL 支持。集成服务中，服务分成了面向流程处理的 Integration Service run-time engine 和面向数据转换的 Integration Service data flow engine，在集成服务中不仅可以完成数据转换，还可以进行数据清理(Cleaning)、聚合(Aggregating)、合并(Merging)和复制(Copying)。集成服务支持通过编程来完成转换，这大大增强了数据转换的控制能力。更为重要的是可以把整个数据包作为对象来调用，这样对于解决应用中繁杂的数据提取、转换和存储工作有很大帮助。

SQL Server 集成服务的包也是由数据源连接、控制流、数据流、时间处理、包参数和配置组成的，新的集成服务中可以用图形设计器或 .Net 语言开发包，包是 Integration Service 提取、执行和保存的单元。一个最基本的包是由一个控制流和数据流组成，而且可以通过一个包模版生成，所有的开发工作可以在设计工具中可视化完成。其中，控制

流元素主要描述的了包中任务的布局，每个任务之间的优先次序和它们如何与数据源连接关联并执行的，为了方便逻辑上可以把一组相关的任务共同保存在一个容器中，可以把它们作为一个集合进行各种配置。

系统中设计了一个服务类来封装 SQLServer 集成服务，DISService 接口提供了客户需要调用的集成服务入口，他通过设置集成服务包的源数据源和目的数据源，并连接集成服务包执行实现数据集成服务。

以下为数据集成相应的类主要方法说明

```
interface DISService{
    public DataReader destDataReader(); //设置数据源
    public DataReader sourceDataReader; //设置目标源
    public Datasource destDataSource; //设置数据源
    public Datasource sourceDataReader; //设置目标源
    public void attach(string dtsx) //链接数据集成包
    public Boolean executeDts(); //执行数据集成包
}
```

#### 4.4 网络拓扑设计

现代企业信息管理系统的系统架构是基于 Internet 网络平台的分层结构，它充分利用了扩展 B/S 结构的优势，既支持办公用户在企业安全局域网内办公，又支持办公用户远程办公。本系统采用了 B/S 和 C/S 双架构，使得既可以利用 C/S 中丰富的桌面表象，又可以利用 B/S 的简单易用。系统网络拓扑结构如图：

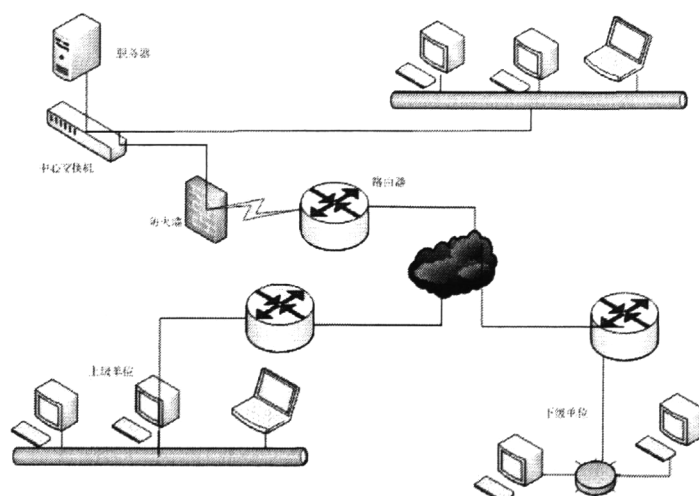


图 4-12 网络拓扑结构

企业内部用户可以访问系统应用服务器，进行业务处理，其他协作单位则可以通过系统接口处理相关业务。

#### 4.5 安全性、可靠性设计

系统安全性方面。(1)系统采用身份/口令认证，针对使用人员的工作性质为各类角色分配不同的可访问资源，采用基于角色的访问控制(RBAC)，根据用户与 workflow 节点的访问映射，为用户的各类角色分配不同的 workflow 节点访问权限；(3)对于数据库的访问，本系统中我们使用混合认证，不但能利用 Windows 认证已经集成了操作系统的安全系统，而且还可以 SQL Server 认证的安全特性。为了避免数据库中的数据被恶意的以不可预知的方式加入到数据库中或者更改数据库中的数据或者查看不被许可查看的数据，我们在数据库中对实际存储数据的表进行了限制，凡是涉及机密数据的表都被限制更改、查看、删除我们根据表和视图中存储的数据的机密程度设置各表以不同的权限。需要注意的是对表和视图的权限设置并不是越高越好，而是在满足实际需求的情况下，尽量降低权限的级别。因为权限的级别越高，系统执行时检查的约束就越多，系统的效率就越低。

