




第一部分 *Part 1*

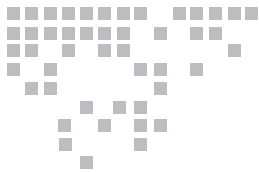
准 备 篇



工作流 (Work Flow) 引擎被广泛应用于各种信息化系统中, 将原本散乱甚至混乱的业务梳理后制定成业务规范流程, 进而约束业务的规范化处理和运转。需求人员、开发人员共同协作制定了符合 BPMN 2.0 规范的流程定义, 之后将其部署到工作流引擎中, 由它自动驱动业务流程的进行。

本部分作为准备篇, 第 1 章先介绍了什么是 Activiti 及其历史背景, 然后介绍工作流、BPM、BPMN 等概念, 使初次接触工作流的读者能快速认识相关概念及各种规范。第 2 章内容从搭建开发环境开始, 之后又介绍了 Activiti 的 Hello World, 帮助读者快速入门。

.....



Chapter 1

第 1 章

认识 Activiti

很多人对 workflow (Workflow) 应该不陌生。生活中到处都是活生生的“流”：在单位要请假，首先要找领导审批，在领导审批通过之后申请才获准；从网上购物，下单的那一刻就已经触发了一条 workflow，此时可以跟踪购物流程，什么时间下单、什么时间付款、什么时候发货、什么时候收到货，在快递单上签字的时候才等于一条 workflow 结束了。

workflow 应用广泛，在由任务驱动的各种系统中都能见到它的身影，例如，CRM、ERP、ECM、BI、OA 等。在企业应用中还有很多产品或平台集成 workflow 引擎，用来处理系统运行过程中发起的业务流程。

workflow 总是以任务 (Task) 的形式驱动人处理业务或者驱动业务系统自动完成作业。有了 workflow 引擎之后，我们不必一直等待其他人的工作进度，直白地说，我们只需要关心系统首页的待办任务数即可，由系统提醒当前有多少待办任务需要处理。

1.1 什么是 Activiti

大家第一次接触 Activiti 的时候不理解它为什么要叫这个名字，从词典中也没有找到对它的解释。可能有人会想到另外一个单词 Activity (活动)，与 Activiti 仅一个字母之差。在 workflow 方面有些基础的读者或许能很快理解，业务流程由多个环节串联起来并且每个环节被赋予任务，而每个任务又可以分为多个活动。举个日常的例子——网上购物的下单环节，首先需要搜索到要购买的商品，然后将其加入到购物车，最后下单填写邮寄地址并付款。这个例子中的每一动作都可以称为活动 (Activity)，也就是业务流程中最小的组成部分。多

个活动在英文中肯定要用复数形式，即 Activities；最后以复数化简的方式标示活动的集合，以此来诠释 Activiti 与工作流的目的与设计。

此项目是 Tom Bayen (jBPM 创始人) 自 2010 年离开 jBoss 加入 Alfresco 公司后的又一力作：第一版在 2010 年 5 月发布，当时仅支持最简单的流程处理，之后的版本陆续完善了对 BPMN 2.0 规范的支持。

值得一提的是，参与项目开发的除了 Tom Bayen 和十几位核心开发人员之外，还有其他公司的员工参与，例如，SpringSource、MuleSoft、Salves、Signavio、FuseSource、NextLevel 等。

Activiti 是一个针对企业用户、开发人员、系统管理员的轻量级 workflow 业务管理平台，其核心是使用 Java 开发的快速、稳定的 BPMN[⊖] 2.0 流程引擎。Activiti 是在 ApacheV2 许可下发布的，可以运行在任何类型的 Java 程序中，例如服务器、集群、云服务等。Activiti 可以完美地与 Spring 集成。同时，基于简约思想的设计使 Activiti 非常轻量级。

