.Net 三款工作流引擎比较：WWF、netBPM 和 ccflow

[fmms](http://www.open-open.com/news/user/profile?uid=2869) 2012-03-19 13:46:49 • 发布

**摘要：下面将对目前比较主流的三款工作流进行介绍和比较，然后通过三款流程引擎分别设计一个较典型的流程来给大家分别演示这三款创建流程的过程.这三款工作流程引擎分别是 Windows Workflow Foundation，NetBPM, CCFlow.**

下面将对目前比较主流的三款工作流进行介绍和比较，然后通过三款流程引擎分别设计一个较典型的流程来给大家分别演示这三款创建流程的过程.这三款工作流程引擎分别是 Windows Workflow Foundation，NetBPM, CCFlow.

NetBPM 与 CCFlow 是两款国内知名的开源软件，尤其是ccflow在国内的发展势头强劲。

这个典型的流程假设:公司有两级领导，一级为主管Chief,一级为老板Boss

场景描述：

  在某公司中，部门员工休假需要主管Chief的批准。   
  如果休假天数大于10天，则 在部门主管同意后，还必须老板Boss批准。   
  如果是部门主管请假则直接提交老板批准。   
  在休假被批准之前，申请人可以撤销休假申请。   
  申请批准后，对休假天数进行修改（也可以是其他业务数据处理）。 每次休假申请结束之 后，不管通过未通过或是否取消，都必须记录下来。   
  流程结束时，系统要把请假的结果信息Email给申请人。

  对于大于10天的申请，如果部门主管已批准同意而上级主管还未批准，这时申请人撤销申请后，系统应发Email通知部门主管申请已撤销。

我们这里只是一个模拟，当然现实生活中情况比这个更加复杂一些;

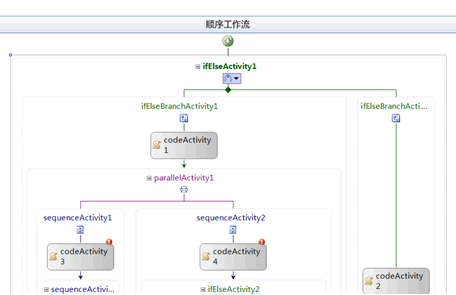
**Windows Workflow Foundation**

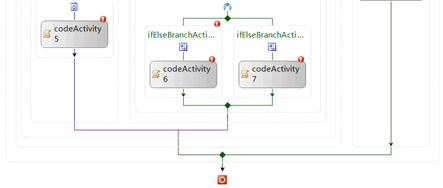
微软的工作流产品，提供一套工作流引擎和VS解决方案自带的流程设计器，但是该流程设计器面对的是程序员而非业务人员，所以界面比较专业，流程运行只能创建控制台应用程序，没有流程运行界面，没有表单库，如需要表单和界面需要二次开发。

使用WWF创建流程：

1. 启动VS2010，创建一个顺序工作流控制台的程序。  
2. 输入项目名称，点击确定，将自动进入流程设计界面。  
3. 自动生成的Workflow1.cs是一个工作流组件。  
4. 工具箱中拖放一个IfElse活动组件到设计界面上。  
5. 此时就需要较多的编码工作和表单界面设计工作，如在idelseBranchActivitiy1左侧分支，用以判断请假人是否新申请请假还是取消请假，激活Conditiong属性，并且添加内部事件EvaluateQingJiaNoValidCode,并激活,在内部输入逻辑代码根据数据库记录判断请假是否通过，未通过则取消请假。也可以走另一分支EvaluateQingJiaCode继续申请新的请假;  
6. 拖放parallelActivity1组件在IfElse节点后，用以判断请假人是否为Chief，设置sequenceActivity1中的codeActivity3属性的ExecuteCode处理程序为EvaluateChiefNoValidCode，并激活，内部代码用以判断不是Chief的情况，另一分支sequenceActivity1中则判断是Chief的情况；  
7.如果不是Chief请假，则需要在EvaluateChiefNoValidCode中进行逻辑判断和表单的设计，填写请假申请单， 并拖放IfElse组件，实现其中的codeActivity6代码用以判断大于10天的情况。  
8.F5即可运行控制台程序，其中的通过未通过或是否取消的数据需要记录，需要通过代码和设计数据库来实现，发送Email也需要代码实现，WWF没有提供该功能。