可靠性方面。(1)建立严格的定期备份制度，通过系统提供的备份功能将数据库中的业务数据备份至其它媒质，在发生数据库故障情况下可以恢复。(2)配置使用服务器的磁盘阵列，进一步提高系统的可靠性。

#### 4.6 系统设计特点

(1)根据油气信息系统的特点，在设计与实现时提供给用户两种不同 workflow 运行方式，第一种适合于运行自动化任务，下一个任务的细节主要是通过当前任务的完成者在任务结束时，根据规则和约束条件进行指定来完成，在 workflow 运行中一般不需要用户介入。第二种方式下一个任务的细节主要是通过当前任务的完成者在任务结束时，根据规则和约束条件进行状态转换来完成，这种方式提供给 workflow 的参与者更大的自由度，一定程度上充分实现了柔性 workflow 特点。

(2)实用的组织机构设计。在系统设计和实现过程中，完全符合实际应用得通用标准，同时，workflow 中组织机构的定义，用户角色的划分，以及用户权限的设置，都可以灵活变化，具有相当的实用性。

(3)同时支持多种访问界面。workflow 编程接口及相关组件可以很容易地实现与 Web/client/移动的集成，通过使用不同的方式来创建过程实例，处理业务等，使用 Web 浏览器就不再需要安装客户端，可以简化系统的安装和维护，使用胖客户可以适应复杂的用户交互方式。

## 第五章 基于工作流的油气信息系统的设计与开发

传统上，油气信息中的各个模块之间存在联系，但不如办公自动化紧密，不同的业务人员处理不同的业务，这些任务的数据间存在勾稽关系，但业务上联系不明显。传统油气信息中，业务人员处理一个业务时，往往不知道自己所处理的业务与其他业务间存在勾稽关系的数据是否已经存在。因此，需要通过工作流来控制业务的执行顺序，实现油气信息系统的业务流程管理。本节以甘谷驿油矿信息管理系统中开发科子系统内的钻井及井位信息管理模块为例说明如何基于工作流思想实现业务流程管理。这个周期的环节包括：1) 固井生产单的填写环节；2) 钻井生产单的填写环节；3) 钻井及井位信息单的填写环节。本章以这三个环节来设计与开发。

### 5.1 钻井及井位信息管理模块业务分析

在钻井及井位信息生成中业务的基本环节和步骤如下：

- 1) 固井业务员填写固井生产信息单，钻井生产所需来自固井的勾稽关系信息满足的情况下，保存固井生产信息；
- 2) 钻井业务员填写钻井生产信息单，钻井及井位所需来自钻井的勾稽关系信息满足的情况下，保存钻井生产信息；
- 3) 钻井及井位业务员填写钻井及井位信息单，保存钻井及井位信息；