Activiti 有着活跃的社区，而且越来越多的企业都选择 Activiti 作为自己的流程引擎或者将其嵌入到自己的系统平台中（例如 ESB）。

接下来简单了解一下工作流及其相关规范的历史。

1.2 工作流基础

1.2.1 什么是 BPM

BPM 是 Business Process Management 的缩写，中文含义是业务流程管理，是一套达成企业各种业务环节整合的全面管理模式。

BPM 是为了实现一定的经营目的而执行的一系列逻辑相关的活动的集合。业务流程的输出是满足市场需要的产品或服务。根据功能、管理范围等的不同，企业流程管理一般分为生产流程层、运作层、计划层和战略层四个层次。BPM 是根据业务环境的变化，推进人与人之间、人与系统之间，以及系统与系统之间的整合及调整的经营方法与解决方案的 IT 工具。

BPM 最早是由工作流和企业应用集成 (Enterprise Application Intergration) 逐步融合而发展起来的，当时是为了满足无纸化办公需求（这也是最早的需求之一）。笔者早期参与 OA 系统开发时曾经见到过“原始的”工作流——没有工作流引擎，整个流程均使用一系列单独为不同任务节点设计的页面串联起来，完成一个节点后在数据库标记当前任务的名称，以此做到“流程驱动”。

⊖ Business Process Modeling Notation，详细信息可参考 <http://www.bpmn.org/>。

4 ❖ 第一部分 准备篇

随着时间的推移，BPM 的定义范围逐步扩展，不仅用来满足无纸化办公需求，现在 BPM 是一种企业集成技术，作为对面向服务系统架构 SOA (Service-Oriented Architecture)、企业应用集成 EAI (Enterprise Application Integration)、企业服务总线 ESB (Enterprise Service Bus) 的补充。

从概念上来说，BPM 包含两个不同方面的意思：管理规范和软件工程。各大 BPM 供应商长期以来试图抽象这两个不同的方面，但是依然混乱。

作为管理规范，BPM 是每一个战略管理者的责任。BPM 是组织必须执行的核心业务流程，包含了企业价值和如何提供其实现。作为日常工作的一部分，业务系统可以借助模型和流程规范地定义业务流程。BPM 流程图表达的是执行流程的步骤，已完成特定目标。特别说明的是这些模型用于人与人的沟通。这些都是诠释未决的，这意味着它们可以包含更高级别有价值的信息而不包括不必要的细节。这种诠释未决的过程模型也被称为抽象业务流程 (Abstract Business Processes.)。

BPM 作为软件工程时可以由 BPM 系统 (BPMS) 执行可执行的业务流程。可执行的业务流程是在一个流程基础上表示不同的流程顺序。流程图完全可以看做一个抽象的业务流程。可执行流程不同于抽象业务流程，因为它总是以最简单的方式运行。这部分内容也是被大多数厂商认同并接受的。

1.2.2 workflow 生命周期

一个完整的工作流生命周期会经过 5 步，并且迭代循环，如图 1-1 所示。

- ❑ **定义**：工作流生命周期总是从流程定义开始。此阶段的任务主要是收集业务需求并转化为流程定义。一般由业务需求人员进行，然后交由开发人员加工转化为计算机可以识别的流程定义。
- ❑ **发布**：由开发人员打包各种资源，然后在系统管理（平台）中发布流程定义。在具体的流程引擎中包括流程定义文件（bpmn20.xml 结尾）、自定义表单、任务监听类。
- ❑ **执行**：具体的流程引擎（例如，Activiti）按照事先定义的流程处理路线以任务驱动的方式执行业务流程。
- ❑ **监控**：此阶段是依赖执行阶段。业务人员在办理任务的同时收集每个任务

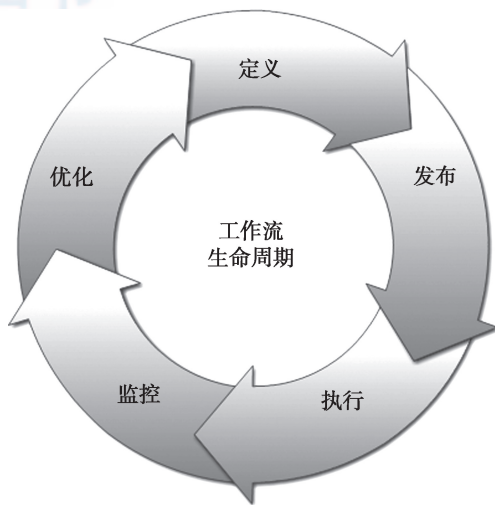


图 1-1 完整的工作流生命周期

(Task)的结果,然后根据结果做出相应处理,例如,在采购办公用品流程中,在通过领导审批之后,采购人员就要根据申请单外出采购。

- **优化**:在此阶段,一个完整的流程已经结束,或许能满足业务需求,或许需要优化,而糟糕的情况是需要重新设计(流程没结束就异常终止),优化与设计正是此阶段需要处理的。根据整个流程的运行过程结果分析问题的根源,然后在此基础上进一步改进,并再次开始一个新的周期。