**WWF下设计的流程图如下：**





NetBPM

从JBpm1移植到.NET平台下的开源工作流项目，二次开发有一定的难度，因为其使用的Castle框架有很多子项目,技术点较多，需要一一熟悉后才能进行流程的二次开发。  
  
使用NetBPM创建工作流过程：  
1.    使用NetBPM的难点之一是要理解生成配置文件，提交请假单配置如下：

|  |  |  |
| --- | --- | --- |
| 01 | | <?xml version="1.0"?> |
| 02 | <!-- NOTE：在定义流程时，建议先画出流程图，然后再来定义，这样思维清晰，也不易于出错 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 03 | 关于processdefiniton.xml如何定义，请严格按照nPdl规定 --> | | |
| 04 | | <process-definition> |

|  |  |  |
| --- | --- | --- |
| 05 | |  |
| 06 | <!-- =================================== --> | | |

|  |  |
| --- | --- |
| 07 | <!-- == PROCESS DEFINITION PROPERTIES == --> |
| 08 | <!-- =================================== --> | |

|  |  |
| --- | --- |
| 09 | <name>请假DEMO</name> |
| 10 | <description>该流程模拟公司的请假流程,  </description> | |

|  |  |  |  |
| --- | --- | --- | --- |
| 11 | <responsible>ae</responsible> | | |
| 12 | |  |

|  |  |
| --- | --- |
| 13 | <!-- ====================== --> |
| 14 | <!-- == START & ENDSTATE == --> |

|  |  |
| --- | --- |
| 15 | <!-- ====================== --> |
| 16 | <start-state name="start leave request"> | |

|  |  |
| --- | --- |
| 17 | <description>提交请假单</description> |
| 18 | <!-- 定义了role，引擎在该start-state节点执行时，就会把执行者信息赋值给角色对应的属性“requester” --> | |

|  |  |
| --- | --- |
| 19 | <role>requester</role> |
| 20 | <!-- 在这里定义start-state的field，它表示该filed相关联的属性，并且在该state，它对属性的访问权利。 | |

|  |  |  |
| --- | --- | --- |
| 21 | 如果需要定义其在web表单上的操作界面，如何进行web表单显示等，需要在webinterface.xml文件对应节点补充该field --> | |
| 22 | <field attribute="start date" access="write-only-required" /> |

|  |  |
| --- | --- |
| 23 | <field attribute="end date"   access="write-only-required" /> |
| 24 | <field attribute="leave days"    access="write-only-required" /> | |

|  |  |  |
| --- | --- | --- |
| 25 | <field attribute="comment"    access="write-only" /> | |
| 26 | <transition to="Is Cancel Fork" /> |

|  |  |  |
| --- | --- | --- |
| 27 | </start-state> | |
| 28 |  |

|  |  |  |
| --- | --- | --- |
| 29 | <!-- 结束节点除名称外不要定义其他--> | |
| 30 | <end-state name="end" /> |

|  |  |
| --- | --- |
| 31 |  |
| 32 |  |

|  |  |
| --- | --- |
| 33 | <!-- ====================== --> |
| 34 | <!-- ==      Actions     == --> |

|  |  |
| --- | --- |
| 35 | <!-- ====================== --> |
| 36 | <!-- 解释：这里定义process-definition节点的action，有效的事件类型为：process-instance-start, process-instance-end and process-instance-cancel --> | |

|  |  |
| --- | --- |
| 37 |  |
| 38 | <!-- 此处具体为：在流程结束的时候, 发送E－Mail消息给申请者，记录请假日志 --> | |

|  |  |
| --- | --- |
| 39 | <action event="process-instance-end" |
| 40 | handler="NetBpm.Example.LeaveOfAbsence.EmailAction, NetBpm.Example.LeaveOfAbsence"   on-exception="log"> | |

|  |  |  |
| --- | --- | --- |
| 41 | <!--定义action参数，供委托类实例化类调用方法时获取使用。如这里的EmailAction的run方法发送邮件，需要知道发给谁，邮件标题等等，那么 | |
| 42 | 参数可以提供辅助--> |