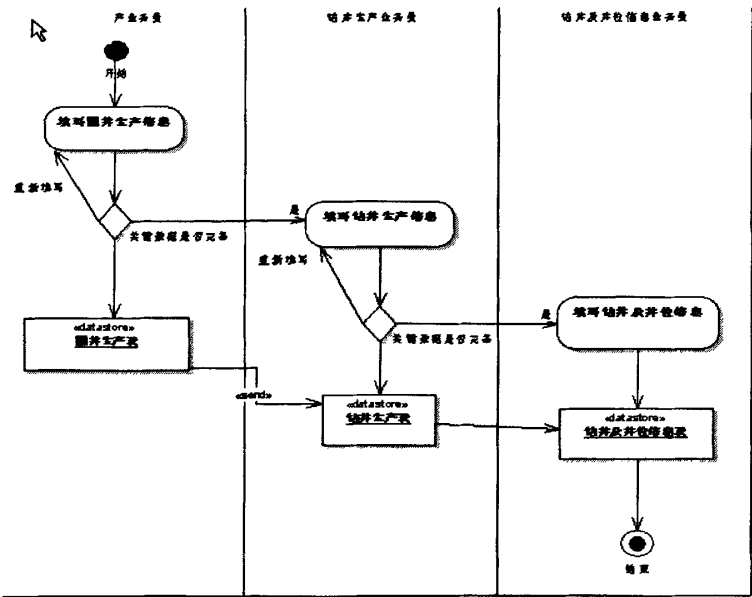


图 5-1 业务流程

## 5.2 数据库设计

以下为业务相关的数据库模型，用来保存各个环节的数据。本章所实现的业务流程中需要存储的数据包括：固井生产表、钻井生产表和钻井及井位表。以下定义三个表的表结构。

钻井及井位	钻井	固井
<p>column</p> <p>校正坐标X: float</p> <p>校正坐标Y: float</p> <p>当前井深: float</p> <p>地层位置: nvarchar(255)</p> <p>井别: nvarchar(255)</p> <p>井口海拔H: float</p> <p>井口坐标X: float</p> <p>井口坐标Y: float</p> <p>井名: float</p> <p>井型: nvarchar(255)</p> <p>开始日期: smalldatetime</p> <p>结束: float</p> <p>人工井底: float</p> <p>套管外筒长度: float</p> <p>元井方式: nvarchar(255)</p> <p>元井日期: smalldatetime</p> <p>元井坐标: nvarchar(255)</p> <p>元井井深: float</p> <p>元井日期: smalldatetime</p> <p>钻机型号: nvarchar(255)</p> <p>钻井单位: nvarchar(255)</p>	<p>column</p> <p>备注: nvarchar(255)</p> <p>套管使用长度(米): nvarchar(255)</p> <p>套管下深(米): float</p> <p>校正器个数: nvarchar(255)</p> <p>校正器使用长度(米): nvarchar(255)</p> <p>校正器下深(米): float</p> <p>甲方负责人: nvarchar(255)</p> <p>井别: nvarchar(255)</p> <p>井名: float</p> <p>井型: nvarchar(255)</p> <p>开始日期: smalldatetime</p> <p>结束: float</p> <p>人工井底(米): nvarchar(255)</p> <p>设计标准套管井深(米): nvarchar(255)</p> <p>设计水层深度(米): nvarchar(255)</p> <p>设计期望井位深度(米): nvarchar(255)</p> <p>实际标准套管井深(米): nvarchar(255)</p> <p>实际水层深度(米): float</p> <p>实际期望井位深度(米): float</p> <p>套管使用日期: smalldatetime</p> <p>元井日期: smalldatetime</p> <p>元井井深(米): float</p> <p>元井日期: smalldatetime</p> <p>乙方负责人: nvarchar(255)</p> <p>钻井单位: nvarchar(255)</p>	<p>column</p> <p>备注: nvarchar(255)</p> <p>套管长度(M): float</p> <p>套管下深(M): float</p> <p>校正器个数(个): float</p> <p>固井日期: smalldatetime</p> <p>固井质量: nvarchar(255)</p> <p>甲方负责人: nvarchar(255)</p> <p>井别: nvarchar(255)</p> <p>井名: nvarchar(255)</p> <p>井型: nvarchar(255)</p> <p>井生产#标准管位置1(M/M): float</p> <p>井生产#标准管位置2(M/M): float</p> <p>井位: nvarchar(255)</p> <p>井型: nvarchar(255)</p> <p>结束: float</p> <p>人工井底(M): nvarchar(255)</p> <p>水层深度(米): float</p> <p>水层深度(米): float</p> <p>套管内径(mm): float</p> <p>乙方负责人: nvarchar(255)</p> <p>固井(M): float</p>

图 5-2 数据模型结构

## 5.3 数据流分析

业务中，各个环节的数据存在一定的勾稽关系，钻井及井位信息生成中，钻井信息中的某些部分来源于固井生产信息，同时还需要用户干预输入其他的信息。而油气井的钻井及井位则来源于钻井生产信息数据，这样就形成了如下图所示的信息产生流程。

