

疯狂

Workflow i#X

基于Activiti的工作流应用开发

杨恩雄 编著

疯狂源自梦想
技术成就辉煌



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

流程

Workflow 讲义

基于Activiti的工作流应用开发

杨恩雄 编著

电子工业出版社

Publishing House of Electronics Industry

内 容 简 介

workflow 框架 Activiti 是由 jBPM4 发展而来的, 为了能帮助广大的 Java 研发者学习该框架, 并将其应用到实际的生产中, 笔者根据多年来的企业应用研发经验, 编写了本书。

书中首先对 Activiti 的基础知识进行讲解, 并从源代码的实现上讲解 Activiti 各个模块的 API。此外详细讲述了 BPMN 2.0 规范的内容, 包括目前 Activiti 对该规范的实现情况。书中还讲解了如何在实际企业应用中使用 Activiti, 并与其他流行的开源框架进行整合, 包括企业应用开发所必需的 Web Service、企业服务总线、规则引擎、IoC 框架和 ORM 框架等。全书的最后通过讲解一个办公自动化系统的开发过程, 让读者更了解 Activiti 在实际生产中的使用, 让本书的内容不仅仅停留在理论的层次, 更为读者开辟了实践的道路。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有, 侵权必究。

图书在版编目 (CIP) 数据

疯狂 Workflow 讲义: 基于 Activiti 的工作流应用开发 / 杨恩雄编著. —北京: 电子工业出版社, 2014.5
ISBN 978-7-121-22673-1

I. ①疯… II. ①杨… III. ①J214.21—44 ②F407.6 ③TP312

中国版本图书馆 CIP 数据核字



责任编辑: 张月萍

特约编辑: 梁卫红

印 刷: 北京京科印刷有限公司

装 订: 三河市皇庄路通装订厂

出版发行: 电子工业出版社

北京市海淀区万寿路173信箱 邮编: 100036

开 本: 787×1092 1/16

印张: 28.25 字数: 746千字

印 次: 2014年5月第1次印刷

定 价: 73.00元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。



前言

随着 Java 成为世界上应用最广泛的编程语言,在 Java 领域出现了众多优秀的框架以及组件,这些组件正在慢慢提高编程的效率,使得编程这项原来枯燥的工作变得更为优雅与简单。在工作流领域,涌现出多个使用 Java 语言编写的框架,如 OpenWFE、jBPM、Shark,甚至在国内市面上出现了不少国产的工作流引擎。在近一两年的工作流领域中,出现了一款“全新”的工作流框架 Activiti,该框架是由 jBPM4 发展而来的。Activiti 一经推出,马上得到开源社区的大力支持。为了能帮助广大的 Java 研发者学习该框架,并将其应用到实际的生产中,笔者根据多年来的企业应用研发经验,编写了本书。

笔者在进行企业应用开发的早期接触了一些 OA 系统,当时这些系统存在大量的工作流程,这些流程夹杂着不同领域的业务,由于当时并没有使用任何的工作流组件,因此进行研发和维护的过程非常痛苦。一旦发生业务流程上的变化,修改源代码成为家常便饭,整个项目举步维艰,而需求的变更则是无处不在,业务流程的变化不可避免。从那时起,笔者思索着实际上可以将这些流程独立开来,作为一个应用组件存在,希望以这样的方式来适应需求的多变,最终可以达到当需求变更时,不需要修改或者少改源代码,这是笔者最初的梦想。可惜后来系统由于修改过于困难,已经完全不能适应业务的变化而被抛弃,笔者也随之离职。加入新的公司(中企动力)后,有幸参与一个人事系统的研发工作,该系统包括一些常规的 OA 功能,例如薪资、绩效、考勤等。当时公司有着一批技术前瞻性很强的同事,选择了当时较为流行的 jBPM 工作流引擎来实现 OA 的流程,也选用了 JBoss 的 Drools 规则引擎计算员工薪资,这是笔者第一次接触工作流引擎。现在回想起来,待在这个项目组与大家一起奋斗,真的无比开心,不仅仅能在工作中学到大量的技术,还与各个同事成为好朋友,这份回忆对笔者来说是一份珍贵的财富。

在接触工作流的框架后,笔者慢慢体会到这些工作流框架与组件给开发者带来的乐趣,当业务流程发生变化时,几乎不需要修改代码,就可以实现系统的变更,对于之前受尽业务变化折磨的我来说,这是一件十分开心的事。笔者有一个朋友,同样在另外一家专门做工作流应用的公司工作,当我第一次接触到他们公司的产品,感觉到工作流应用可以做得更为强大。他们的产品叫作 OBPM,是一款国产的工作流引擎,使用该产品,可以直接通过界面来定制业务流程、设置表单内容,甚至可以在上面进行数据库设计,几乎不用编写一行代码,一个简单的业务系统即可完成。这种高效的开发平台对笔者触动很大,在往后的职业生涯中,一直致力于提高研发的效率。

本书内容概括

本书是一本介绍 Java 工作流领域的书,以 Activiti 为核心,内容囊括了多个流行的企业级 Java EE 框架,全书主要可分为以下几个部分。

第 1 部分:对 Activiti 的基础知识进行讲解,包括框架起源、基本的设计模式、数据库设置以及框架配置等,该部分知识可以帮助读者对 Activiti 的基础有一个更深入的认识,对 Activiti 的设计有一个初步的印象,该部分内容也可以作为整合 Activiti 到项目中的参考。

