FAQ

1、为什么要写 Fire Workflow

本人从事企业 MIS 系统开发很多年头了，感觉 MIS 系统很多领域都有比较好的解决方案并已成为事实标准，例如 Spring，Hibernate 等等；然而工作流还没有令人满意的开源产品。我了解过的工作流产品（主要是开源的，收费产品没有什么研究，仅仅看看其白皮书而已）都存在如下毛病：

1)缺乏严密的理论做支撑，工作流模型大多千篇一律地照搬 WfMC 的 xpdl，

2)因为缺乏理论支撑，所以工作流引擎的算法有点七拼八凑，扩展性也比较差。

3)没有好的设计器，应用比较困难

最近研究并应用了一下 JBoss 的 Jbpm，除了其面向图的引擎算法让我眼前一亮外，其他的也不是令人满意。其引擎的扩展性不好，表结构太复杂，在大数据量系统中，性能令人堪忧。

鉴于此，我动手写了一个 Fire Workflow，抛砖引玉。

2、Fireflow 的特点

理论严密

Fire Workflow 以 Petri Net 作为理论基础，流程的顺序流转、分支、汇聚、跳转等算法都有定义/定理为依据。

设计合理

Fire workflow 将工作流引擎的职责分解委派到各种服务中，每中服务都可以被扩展或者替换。

应用简单

Fire workflow 的 API 以及数据库表结构非常简单。

性能优良

Fire workflow 着重在流程实例的数据量，数据库 IO 等方面进行性能优化。

3、Fireflow 的构成

Fire Workflow 由模型、引擎、设计器（包含模拟器）三部分组成。

模型部分规定了流程定义文件的各种元素及其相互关系，例如流程(WorkflowProcess)、活动（Activity）、转移（Transition）、开始节点（StartNode）、结束节点（EndNode）、同步器（Synchronizer）。模型部分的实现在 org-fireflow-model.jar 中。

引擎读取流程定义文件并解释执行。引擎提供一组对象和相关的 API 供外部系统调用，如流程实例（ProcessInstance）、任务实例（TaskInstance）、工单（WorkItem）、事件等等。引擎部分的实现在 org-fireflow-engine.jar 中。

设计器编辑并输出流程定义文件。Fire Workflow 的设计器附带了强大的模拟器，可以在设计时模拟流程的执行，从而检查流程定义的正确性。

此处附带解释一下我的一个观点：我认为，流程定义文件和 java 文件一样，是应用系统源代码的一部分。因此，流程设计器做成了当前流行的 IDE 的插件，便于开发人员进行流程开发。而且每个流程单独一个定义文件，就像每个 java 类在通常情况下单独一个文件一样。

4、Fire Workflow 的流程定义语言为什么不使用 Xpdl

本人认为 Xpdl 好看不好用。

相较于 Xpdl，Fire workflow 的流程定义语言主要做了如下变动。

A)废除 Package 的概念：在我看来一个流程一个文件比较方便开发，流程定义文件在某中程度上和 java 类文件一样，是系统源代码的一部分

B)废除全局和局部的概念：在 xpdl 中有全局 DataField 和局部 DataField 区分，实际上其作用不大。

C)增加同步器节点：Fire Workflow 将流程中的节点分成两类 ，即 Acitivyt 和 Sychronizer（Start Node 和 End Node 是 synchronizer 的特例）。这两类节点分别代表了业务子系统的逻辑操作和工作流子系统的逻辑操作。

D)增加 Task 元素：一个 Activity 可以包含多个 Task，Task 代表实际的业务逻辑。

5、