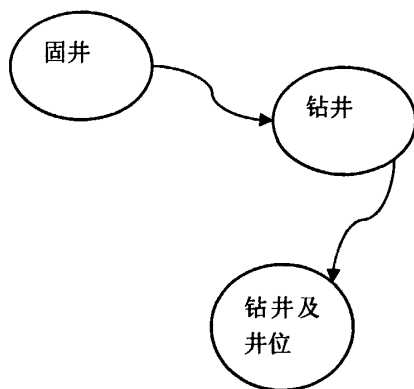


图 5-3 数据流

下面对各个环节的勾稽关系加以说明。

钻井生产表的井名、区块、井别、井型、[人工井底(米)]、[扶正器个数(个)]、[实测水泥返深(米)]来自固井生产表的井名、区块、井别、井型、[人工井底(米)]、[扶正器个数(个)] [水泥返深(米)]产信息。

钻井及井位信息的井名、区块、井别、井型钻井单位、开钻日期、完钻日期、完钻井深、完井日期来自钻井生产表的井名、区块、井别、井型、钻井单位、开钻日期、完钻日期、完钻井深、完井日期。

## 5.4 流程的定制

工作流是一个有穷自动机，它只维护流程的运转，即由一个状态跳转到另外一个状态，但是由谁在什么情况下执行何种任务构成流程的跳转，工作流仅仅留出了接口，在实际流程中需要开发者自己定制。为了实现业务流程，这里我们定义工作流描述，业务中我们以状态机工作流来定义业务的流程。固井生产环节，业务人员保存固井信息后，业务流程流转到钻井生产；钻井生产环节，业务人员保存钻井信息后，业务流程流转到钻井及井位信息；钻井及井位信息信息保存后，流程结束。

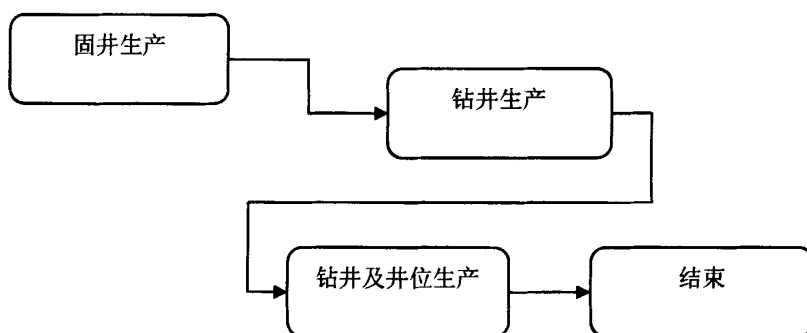


图 5-4 流程状态转换图

定义的流程描述文件如下。

```
<Workflow name="钻井及井位" init="填写固井生产单" finish="填写钻井及井位信息">
```

```
<StateActivity state="填写固井生产单">
```

```
<IfElseActivity>
```

```
<IfElseBranchActivity>
```

```
<![CDATA[
```

```
//检查勾稽关系数据是否完备。井名、区块、井别、井型、人工井底、扶正器个数、水泥返深信息检查
```

```

        @gjsc. jm@!=null&&@ zjsc. qk@!=null&&@ gjsc. jb@!=null&&
        @gjsc. jx@!=null&&@
        gjsc. rgjd@!=null&&@gjsc. fzqgs@!=null&&@gjsc. snfs@!=null
    ]]>
    <CodeActivity classname = "NewGjscActivity" />
    <SetStateActivity TargerState="填写钻井生产单" />
</IfElseBranchActivity>
<IfElseBranchActivity>
    <SetStateActivity TargerState="填写固井生产单" />
</IfElseBranchActivity>
</IfElseActivity>

</StateActivity>
<StateActivity state="填写钻井生产单">
    <IfElseBranchActivity>
        <![CDATA[
            <![CDATA[
                //检查勾稽关系数据是否完备。井名、区块 、井别 、井型钻井单位、
                开钻日期 、完钻日期、完钻 井深、完井日期信息检查
                @zjsc. jm@!=null&&@ zjsc. qk@!=null&&@ zjsc. jb@!=null&&@
                zjsc. jx@!=null&&@ zjsc. kzrq@!=null&&@ zjsc. wzrq@!=null&&
                @wzjs@!=null&&@ zjsc. wzjs@!=null&&@ zjsc. wjrj@!=null
            ]]>
            <CodeActivity classname = "NewZjscActivity" />
            <SetStateActivity TargerState="填写钻井及井位信息" />
        </IfElseBranchActivity>
        <IfElseBranchActivity>
            <SetStateActivity TargerState="填写钻井生产单" />
        </IfElseBranchActivity>
    </StateActivity>
<StateActivity state="填写钻井及井位信息">
    <CodeActivity classname = "NewZjandJwActivity" />
</StateActivity>
</Workflow>