第 2 部分:从源代码的实现上讲解 Activiti 各个模块的 API,除了讲述这些 API 的作用外,还会引领读者深入到这些 API 的内部,此部分内容可以作为一份详细的 Activiti API 的帮助文档。

第3部分：详细讲述了 BPMN 2.0 规范的内容，包括目前 Activiti 对该规范的实现情况，在讲解 BPMN 2.0 规范时，将规范与 Activiti 的实现进行结合，在通俗易懂的案例下，帮助读者对 Activiti 的实现以及 BPMN 2.0 规范有更深入的了解。

第4部分：如何在实际企业应用中使用 Activiti，并与其他流行的开源框架进行整合，包括企业应用开发所必需的 Web Service、企业服务总线、规则引擎、IoC 框架和 ORM 框架等，除了本书主要的 Activiti 知识外，这些企业中常用框架的知识，也可以在该部分内容得到体现，通过学习这部分内容，可以极大丰富读者的实战知识，让读者可以成为一个更全面的技术人员。

第5部分：在第17章中讲述了 Activiti 的核心架构流程虚拟机以及 Activiti 的表单知识，本书最后一章，通过讲解一个办公自动化系统的开发过程，让读者更了解 Activiti 在实际生产中的使用，让本书的内容不仅仅停留在理论的层次，更为读者开辟了实践的道路。

本书特点

笔者长期工作于企业的 IT 部门，有着丰富的企业应用开发经验，因此本书有着以下特点。

1. 内容深入

从笔者接触编程开始，就养成了查看源代码的习惯，书中的案例不仅仅讲解 Activiti 的功能，更模仿 Activiti 的思路去模拟功能的实现，读者可以深入了解其中的原理。

2. 开发环境与案例更贴近实际

书中案例的开发环境、使用的框架以及工具均是来自企业的实际应用，案例的选取与研发过程更贴近实际。

3. 注释详细

本书的代码，几乎每行核心 Java 代码均有注释，读者可以很容易地了解代码的意思，轻松掌握相应的知识。

本书写给谁看

如果你有一定的 Java 语言基础，进行过 Web 项目的开发，对工作流有一定的认识，那么本书可以帮助你提升工作流的知识。如果你是一个从事过企业应用开发的程序员，本书同样适合，本书的知识可以帮助你深入工作流引擎，你可以将这些工作流框架应用到实际的企业生产中。读者在阅读本书的过程中如果有需要交流的问题，可以发邮件至 yangenxiong@163.com。

衷心感谢

首先非常感谢李刚老师，一直以来，他既是我的老师，也是我的技术后盾。刚进入行业的早期，遇到很多不会解决的问题，总会找到李刚老师一起讨论，在和他的讨论中，不经意间，总会找到我想要的答案。随着自己的成长，现在找李刚老师聊的更多的是人生、家庭，非常幸运人生能有这样一位良师益友。

此外感谢我公司的同事，是你们长期与我奋斗在企业第一线，为客户解决问题，累积个人经验，你们也是我不停前进的动力。

杨恩雄

2014年2月20日于广州

目 录 CONTENTS

第 1 章 Activiti 介绍	1	3.3.4 关系表	22
1.1 工作流介绍	2	3.4 运行时数据表	23
1.2 BPMN 2.0 规范简述	2	3.4.1 流程实例 (执行流) 表	23
1.2.1 BPMN 2.0 概述	3	3.4.2 流程任务表	23
1.2.2 BPMN 2.0 元素	3	3.4.3 流程参数表	24
1.2.3 BPMN 2.0 的 XML 结构	4	3.4.4 流程与身份关系表	24
1.3 Activiti 介绍	4	3.4.5 工作数据表	24
1.3.1 Activiti 的出现	5	3.4.6 事件描述表	25
1.3.2 Activiti 的发展	5	3.5 历史数据表	25
1.3.3 选择 Activiti 还是 jBPM	5	3.5.1 流程实例表	25
1.4 本章小结	5	3.5.2 流程明细表	26
第 2 章 安装与运行 Activiti	6	3.5.3 历史任务表和历史行为表	26
2.1 下载与运行 Activiti	7	3.5.4 附件表和评论表	26
2.1.1 下载和安装 Ant	7	3.6 本章小结	27
2.1.2 下载和安装 Maven	8	第 4 章 Activiti 流程引擎配置	28
2.1.3 下载和安装 Activiti	9	4.1 ProcessEngineConfiguration	29
2.1.4 运行第一个 Activiti 应用	9	4.1.1 createProcessEngineConfigurationFromResourceDefault 方法	29
2.2 安装开发环境	11	4.1.2 createProcessEngineConfigurationFromResource 方法	30
2.2.1 下载 Eclipse	11	4.1.3 createProcessEngineConfigurationFromInputStream 方法	30
2.2.2 安装 Activiti 插件	11	4.1.4 createStandaloneInMemProcessEngineConfiguration 方法	30
2.2.3 获取 Activiti 源代码	13	4.1.5 createStandaloneProcessEngineConfiguration 方法	31
2.3 编写第一个 Activiti 程序	15	4.2 数据源配置	31
2.3.1 如何运行本书案例	15	4.2.1 Activiti 支持的数据库	32
2.3.2 建立工程环境	15	4.2.2 Activiti 与 Spring	32
2.3.3 创建配置文件	16	4.2.3 JDBC 配置	32
2.3.4 创建流程文件	16	4.2.4 DBCP 数据源配置	33
2.3.5 加载流程文件与启动流程	17	4.2.5 C3P0 数据源配置	34
2.4 本章小结	18	4.2.6 Activiti 其他数据源配置	35
第 3 章 Activiti 数据库设计	19	4.2.7 databaseSchemaUpdate 配置	35
3.1 通用数据表	20	4.2.8 databaseType 配置	36
3.1.1 资源表	20	4.3 其他属性配置	37
3.1.2 属性表	20	4.3.1 history 配置	37
3.2 流程存储表	21	4.3.2 jobExecutorActivate 配置	38
3.2.1 部署数据表	21	4.3.3 邮件服务器配置	38
3.2.2 流程定义表	21	4.4 ProcessEngineConfiguration bean	38
3.3 身份数据表	21	4.4.1 ProcessEngineConfiguration 及其	
3.3.1 用户表	21		
3.3.2 用户账号 (信息) 表	22		
3.3.3 用户组表	22		