1.2.3 什么是 BPMN

Business Process Modeling Notation,简称 BPMN,中文译为业务流程建模标注,是由 BPMN 标准组织发布的,其第一版 BPMN 1.0 规范于 2004 年 5 月发布。经过多年的改进新的规范 BPMN 2.0 于 2011 年发布。之后各大厂商、开源社区均基于 2.0 规范设计自己的流程引擎,结束了各个厂商“各自为政”的局面,相应地统一了标准,从而利于以后的产品迁移。

BPMN 定义了业务流程图,其基于流程图技术,同时对创建业务流程操作的图形化模型进行了裁减。业务流程的模型就是图形化对象的网图,包括活动(也可以说工作)和定义操作顺序的流控制。

在 BPMN 1.x 版本中的一些概念,如人工任务、可以执行脚本、自动决策等,都是独立于供应商的可视化标准化的方式。在 BPMN 2.0 规范中重点聚焦在如何执行语义和一个被业界认可的通用交换格式。这意味着基于 BPMN 2.0 的流程建模不仅在流程设计器上可以通用,还可以在任何符合 BPMN 2.0 规范的流程引擎上执行。

关于 BPMN 的细节内容有很多,官方文档足足有 500 多页,本书会在第 4 章介绍 Activiti 支持的 BPMN 2.0 规范以及 Activiti 在 BPMN 2.0 规范基础上的扩展。关于 BPMN 的其他内容本书就不一一列举了,有兴趣的读者可以仔细阅读官方文档。

1.3 Activiti 的特点

1. 数据持久化

Activiti 的设计思想是简洁、快速。有过应用开发经验的开发人员都知道应用的瓶颈体现在和数据库交换数据的过程中,针对这一点 Activiti 选择了使用 MyBatis,从而可以通过最优的 SQL 语句执行 Command,仅凭如此就能让引擎在速度上保持最高的性能。

2. 引擎 Service 接口

Activiti 引擎提供了七大 Service 接口,均通过 ProcessEngine 获取,并且支持链式

6 ❖ 第一部分 准备篇

API 编程风格。表 1-1 简单列出七个 Service 接口及其作用，具体使用会在后面的章节陆续介绍。

表 1-1 Activiti 引擎的七大 Service 接口

Service 接口	作用
RepositoryService	流程仓库 Service，用于管理流程仓库，例如，部署、删除、读取流程资源
IdentifyService	身份 Service，可以管理和查询用户、组之间的关系
RuntimeService	运行时 Service，可以处理所有正在运行状态的流程实例、任务等
TaskService	任务 Service，用于管理、查询任务，例如，签收、办理、指派等
FormService	表单 Service，用于读取和流程、任务相关的表单数据
HistoryService	历史 Service，可以查询所有历史数据，例如，流程实例、任务、活动、变量、附件等
ManagementService	引擎管理 Service，和具体业务无关，主要是可以查询引擎配置、数据库、作业等

3. 流程设计器

在 jBPM4 时代有专门的 Eclipse 插件可以用来设计 jPDL，同样 Activiti 团队也专门设计了用来设计 BPMN 2.0 规范的流程设计器——Eclipse Designer。此外还有 Signavio 公司为 Activiti 定制的基于 Web 的 Activiti Modeler 流程设计器。

4. 原生支持 Spring

Activiti 原生支持 Spring，这一点对企业应用来说尤为重要：可以很轻松地进行 Spring 集成，非常方便管理事务和解析表达式（Expression）。