```

## 5.5 活动的实现

接下来我们以钻井生产环节来考察业务。钻井业务员登录后，提示出现新井的钻井作业生产单任务，钻井业务员开始填写钻井信息后，保存填写信息时，触发一个钻井信息保存事件，被保存的数据通过钻井信息保存事件参数发送到工作流中，工作流执行一个事件处理方法，调用本地服务的保存钻井生产信息方法，保存工作流中的钻井生产数据，然后流程状态转换到钻井及井。

```
<StateActivity state="填写钻井生产单">
    <IfElseBranchActivity>
        <![CDATA[
            <![CDATA[
                //检查勾稽关系数据是否完备。井名、区块、井别、井型钻井单位、
                开钻日期、完钻日期、完钻 井深、完井日期信息检查
                @jm@!=null&&@qk@!=null&&@jb@!=null&&@jx@!=null&&
                @kzrq@!=null&&@wzrq@!=null&&@wzjs@!=null&&@wzjs@!=null&&@wjrq@!=null
            ]]>
                <CodeActivity classname="NewZjscActivity" />
                <SetStateActivity TargerState="填写钻井及井位信息" />
            </IfElseBranchActivity>
            <IfElseBranchActivity>
                <SetStateActivity TargerState="填写钻井生产单" />
            </IfElseBranchActivity>
        </StateActivity>
```

这里我们需要定制一个代码活动来完成钻井信息保存的实际动作。NewZjscActivity 代码说明：

```
public class NewZjscActivity extends Code {
    protected void execute_code(BookmarkManager mgr) {
        //获取本地服务
        KfkService kfkService = (KfkService) mgr.getService(KfkService.class);
        //调用本地服务的方法
        try{
            kfkService.newZjsc(zjsc);
        }catch(SQLException e){
```



```
}  
//迁移到活动完成状态  
mgr. done();  
}  
}
```

为了实现整个业务流程需要用到以下的自定义活动

表 5-1 自定义活动表

活动类型	功能描述	参数
NewGjscActivity	保存固井生产信息	要跟新的值
NewZjscActivity	保存钻井生产信息	要跟新的值
NewZjandJwActivity	保存钻井及井位信息	要跟新的值

5.6 运行界面

用户以固井业务员身份登录，进入操作台后，可以发起一个新流程，开始填写某口井的固井生产单。

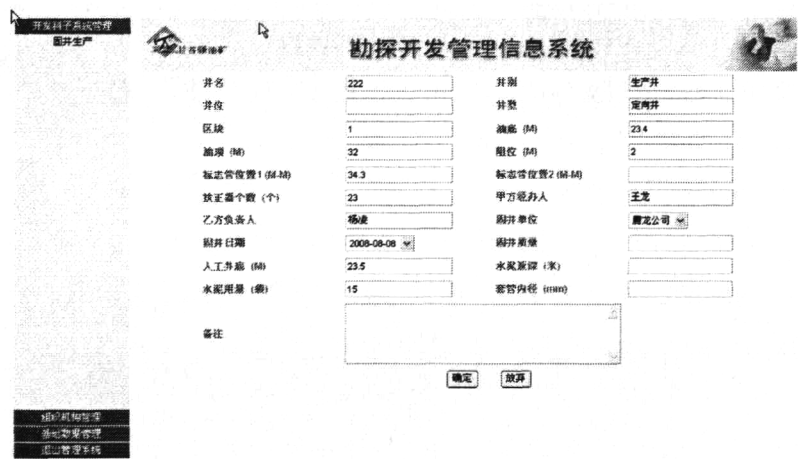


图 5-5 固井生产单录入

用户以钻井业务员身份登录，进入操作台后，可以看到有一个提示信息，提示可以开始某口井的钻井生产信息填写。

任务编号	井名	开始钻井生产任务
1	222	<input type="button" value="开始"/>
2	221	<input type="button" value="开始"/>