子类	38	6.3.5 验证用户密码	69
4.4.2 自定义 ProcessEngineConfiguration	39	6.3.6 用户数据查询	70
4.5 Activiti 的命令拦截器	41	6.3.7 设置用户权限	71
4.5.1 命令模式	41	6.4 用户信息管理	72
4.5.2 责任链模式	43	6.4.1 添加和删除用户信息	72
4.5.3 编写自定义拦截器	45	6.4.2 查询用户信息	73
4.6 本章小结	47	6.4.3 添加与删除用户账号	74
第 5 章 流程引擎的创建	48	6.4.4 查询用户信息	75
5.1 ProcessEngineConfiguration 的 buildProcessEngine 方法	49	6.4.5 设置用户图片	76
5.2 ProcessEngines	49	6.5 用户组与用户关系	77
5.2.1 init 与 getDefaultProcessEngine 方法	49	6.5.1 绑定关系	77
5.2.2 registerProcessEngine 和 unregister 方法	50	6.5.2 解除绑定	78
5.2.3 retry 方法	50	6.5.3 查询用户组下的用户	78
5.2.4 destroy 方法	51	6.5.4 查询用户所属的用户组	79
5.3 ProcessEngine 对象	51	6.6 本章小结	80
5.3.1 服务组件	52	第 7 章 流程存储	81
5.3.2 close 方法	52	7.1 流程文件部署	82
5.3.3 流程引擎名称	53	7.1.1 Deployment 对象	82
5.4 本章小结	54	7.1.2 DeploymentBuilder 对象	82
第 6 章 用户组与用户	55	7.1.3 添加输入流资源	83
6.1 用户组管理	56	7.1.4 添加 classpath 资源	84
6.1.1 Group 对象	56	7.1.5 添加字符串资源	85
6.1.2 创建用户组	56	7.1.6 添加压缩包资源	85
6.1.3 修改用户组	57	7.1.7 修改部署名称	86
6.1.4 删除用户组	58	7.1.8 过滤重复部署	86
6.2 Activiti 数据查询	58	7.1.9 资源部署	87
6.2.1 查询对象	59	7.2 流程定义的管理	87
6.2.2 list 方法	59	7.2.1 ProcessDefinition 对象	87
6.2.3 listPage 方法	60	7.2.2 流程部署	88
6.2.4 count 方法	60	7.2.3 流程图部署	89
6.2.5 排序方法	61	7.2.4 中止与激活流程定义	90
6.2.6 ID 排序问题	62	7.3 流程定义权限	91
6.2.7 多字段排序	63	7.3.1 设置流程定义的用户权限	91
6.2.8 singleResult 方法	65	7.3.2 设置流程定义的用户组权限	93
6.2.9 用户组数据查询	65	7.3.3 IdentityLink 对象	93
6.3 用户管理	67	7.3.4 查询权限数据	93
6.3.1 User 对象	67	7.4 RepositoryService 数据查询与删除	95
6.3.2 添加用户	67	7.4.1 查询部署资源	96
6.3.3 修改用户	68	7.4.2 查询流程文件	96
6.3.4 删除用户	68	7.4.3 查询流程图	97
		7.4.4 查询部署资源名称	98
		7.4.5 删除部署资源	98
		7.4.6 DeploymentQuery 对象	99
		7.4.7 ProcessDefinitionQuery 对象	100
		7.5 本章小结	100