5. 分离运行时与历史数据

Activiti 继承自 jBPM4，在表结构设计方面也遵循运行时与历史数据的分离，这样的设计可以快速读取运行时数据，仅当需要查询历史数据时再从专门的历史数据表中读取。这种设计方式可以大幅提高数据的存取效率，尤其是当数据日积月累时依然能够快速反应。

1.4 Activiti 的应用

目前 Activiti 在国外已被很多厂商所使用，甚至有人专门成立了公司来培训 Activiti 的使用。Activiti 在国内的发展正在呈直线趋势上升，已经成立了由很多热情的技术爱好者参与的技术社区。目前，很多新项目、新产品都开始采用 Activiti 作为新一代工作流引擎。

1. 在系统集成方面应用

- ❑ 与 ESB (Enterprise Service Bus, 企业服务总线) 整合, 例如 Mule。
- ❑ 与规则引擎 (Rule Engine) 整合, 例如 JBoss Drools。
- ❑ 嵌入已有系统平台, 例如, 很多公司都开发了自己的系统平台, 在其中嵌入 Activiti 作为平台的一部分。

2. 在其他产品中应用

Alfresco[⊖]公司的 ECM (Enterprise Content Management) 产品 Alfresco 在企业中应用广泛, 主要涉及文档管理、协作、记录管理、知识库管理、Web 内容管理等。

如果企业或客户正在使用 Alfresco 管理文档, 那么针对文档管理流程设计的流程定义可以直接部署在 Alfresco 上使用; 如果之前没有接触过 jBPM 而现在学会了使用 Activiti, 那么不用再去学习其他的流程引擎。关于 Activiti 与 jBPM 的区别在 1.6 节会提到。

在 Activiti 没有发布之前一直使用 jBPM 作为流程引擎, 在 Activiti 成熟以后 Alfresco 同时支持两者, 当然在以后的某个时间可能会取消对 jBPM 的支持。

1.5 Activiti 架构与组件

Activiti 架构中最重要的肯定是引擎, 当然还有刚刚提到的外部工具和组件, 如图 1-2 所示。

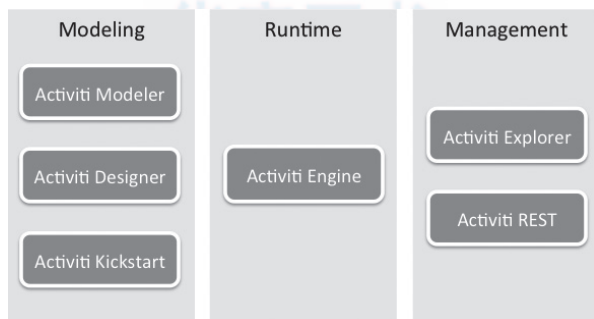


图 1-2 Activiti 架构图

下面依次介绍 Activiti 架构图中的各个组件。

- ❑ **Activiti Engine** : 作为最核心的模块, 提供针对 BPMN 2.0 规范的解析、执行、创建、管理 (任务、流程实例)、查询历史记录并根据结果生成报表。
- ❑ **Activiti Modeler** : 是模型设计器, 其并非由 Activiti 公司所开发, 而是由业界认可

⊖ <http://www.alfresco.com/>

8 ❖ 第一部分 准备篇

的 Signavio 公司赠送的 (Signavio[⊖]原本是收费的产品, 现在被免费授权给 Activiti 用户使用)。适用于业务人员把需求转换为规范流程定义。

- ❑ **Activiti Designer** : 功能和 Activiti Modeler 类似, 同样提供了基于 BPMN 2.0 规范的可视化设计功能, 但是目前还没有完全支持 BPMN 规范的定义。适用于开发人员, 可以把业务需求人员用 Signavio 设计的流程定义 (XML 格式) 导入到 Designer 中, 从而让开发人员将其进一步加工成为可以运行的流程定义。
- ❑ **Activiti Explorer** : 可以用来管理仓库、用户、组, 启动流程、任务办理等。此组件使用 REST 风格 API (目的在于让开发人员快速入门), 提供一个基础的设计模型。如果业务简单, 也可以直接使用无需开发。还可以作为后台管理员的流程、任务管理系统使用。
- ❑ **Activiti REST** : 提供 Restful 风格的服务, 允许客户端以 JSON 的方式与引擎的 REST API 交互, 通用的协议具有跨平台、跨语言的特性。