图 5-6 钻井生产任务

选择相应井名，点击开始继续流程，并转到钻井信息填写表单。

勘探开发管理信息系统			
井名	<input type="text" value="222"/>	井别	<input type="text" value="生产井"/>
井位	<input type="text" value="1"/>	井制	<input type="text" value="定向井"/>
区块	<input type="text" value="1"/>	校正垂深(米)	<input type="text" value="23"/>
甲方经办人	<input type="text"/>	乙方负责人	<input type="text"/>
表座下偏(米)	<input type="text"/>	接管下偏(米)	<input type="text"/>
开钻日期	<input type="text" value="2008-08-08"/>	人工升隔(米)	<input type="text" value="23.5"/>
完井日期	<input type="text" value="2008-08-08"/>	完钻升隔(米)	<input type="text"/>
完钻升隔(米)	<input type="text"/>	完钻日期	<input type="text" value="2008-08-08"/>
钻井单位	<input type="text" value="巨龙公司"/>	钻靶器使用长度(米)	<input type="text"/>
钻靶器使用长度(米)	<input type="text"/>	钻进计划管井段(米)	<input type="text"/>
钻设计水泥浆量(米)	<input type="text"/>	钻设计粗砂环空量(米)	<input type="text"/>
钻实测标志管井段(米)	<input type="text"/>	钻实测水泥浆量(米)	<input type="text"/>
钻实测粗砂环空量(米)	<input type="text"/>	钻套管使用日期	<input type="text" value="2008-08-08"/>
备注	<input type="text"/>		

图 5-7 钻井生产单录入

用户以钻井及井位业务员身份登录，进入操作台后，可以看到有一个提示信息，提示可以开始某口井的钻井及井位信息填写。

任务编号	井名	开始钻井及井位任务
1	222	<input type="button" value="开始"/>
2	221	<input type="button" value="开始"/>

图 5-8 钻井及井位任务

选择相应井名，点击开始继续流程，并转到钻井及井位信息填写表单。

开发科学系统管理		钻井及井位	
 订单编号: 00000000000000000000			
勘探开发管理信息系统			
井名	<input type="text" value="221"/>	井别	<input type="text" value="直井"/>
井位	<input type="text"/>	井组	<input type="text" value="采油井"/>
区块	<input type="text" value="1"/>	基本客套外圈长度	<input type="text"/>
靶点坐标X	<input type="text"/>	靶点坐标Y	<input type="text"/>
曲轴半径	<input type="text" value="750"/>	地理位置	<input type="text"/>
井口海拔H	<input type="text"/>	井口倾斜M	<input type="text"/>
井口坐标V	<input type="text"/>	开钻日期	<input type="text" value="2009-08-09"/>
人工井底	<input type="text" value="2500"/>	完井方式	<input type="text" value="2009-08-09"/>
完钻井深	<input type="text" value="8730"/>	完钻层位	<input type="text" value="请选择数据"/>
完钻日期	<input type="text" value="2009-08-09"/>	钻机型号	<input type="text" value="请选择数据"/>
钻井单位	<input type="text" value="震龙公司"/>		
		<input type="button" value="确定"/>	<input type="button" value="返回"/>

图 5-9 钻井及井位录入

## 第六章 总结与展望

workflow 技术是计算机支持的协同工作与集成研究领域一门新出现的学科, 基于自定义业务流程的 workflow 管理系统的研究对于企业的业务流程重组、协同技术和应用与数据集成的发展具有十分重要的意义。 workflow 技术所要解决的正是企业业务过程建模领域的问题, 将 workflow 管理的思想和技术引入油气信息管理系统, 不仅可以解决企业业务过程管理、优化业务流程存在的问题, 提高系统的柔性, 而且能够实现与其他业务管理系统, 如办公自动化系统、数据集成服务等集成。