第8章 流程任务管理..... 101

8.1 任务的创建与删除..... 102

8.1.1 Task 对象..... 102

8.1.2 创建与保存 Task 实例..... 103

8.1.3 删除任务..... 103

8.2 任务权限..... 104

8.2.1 设置候选用户组..... 105

8.2.2 设置候选用户..... 106

8.2.3 权限数据查询..... 106

8.2.4 设置任务持有人..... 108

8.2.5 设置任务受理人..... 109

8.2.6 添加任务权限数据..... 110

8.2.7 删除用户组权限..... 112

8.2.8 删除用户权限..... 113

8.3 任务参数..... 115

8.3.1 基本类型参数设置..... 115

8.3.2 序列化参数..... 116

8.3.3 获取参数..... 117

8.3.4 参数作用域..... 118

8.3.5 设置多个参数..... 119

8.4 任务附件管理..... 120

8.4.1 AttachmentEntity 对象..... 120

8.4.2 创建任务附件..... 120

8.4.3 附件查询..... 122

8.4.4 删除附件..... 123

8.5 任务评论与事件记录..... 123

8.5.1 CommentEntity 对象..... 123

8.5.2 新增任务评论..... 124

8.5.3 事件的记录..... 124

8.5.4 数据查询..... 126

8.6 任务声明与完成..... 126

8.6.1 任务声明..... 126

8.6.2 任务完成..... 127

8.7 本章小结..... 128

第9章 流程控制..... 129

9.1 流程实例与执行流..... 130

9.1.1 流程实例与执行流概念..... 130

9.1.2 流程实例和执行流对象..... 130

(ProcessInstance 与 Execution)

..... 130

9.2 启动流程..... 131

9.2.1 startProcessInstanceById 方法..... 131

9.2.2 startProcessInstanceByKey 方法..... 133

9.2.3 startProcessInstanceByMessage..... 134

9.3 流程参数..... 135

9.3.1 设置与查询流程参数..... 136

9.3.2 流程参数的作用域..... 136

9.3.3 其他设置参数方法..... 138

9.4 流程操作..... 138

9.4.1 发送信号..... 138

9.4.2 触发信号事件..... 140

9.4.3 触发消息事件..... 143

9.4.4 中断与激活流程..... 145

9.4.5 删除流程..... 145

9.5 流程数据查询..... 146

9.5.1 执行流查询..... 146

9.5.2 流程实例查询..... 149

9.6 本章小结..... 150

第10章 历史数据管理和流程引擎管理..... 151

10.1 历史数据管理..... 152

10.1.1 历史流程实例查询..... 152

10.1.2 历史任务查询..... 154

10.1.3 历史行为查询..... 157

10.1.4 历史流程明细查询..... 159

10.1.5 删除历史流程实例和历史..... 161

任务..... 161

10.2 工作的产生..... 162

10.2.1 异步任务产生的工作..... 162

10.2.2 定时中间事件产生的工作..... 164

10.2.3 定时边界事件产生的工作..... 165

10.2.4 定时开始事件产生的工作..... 166

10.2.5 流程抛出事件产生的工作..... 167

10.2.6 工作的执行..... 169

10.3 工作管理..... 170

10.3.1 JobQuery 对象..... 170

10.3.2 获取工作异常信息..... 173

10.3.3 设置工作执行次数..... 173

10.4 数据库管理..... 174

10.4.1 查询引擎属性..... 175

10.4.2 数据表信息查询..... 175

10.4.3 数据库操作..... 176

10.4.4 数据表查询..... 177

10.5 本章小结..... 178

第11章 流程事件..... 179

11.1 事件分类..... 180

11.1.1 按照事件的位置分类..... 180

11.1.2 按照事件的特性分类..... 180

11.2 事件定义..... 180

11.2.1 定时器事件定义..... 181

11.2.2	cron 表达式	181	12.2.5	使用 JUEL 分配权限	229
11.2.3	错误事件定义	183	12.3	脚本任务	231
11.2.4	信号事件定义	183	12.3.1	脚本任务	231
11.2.5	消息事件定义	184	12.3.2	JavaScript 脚本	232
11.2.6	取消事件定义	184	12.3.3	Groovy 脚本	233
11.2.7	补偿事件定义	185	12.3.4	设置返回值	233
11.2.8	其他事件定义	185	12.3.5	JUEL 脚本	234
11.3	开始事件	185	12.4	服务任务	235
11.3.1	无指定开始事件	185	12.4.1	Java 服务任务	235
11.3.2	定时器开始事件	186	12.4.2	实现 JavaDelegate	236
11.3.3	消息开始事件	187	12.4.3	使用普通 Java Bean	237
11.3.4	错误开始事件	189	12.4.4	在 Activiti 中调用 Web Service	238
11.4	结束事件	191	12.4.5	import 元素	238
11.4.1	无指定结束事件	191	12.4.6	itemDefinition 元素和 message 元素	239
11.4.2	错误结束事件	191	12.4.7	interface 元素与 operation 元素	239
11.4.3	取消结束事件和取消边界 事件	193	12.4.8	设置 Web Service 参数与 返回值	239
11.5	边界事件	196	12.4.9	发布 Web Service	240
11.5.1	定时器边界事件	196	12.4.10	使用 Web Service Task	241
11.5.2	错误边界事件	198	12.4.11	JavaDelegate 属性注入	244
11.5.3	信号边界事件	200	12.4.12	在 JavaDelegate 中调用 Web Service	246
11.5.4	补偿边界事件	202	12.4.13	Shell 任务	247
11.6	中间事件	205	12.5	其他任务	249
11.6.1	中间事件分类	205	12.5.1	手工任务和接收任务	249
11.6.2	定时器中间事件	206	12.5.2	邮件任务	250
11.6.3	信号中间 Catching 事件	207	12.5.3	Mule 任务和业务规则任务	252
11.6.4	信号中间 Throwing 事件	209	12.6	任务监听器	252
11.6.5	消息中间事件	211	12.6.1	使用 class 指定监听器	252
11.6.6	无指定中间事件	211	12.6.2	使用 expression 指定监听器	253
11.7	补偿中间事件	211	12.6.3	使用 delegateExpression 指定 监听器	254
11.7.1	补偿执行次数	211	12.6.4	监听器的触发	254
11.7.2	补偿执行范围	213	12.6.5	属性注入	255
11.7.3	补偿的执行顺序	217	12.7	流程监听器	256
11.7.4	补偿的参数设置	219	12.7.1	配置流程监听器	256
11.8	本章小结	220	12.7.2	触发流程监听器的事件	257
第 12 章	流程任务	221	12.8	本章小结	259
12.1	BPMN 2.0 中的任务	222	第 13 章	其他流程元素	260
12.1.1	任务的继承	222	13.1	子流程	261
12.1.2	XML 约束	222	13.1.1	嵌入式子流程	261
12.1.3	任务的类型	224			
12.2	用户任务	225			
12.2.1	分配任务候选人	225			
12.2.2	分配任务受理人	226			
12.2.3	权限分配扩展	227			
12.2.4	使用任务监听器进行权限 分配	228			