1.6 Activiti 与 jBPM5 比较

目前流行的工作流引擎有 Activiti 和 jBPM5, 而在 jBPM5 发布以前大多数项目、平台都是基于 jBPM3、jBPM4 开发的。本节内容从技术和实际应用上对 Activiti 和 jBPM5 进行比较。表 1-2 从技术层面比较了两者的区别。

表 1-2 Activiti 与 jBPM5 的技术层面对比

技术组成	Activiti	jBPM
ORM 框架	MyBatis3	Hibernate3
持久化标准	无	EJB JPA 规范
事务管理	MyBatis 自带 /Spring 集成事务	Bitronix, 基于 JTA 事务管理
数据库连接方式	Jdbc/DataSource	Jdbc/DataSource
Spring 支持	原生支持 Spring, 在流程中可以使用 Spring 代理的 Bean 作为表达式的一部分, 并且支持 JPA 及事务管理	默认没有提供对 Spring 的支持
支持的数据库	Oracle、SQL Server、MySQL、H2、内存数据库等	Oracle、SQL Server、MySQL、内存数据库等
设计模式	命令模式、观察者模式等	
内部服务通信	Service 间通过 API 调用	基于 Apache Mina 异步通信
集成接口	SOAP、Mule、RESTful	消息通信

⊖ <http://www.signavio.com/en.html>

(续)

技术组成	Activiti	jBPM
支持的流程格式	BPMN2、xPDL、jPDL 等 (由 PVM 实现)	目前仅支持 BPMN2 xml
引擎核心	PVM (流程虚拟机)	Drools
技术前身	jBPM3、jBPM4	Drools Flow
团队成员	除了 Alfresco 公司的雇员之外还有 Spring-Source、MuleSoft、Salves、Signavio、FuseSource、NextLevel 等公司的员工加入	有一个专门的团队, 此外还有一些个人参与者
附加工具	提供了基于 Eclipse 插件的流程设计器——Eclipse Designer, 提供基于 REST 风格的 Activiti Explorer, 可以用来管理仓库、用户、组、启动流程、任务办理等	同样提供 Eclipse 插件和一个 Web 应用管理流程
发布周期	固定每两个月发布一版, 其中包括: 引擎、Eclipse Designer、Activiti Explorer、REST 应用	jBPM 的发布周期相对来说不太固定, 发布内容包括引擎及基于 Eclipse 的设计器

Activiti 是基于 jBPM4 设计的衍生版本, 如果选择 Activiti 可以继续沿用 jBPM 的设计理念设计、整合 Activiti 到项目或平台中, 这也是相对于 jBPM5 来说的一个优势; 相反, 对于 jBPM5 来说要花点时间重新接受开发者的设计思想。

在各个流程引擎社区中有很多关于该如何选择 Activiti 和 jBPM5 的讨论, 这两者有着很多相似的地方, 争论主要是对规则引擎的支持: jBPM5 是基于 Drool Flow 所有自然深度继承而来的规则引擎 Drools; 早期的 Activiti 功能比较简单, 后来陆续添加的新特性也支持规则引擎 Drools, 开发人员只要简单配置规则接口即可达到与 jBPM5 一样的效果。

1.7 本章小结

本章内容主要是以初识 workflow 和 Activiti 的角度去讲解什么是 workflow、什么是 Activiti。从 Activiti 项目的发起、特点、应用、架构, 以及与其他同类产品比较的角度在概念层给读者一个引导和认识。

作为一个开发人员, 能够知道为什么需要学习 Activiti, Activiti 能帮助企业解决什么问题, 为什么要选择 Activiti 而不是其他的工作流引擎。

看到这里你或许会不耐烦了, 会想起一句话: “Talk is cheap. Show me the code[⊖]”。下一章将带领你体验 Activiti 的 Hello World。

⊖ 这是 Linux 的作者 Linus Torvalds 的名言, 意思是: 能说算不上什么, 有本事就把你的代码给我看看。