目前, 油气信息系统中引入 workflow 技术的研究与应用在我国尚处于起步阶段, 虽然已有研究人员从不同的角度开展了 workflow 技术应用与信息开发的研究, 并取得了一些成功的案例, 但与油气信息结合案例还比较少。为了满足石油行业的实际需求, 必须结合计算机领域较新技术的基础上, 吸取国内外最新研究成果的基础上将 workflow 技术引入我国石油行业, 开发基于 workflow 的管理信息系统。

本文研究了 workflow 技术领域中的重要技术: workflow 建模、 workflow 管理系统体系结构等, 并基于 workflow 技术开发了一个轻量级 workflow 引擎, 同时在工作流相关理论技术的研究基础上和对油气信息管理系统应用中存在的工作流模型的分析, 将可以实现自定义工作流程的 workflow 技术和油气管理信息系统有机的结合在一起, 实现了一套基于 workflow 技术的, 可柔性定义企业业务流程和集成第三方业务系统的具备高度可扩展的管理信息系统, 通过实践充分说明了应用中采用 workflow 技术的优势。

本文主要完成了以下几方面的工作:

- 1) 分析 workflow 技术与油气信息系统的关系, 研究 workflow 技术在油气信息系统的应用。
- 2) 以 workflow 技术理论为基础, 设计了一个适用于油气信息系统的轻量级 workflow 引擎。
- 3) 设计一种基于 workflow 技术的油气信息系统的体系结构, 并基于所设计的体系结构, 结合现代企业单位在信息化过程中所遇到的具体问题, 详细分析并设计实现了的油气信息系统。

workflow 管理技术综合了计算机科学和管理科学中多个研究领域的原理、方法和技术。随着计算机网络、分布式计算、数据库及数据仓库等技术的不断发展, workflow 管理技术作为现代化企业实现过程管理与过程控制的一项关键技术必将得到更大的发展, 基于 workflow 技术的油气管理信息系统也必将取代传统的油气管理信息系统。随着石油行业的进一步与国际接轨, 石油行业将真正进入全球竞争时代, 油气信息化基础设施将进一步完善。软硬件基础设备将进一步增强, 信息网络覆盖全系统并进一步扩充网络功能, 使油气信息化将继续进一步加深应用。因此, 研究具有先进技术的适应石油行业发展需要的管理信息系统, 使油气信息化真正为石油行业的可持续发展服务是具有深远的理论意义

和广阔的市场应用前景的。

针对 workflow 技术在油气信息协同与集成系统中的应用，本文在 workflow 建模，引擎设计、实施和管理等方面进行了研究和实现，虽然取得了一定的成果，但是，由于时间和精力有限，目前的工作尚有不足之处，有待下一步工作的完善。

## 致 谢

在本文即将结束之际，我要衷心地感谢多年来关心和支持我的所有人！

在论文完成之际，衷心感谢我的导师方明教授，他是一位具有创新精神的科学工作者，也是一位出色的指导老师。他在各方面给了我很多关心和指导，并且给我提供了一个良好的发展空间，使我在理论研究与实际开发上都取得了很大的进步。从方老师身上，我不仅仅学到了知识，更多的是学到了做人和做学问的道理，这将成为我一生宝贵的财富。