13.1.2	调用式子流程	263	14.6.4	行为语法	316
13.1.3	调用式子流程的参数传递	265	14.7	Activiti 调用规则	317
13.1.4	事件子流程	267	14.7.1	业务规则任务	317
13.1.5	事务子流程	269	14.7.2	制定销售单优惠规则	318
13.2	顺序流	271	14.7.3	实现销售流程	321
13.2.1	条件顺序流	272	14.8	本章小结	323
13.2.2	默认顺序流	273	第 15 章	整合第三方框架	324
13.3	流程关口	275	15.1	Spring 框架	325
13.3.1	单向关口	275	15.1.1	Spring 的 IoC	325
13.3.2	并行关口	277	15.1.2	Spring 的 AOP	326
13.3.3	兼容关口	279	15.1.3	使用 IoC	326
13.3.4	事件关口	281	15.1.4	使用 AOP	327
13.4	流程活动特性	283	15.2	Activiti 整合 Spring	328
13.4.1	多实例活动	283	15.2.1	SpringProcessEngineConfiguration	328
13.4.2	设置循环数据	285	15.2.2	ProcessEngineFactoryBean	330
13.4.3	获取循环元素	286	15.2.3	在 bean 中注入 Activiti 服务	331
13.4.4	循环的内置参数	288	15.2.4	在 Activiti 中使用 Spring 的 bean	331
13.4.5	循环结束条件	290	15.3	Activiti 整合 Web 项目	333
13.4.6	补偿处理者	291	15.3.1	安装 Tomcat 插件	334
13.5	本章小结	292	15.3.2	加入 Spring	336
第 14 章	Activiti 与规则引擎	293	15.3.3	整合 Hibernate	337
14.1	概述	294	15.3.4	配置声明式事务	339
14.1.1	规则引擎 Drools	294	15.3.5	添加 Struts 配置	340
14.1.2	Drools 下载与安装	294	15.3.6	实现一个最简单的逻辑	340
14.2	开发第一个 Drools 应用	295	15.3.7	测试事务	342
14.2.1	建立 Drools 环境	295	15.3.8	加入 Activiti	343
14.2.2	编写规则	296	15.4	Mule ESB	344
14.2.3	加载与运行	296	15.4.1	Mule 简介	344
14.3	Drools 规则语法概述	297	15.4.2	开发第一个 Mule 应用	344
14.3.1	规则文件结构	297	15.4.3	请求 Mule 服务	345
14.3.2	关键字	298	15.4.4	代理 HTTP 服务	346
14.3.3	规则编译	299	15.4.5	Mule 与 Spring 整合	347
14.4	类型声明	299	15.4.6	在 Activiti 中向 Mule 发送请求	348
14.4.1	声明新类型	300	15.5	本章小结	351
14.4.2	使用 ASM 操作字节码	301	第 16 章	Activiti 开放的 Web Service	352
14.4.3	类型声明的使用	302	16.1	Web Service 简介	353
14.4.4	类型的继承	303	16.1.1	Web Service	353
14.4.5	声明元数据	304	16.1.2	SOAP 协议	353
14.5	函数和查询	305	16.1.3	REST 架构	353
14.5.1	函数定义和使用	305	16.1.4	REST 的设计准则	354
14.5.2	查询的定义和使用	307	16.1.5	REST 的主要特性	354
14.6	规则语法	308			
14.6.1	全局变量	308			
14.6.2	规则属性	310			
14.6.3	条件语法	313			

16.1.6	SOAPRPC 与 REST 的区别	355
16.2	Restlet 简介	355
16.2.1	在 Web 工程中使用 Restlet	356
16.2.2	发布 REST 的 Web Service	357
16.2.3	编写客户端	358
16.2.4	访问控制	358
16.2.5	部署 Activiti 的 Web Service	361
16.2.6	准备测试数据	361
16.2.7	查询流程引擎信息	362
16.2.8	使用 HttpClient 编写客户端	362
16.2.9	使用 CXF 编写客户端	363
16.3	流程存储服务	364
16.3.1	上传部署文件	364
16.3.2	部署数据查询	366
16.3.3	部署资源查询	367
16.3.4	查询单个部署资源	368
16.3.5	删除部署资源	369
16.3.6	删除多个部署资源	370
16.4	流程服务	371
16.4.1	查询流程定义集合	371
16.4.2	流程定义表单属性	372
16.4.3	查询流程定义表单	373
16.4.4	查询流程图	373
16.4.5	启动流程	373
16.4.6	查询流程实例集合	374
16.4.7	查询单个流程实例	374
16.4.8	查询流程图片	375
16.4.9	查询流程实例任务	376
16.4.10	发送信号	376
16.5	流程任务服务	377
16.5.1	查询任务摘要	377
16.5.2	获取任务集合	378
16.5.3	查询单个任务	379
16.5.4	查询任务表单	379
16.5.5	任务操作	379
16.6	身份服务	380
16.6.1	身份验证	380
16.6.2	查询用户	380
16.6.3	查询用户所属的用户组	381
16.6.4	查询用户组	381
16.6.5	查询用户组下的用户	381
16.7	管理服务	382
16.7.1	工作管理	382
16.7.2	数据库管理	382
16.8	本章小结	382

第 17 章	Activiti 功能进阶	384
17.1	流程虚拟机	385
17.1.1	概述	385
17.1.2	初试 PVM	385
17.1.3	自定义流程语言	386
17.1.4	执行流程语言	389
17.2	Activiti 的表单	391
17.2.1	概述	392
17.2.2	表单属性	392
17.2.3	外部表单	393
17.2.4	关于动态工作流和动态表单	394
17.3	流程图 XML	394
17.3.1	节点元素	395
17.3.2	衔接元素	395
17.3.3	流程图与流程文件的转换	395
17.4	流程操作	397
17.4.1	流程回退	397
17.4.2	会签	398
17.5	本章小结	400
第 18 章	办公自动化系统	401
18.1	使用技术	402
18.1.1	表现层技术	402
18.1.2	MVC 框架	402
18.1.3	Spring 和 Hibernate	402
18.2	功能简述	403
18.2.1	系统的角色管理	403
18.2.2	薪资计算流程	403
18.2.3	请假流程	403
18.2.4	薪资调整流程	403
18.2.5	报销流程	404
18.3	框架整合	404
18.3.1	创建 Web 工程	404
18.3.2	加入 Spring	406
18.3.3	加入 Hibernate	407
18.3.4	整合 Struts2	408
18.3.5	整合 Activiti	409
18.4	数据库设计	410
18.4.1	薪资表	410
18.4.2	请假记录表	411
18.4.3	薪资调整记录表	412
18.4.4	报销记录表	412
18.5	初始化数据	413
18.5.1	初始化角色数据	413
18.5.2	薪资计算流程	414
18.5.3	请假流程	415

18.5.4	报销流程	415	18.8.1	申请列表	429
18.5.5	薪资调整流程	416	18.8.2	请假申请列表	430
18.6	角色管理	418	18.8.3	报销申请列表	431
18.6.1	用户组管理	418	18.8.4	薪资调整列表	431
18.6.2	用户列表	420	18.8.5	查看流程图	432
18.6.3	新建用户	421	18.9	流程任务	434
18.6.4	用户登录	422	18.9.1	待办任务列表	434
18.7	流程启动	423	18.9.2	领取任务与受理任务列表	436
18.7.1	启动请假流程	423	18.9.3	查询任务信息	437
18.7.2	启动报销流程	426	18.9.4	任务审批	439
18.7.3	启动薪资调整流程	428	18.9.5	运行 OA 的流程	439
18.8	申请列表	429	18.10	本章小结	440

第 1 章

Activiti 介绍

本章要点

- ✎ 工作流介绍
- ✎ BPMN 2.0 规范介绍
- ✎ Activiti 介绍

在计算机尚未普及时，许多工作流程采用手工传递纸张表单的方式，一级一级审批签字，工作效率非常低下。而对于数据统计以及生成报表的工作，则需要经过大量的手工操作才能完成。随着计算机的普及，这些工作的参与者只需要在计算机系统中输入工作内容，系统就会按照定义好的流程自动执行，从而各级审批者可以得到工作的信息并做出相应的审批和管理操作。数据统计和报表的生成也由系统代为完成，这大大提高了工作效率。在这种背景下，各种工作流应用以及中间件应运而生。

随着工作流应用在日常工作中越来越广泛，Java EE 领域出现了许多优秀的工作流引擎，例如 JBoss 社区的 jBPM、OpenSymphony 的 OSWorkflow 等。在 2010 年 5 月 17 日，以 Tom Baeyens 为首的工作流小组发布了一个全新的工作流引擎——Activiti，该工作流引擎的第一个版本为 5.0 alpha1。由于 Tom Baeyens 是 jBPM 的创始人（由于意见分歧离开 JBoss），因此 Activiti 的团队希望该流程引擎是 jBPM 4 的延伸，他们希望在从 jBPM 中积累的经验和知识的基础上，继续进行新一代工作流解决方案的开发，因此将第一个 Activiti 版本定义为 5.0 alpha1。本书将以 Activiti 5.10 为基础，深入探讨 Activiti 的特性。