在我攻读硕士学位期间，得到计算机学院诸位同学的指导和帮助，使我受益非浅。在此向那些关心和帮助过我的老师和同学表示最衷心的感谢！

最后还要感谢我的家人，你们给了我无私的帮助、关怀和爱，给了我生活中无尽的乐趣和温馨，你们是我人生和事业前进的动力，是我永远最珍爱的人。

我还要感谢在此不能一一列举的所有关心我、帮助过我的老师、同学、朋友和家人们，感谢你们对我的帮助，祝你们身体健康，永远快乐！

## 参考文献

- [1] 薛华成.管理信息系统.第三版[n2].北京:清华大学出版,1999:20~50
- [2] 朱学伟, 孙奎, 路鹏, 殷兆麟. workflow模式的概述[D].福建电脑,2007.12:30~31
- [3] MohanC, MVoorhoeve. Recenttrendsinworkflow management products[J]. standardsandresearch.InformationSystems,1997.95(4):127~131.
- [4] 顾小钧, 柳盛;基于 workflow技术的管理信息系统的设计与实现[J]. 中国科技信息, 2008.6(52):56~58
- [5] 罗海滨, 范玉顺, 吴澄.workflow技术综述[D].软件学报,2000.11(7):899~907
- [6] 田熙清等. 基于 workflow技术的项目管理系统的分析和设计[J]. 计算机工程与应用,2003.39(8):131~134
- [7] 林彤, 张载鸿. workflow系统中过程模型的应用研究[J]. 计算机应用, 2002.2 (6):3~18.
- [8] 范玉顺.workflow管理技术基础.北京:清华大学出版社, 2001: 31~36
- [9] DBob Schmidt. Essential Windows Workflow Foundation [M]. 北京: 机械工业出版社,2007:22~35
- [10] 赵刚, 杨宗凯.基于 workflow和 Web 技术的 OA 系统设计.计算机工程与应用, 2002.9: 235~238
- [11] 谭慧蓉.办公自动化的发展趋势及策略.太原科技, 2003(5):10~11
- [12] 毛宇,罗中先. 戴跃洪;基于服务的企业应用集成[J] 纺织机械, 2008.1:34~36
- [13] 康一梅; 企业应用集成中流程集成模型的研究[J]. 计算机工程与应用, 2007.12(66): 243~246
- [14] Wittner, Susan, Wood, Chris. Business integration Gatfworld[J]. Lecture Notes in Computer Science. 2006.2(18):24
- [15] Rodiono, I.B. Integration of monitoring data in multiagent information systems[J]. Computer Science, 2005.6(64):437~445
- [16] Tomas. SQL Server 2005 集成服务高级编程[M]. 北京:清华大学出版社, 2007:20~40
- [17] 朱海平, 肖诗旺, 车争. 支持业务流程分析的工作流仿真研究[J]. 小型微型计算机系统, 2006.8(27):1531~1535
- [18] Jobs Migrating. Application Integrations[M]. Springer Berlin / Heidelberg, 2004.5 94-103
- [19] Christoph Bussler. The Application of Workflow Technology in Semantic B2B Integration[J]. Distributed and Parallel Databases, 2002.9(12):2~3
- [20] 周坤等. 面向 Web Services 动态复合的流程自动化系统的研究与实现[J]. 计算机应用, 2005.1(25):85~87
- [21] 范玉顺. workflow管理技术基础—实现企业业务过程重组. 过程管理与过程自动化的

核心技术[M]. 北京:清华大学出版社,2001:120~128

[22] 李红臣, 史美林, 陈信祥. workflow系统中的业务过程描述及分析[J]. 计算机研究与发展, 2001. 38(7):798~804

[23] 张晶, 张云生, 向凤红. 分层异构控制软件的构件化建模和设计[J]. 计算机工程, 2008.4(43):93~95

[24] 李明. 基于三层结构的系统开发方法与应用研究. 河海大学硕士学位论文 2004:30~42

[25] Grady Booch. 面向对象分析与设计. 北京:机械工业出版社, 2003:140~173

[26] 范玉顺, 曹军威. 复杂系统的面向对象建模、分析与设计. 北京:清华大学出版社. 施普林格出版社, 2000:239~280

[27] 于松涛, 徐振成. 精通 SQL Server 2000 数据库管理与开发[M]. 人民邮电出版社, 2001:3~22

[28] 张砚秋, 陈川, 何明德. 基于 MVC 设计模式构筑 JSP/Servlet+EJB 的 Web 应用[J]. 计算机工程, 2001.27(11):71~74.

[29] 张波, 张福炎. 基于 JSP 技术的 Web 应用程序开发[J]. 计算机应用研究, 2001.18(5):99~102

[30] 张晓光, 曹健, 张申生. 策略约束面向角色和团队关系的工作流任务分配管理 计算机研究与发展[J], 2002.39(12):1156~1163

[31] 刘建勋, 张申生, 步丰林. 基于角色访问控制在 workflow 管理系统中的应用研究[J]. 小型微型计算机系统, 2003.24(6):1067~1070

## 附 录

### 攻读硕士学位期间所发表的论文

- [1] 王锬 WEB 文档信息抽取方法研究王锬 福建电脑 2008.3:133~134
- [2] 方明, 王锬 Ajax 技术研究与应用 现代电子技术 2008.3:93~97