1.1 工作流介绍

工作流（Workflow），是对工作流程及其各操作步骤之间业务规则的抽象、概括和描述。工作流建模，则是将工作流程中的工作如何前后组织在一起的逻辑和规则在计算机中以恰当的模式进行表示并对其进行实施计算。工作流要解决的主要问题是：为实现某个业务目标，在多个参与者之间，利用计算机，按某种预定规则自动传递文档、信息或者任务。工作流管理系统（Workflow Management System, WfMS）的主要功能是通过计算机技术的支持去定义、执行和管理工作流，协调工作流执行过程中工作之间以及群体成员之间的信息交互。工作流需要依靠工作流管理系统来实现。工作流属于计算机支持的协同工作（Computer Supported Cooperative Work, CSCW）的一部分。后者研究一个群体如何在计算机的帮助下实现协同工作的。（注：本段内容来自维基百科。）

早在 20 世纪 70 年代，办公自动化概念出现的时候，工作流思想就已经出现，那时人们希望新的技术可以改善办公效率，但是由于当时计算机并没有普及，网络技术还不普遍等原因，那时的工作流技术仅仅停留在研究领域。到了 20 世纪 90 年代以后，各种技术条件逐渐成熟，工作流技术被应用于电信、软件、制造、金融和办公自动化等领域。随着工作流技术的兴起，为了给全部业务的参与者提供易与理解的标准标记法，由业务流程管理倡议组织（BPMI）开发出了“业务流程建模标记法”（Business Process Modeling Notation, BPMN）。BPMI 组织于 2005 年并入到 OMG 组织，当前 BPMN 规范由 OMG 组织进行维护。

1.2 BPMN 2.0 规范简述

BPMN 规范 1.0 版由 BPMI 组织于 2004 年发布。BPMN 规范的发布是为了让业务流程的全部参与人员可以对流程进行可视化管理，提供一套让所有参与人员都易于理解的语言和标记，为业务流程的设计人员（非技术人员）和实现人员（技术人员）建立起一座桥梁。BPMI 组织于 2005 年合并到 OMG（Object Management Group）组织中，其在 2008 年 1 月发布了 BPMN 1.1 规范。BPMN 2.0 规范于 2011 年 1 月正式发布，并且全称改为 Business Process Model And Notation（业务流程模型和符号）。

在 1.0 版本的 BPMN 规范中，只注重流程元素的图形，这使其在流程分析人员中非常受

欢迎，而 BPMN 2.0 版本不但继承了 1.0 版本的内容，更注重流程执行语法和标准交换格式。

1.2.1 BPMN 2.0 概述

BPMN 2.0 规范定义了业务流程的符号以及模型，并且为流程定义设定了转换格式，目的是让流程的定义实现可移植性，这样一来，用户可以在不同的供应商环境中定义流程，并且这些流程可以移植到其他遵守 BPMN 2.0 规范的供应商环境中。BPMN 2.0 在以下方面扩展了 BPMN 1.2。

- 规范了流程元素的执行语法。
- 定义了流程模型和流程图的扩展机制。
- 细化了事件的组成。
- 扩展了参与者的交互定义。
- 定义了编排模型。

1.2.2 BPMN 2.0 元素

推出 BPMN 2.0 的目的是建立简单的并且易懂的业务流程模型，但是同时它又需要处理高度复杂的业务流程，因此要解决这两个矛盾的要求，需要在规范中定义标准的图形和符号。BPMN 中定义了 5 类基础的元素分类。

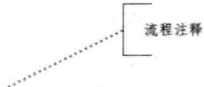
- **流对象 (Flow Objects)**：在一个业务流程中，流对象是用于定义行为的图形元素，主要有事件 (Events)、活动 (Activities) 和关口 (Gateways) 3 种流对象。
- **数据 (Data)**：主要有数据对象 (Data Objects)、数据输入 (Data Inputs)、数据输出 (Data Outputs) 和数据存储 (Data Stores) 4 种元素。
- **连接对象 (Connecting Objects)**：用于连接流对象，主要有 4 种连接流对象的方式，包括顺序流 (Sequence Flows)、消息流 (Message Flows)、关联 (Associations) 和数据关联 (Data Associations)。
- **泳道 (Swimlanes)**：泳道有两种途径组织基础的模型元素，分别是池 (Pools) 和道 (Lanes)。
- **制品 (Artifacts)**：制品主要用于为流程提供附加信息，当前制品包括组 (Group) 和注释 (Text Annotation)。

以上的元素分类以及其下的元素，均是 BPMN 规范中元素的组成部分，其中，每个对象均有自己对应的图形。表 1-1 给出了各个元素的图形及其描述。

表 1-1 BPMN 元素的图形及其描述

元 素	图 形	描 述
事件 (Events)		用于描述流程中发生的事件，其会对流程产生影响，并会被触发或者会产生结果
活动 (Activities)		活动是工作流中一个通用的术语，活动包括任务 (Task) 和子流程 (Sub-Process)
关口 (Gateways)		关口主要用于控制流程中顺序流的走向，使用关口可以控制流程的分支与合并
顺序流 (Sequence Flows)		顺序流显示流程将会执行哪个活动
消息流 (Message Flows)		消息流主要用于显示消息在流程参与者之间的传递情况

续表

元 素	图 形	描 述
关联 (Associations)		主要用于连接流程元素及其制品 (流程信息)
池 (Pools)		存放道的容器
道 (Lanes)		用于区分流程参与人的职能范围
数据对象 (Data Objects)		数据对象主要表示活动需要的或者产生的信息
消息 (Message)		消息主要用于描述流程参与者之间的沟通内容
组 (Group)		组主要用于存放一些流程信息, 包括流程文档、流程分析信息等
注释 (Text Annotation)		主要为阅读流程图的人提供附加的文字信息

以上为 BPMN 规范中定义的基本元素, 在这些元素的基础上, 还会产生多种子元素, 例如关口 (Gateways) 元素, 还可以细分为单向关口、并行关口等, 这些细分的元素将会在本书的 BPMN 2.0 规范章节详细讲解。

1.2.3 BPMN 2.0 的 XML 结构

BPMN 2.0 规范除了定义流程元素的图形外, 还对流程描述文件做了语法上的定义。例如在定义一个 userTask 的时候, BPMN 2.0 规范规定需要有 id 和 name 属性; 定义一个顺序流, 需要提供 id、name、sourceRef 和 targetRef 属性。BPMN 2.0 定义了 XML 规范, 这样一份流程描述文件可以在不同的流程引擎中使用 (流程引擎需要遵守 BPMN 2.0 规范)。

除了 BPMN 2.0 规定的元素及属性外, 工作流引擎的供应商还可以在这些规范的基础上添加额外的属性, 但是这些额外的属性不允许与任何的 BPMN 2.0 元素产生冲突。除此之外, 供应商在对属性进行扩展时, 所产生的流程模型与流程图, 必须让流程的参与者能够轻松看懂, 而且规范中最基础的流程元素不允许发生改变, 因为这是 BPMN 2.0 规范的初衷。

BPMN 定义的 XML 元素以及各个元素的作用及其使用, 将会在本书后面章节中讲述。

1.3 Activiti 介绍

BPMN 2.0 规范在 2011 年发布后, 各个工作流引擎的供应商均向其靠拢, 包括 jBPM 5 和本书所介绍的 Activiti。Activiti 的第一个版本为 5.0 alpha1, 2010 年 12 月才发布了 Activiti 5.0 的正式版, 期间经历了 4 个 alpha 版本、2 个 beta 版本和 1 个 rc 版本, 直到 5.0 正式版才出现

对 BPMN 2.0 规范的支持。

▶▶ 1.3.1 Activiti 的出现

Activiti 的创始人 Tom Baeyens 是 jBPM 的创始人，由于与合作伙伴在 jBPM 的未来架构上产生意见分歧，Tom Baeyens 在 2010 年离开了 JBoss 并加入 Alfresco 公司。Tom Baeyens 的离开使得 jBPM 5 完全放弃了 jBPM 4 的架构，基于 Drools Flow 重新开发。而在 2010 年的 5 月，Tom Baeyens 发布了第一个 Activiti 版本（5.0 alpha1），由此看来，Activiti 更像是 jBPM 4 的延续，也许是为了让其看起来更像 jBPM 4 的延续，Activiti 团队直接将 Activiti 的第一个版本定义为 5.0。

▶▶ 1.3.2 Activiti 的发展

从 2010 年 5 月发布第一个 Activiti 版本至今（2013 年 3 月），Activiti 经历了近 20 个版本的演化，本书成书时已经发布到 5.12。Activiti 采用了宽松的 Apache Licence 2.0 开源协议，因此，Activiti 一出现，就得到了开源社区的大力支持。在开源社区的支持下，Activiti 吸引了更多的工作流专家参与到项目中，这促使了 Activiti 在工作流领域中的创新。通过开源社区的努力，相信 Activiti 会有一个美好的未来。

▶▶ 1.3.3 选择 Activiti 还是 jBPM

根据上述内容我们知道，jBPM 5 和 Activiti 同样支持 BPMN 2.0 规范，但是实际上 jBPM 5 已经推翻了 jBPM 3 和 jBPM 4 的架构，使用了 Drools Flow 作为工作流架构，这对于原来使用 jBPM 3 和 jBPM 4 的用户来说是非常郁闷的一件事（从零开始重新学习 jBPM 5）。而 Activiti 更像是原来 jBPM 4 的延续，因此对于原来使用 jBPM 3 和 jBPM 4 的用户来说，推荐使用 Activiti。但是由于 JBoss 中有一些优秀的项目（例如规则引擎 Drools、Seam 等），jBPM 5 与这些项目进行整合具有先天的优势，因此如何选择还需要权衡。

除了原来的架构有所改变之外，还需要考虑的是，jBPM 5 采用的是 LGPL 开源协议，如果要在其基础上使用以修改和衍生的方式做二次开发的商业软件，涉及的修改部分就需要使用 LGPL 协议，对于这些商用的软件来说，需要对 jBPM 5 的源代码进行修改并做二次开发，显然这不是个明智的选择。相对于 jBPM 5，Activiti 采用了更为宽松的 Apache License 2.0 协议，该协议鼓励代码共享并尊重原作者的著作权，其允许对代码进行修改和发布而不管其用途。

1.4 本章小结

本章对工作流的起源以及发展做了简单的介绍，其中主要介绍了工作流领域的 BPMN 2.0 规范。BPMN 2.0 规范为工作流应用提供了语言及图形的标准。本章还讲述了 Activiti 的产生背景，简述了 Activiti 目前所具有的优势，并且与“成熟”的 jBPM 进行了对比。下一章将带领读者开始 Activiti 之旅。