|  |  |  |
| --- | --- | --- |
| 43 | <parameter name="to">previousActor</parameter> | |
| 44 | <parameter name="subject">您提交了请假申请</parameter> |

|  |  |  |
| --- | --- | --- |
| 45 | <parameter name="message">you requested a holiday from ${start date} to ${end date} with comment ${comment}</parameter> | |
| 46 | </action> |

|  |  |  |
| --- | --- | --- |
| 47 | <!-- 此处具体为：在流程结束的时候记录请假日志, 此处Log模拟 注意：每个节点可以定义多个action --> | |
| 48 | <action event="process-instance-end" |

|  |  |  |
| --- | --- | --- |
| 49 | handler="NetBpm.Example.LeaveOfAbsence.LogLeaveInfoAction, NetBpm.Example.LeaveOfAbsence"   on-exception="log"> | |
| 50 | <parameter name="LogInfo">记录请假日志? :) </parameter> |

|  |  |  |  |
| --- | --- | --- | --- |
| 51 | </action> | | |
| 52 | |  |

|  |  |
| --- | --- |
| 53 | <!-- ================ --> |
| 54 | <!-- == ATTRIBUTES == --> |

|  |  |
| --- | --- |
| 55 | <!-- ================ --> |
| 56 | <!-- 解释：定义属性值及其序列化方式。属性值一般包括3类 --> | |

|  |  |
| --- | --- |
| 57 | <!-- one:角色对应的属性 --> |
| 58 | <attribute name="requester" type="actor" /> | |

|  |  |  |
| --- | --- | --- |
| 59 | <attribute name="chief" type="actor" /> | |
| 60 | <attribute name="boss" type="actor" /> |

|  |  |
| --- | --- |
| 61 |  |
| 62 | <!-- two:所有acitivity-state(包括start-state)处需要更新的属性，和用户表单内容对应 --> | |

|  |  |  |
| --- | --- | --- |
| 63 | <attribute name="start date" type="date" /> | |
| 64 | <attribute name="end date" type="date" /> |

|  |  |
| --- | --- |
| 65 | <attribute name="leave days" type="integer" /> |
| 66 | <attribute name="comment" type="text" initial-value="请假理由或者备注" /> | |

|  |  |  |
| --- | --- | --- |
| 67 | <attribute name="Chief evaluation result" type="evaluation" /> | |
| 68 | <attribute name="Boss evaluation result" type="evaluation" /> |

|  |  |  |
| --- | --- | --- |
| 69 | |  |
| 70 | </concurrent-block> | | |

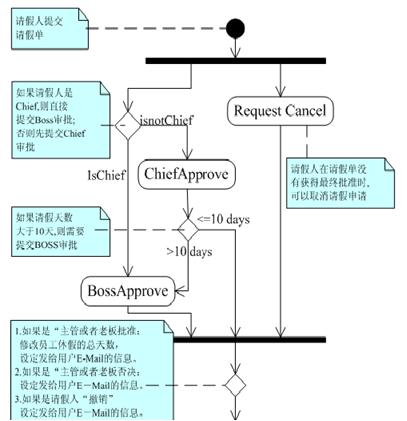
|  |  |  |
| --- | --- | --- |
| 71 |  | |
| 72 | | </process-definition> | |

2. 其它配置文件代码太长就不一一贴出来；  
3. 定义委托类：委托类包含在lib文件夹下的程序集中。

因为委托类数目众多，这里仅贴出几个典型的委托类：

1. NetBpm.Example.LeaveOfAbsence.AutoSetAttributionsAction:该委托类设计为一个通用委托类，这里用来设置表识属性，如流程经过用户取消请假路径，则把RunTrace属性设置为requestercancel，供WhichWayDicision作判断用。  
2. NetBpm.Example.LeaveOfAbsence.AnyOneJoin: 该委托主要用来设置激活父flow机制，这里是只要任何一条路径到达了join，则激活父flow，流程往下流。  
3. NetBpm.Example.LeaveOfAbsence.WhichWayDecision:该委托根据流程实际流过路径，根据标识属性RunTrace等进行走哪条边的抉择。  
4. 本文仅仅是一个示例，给大家提供一个运用nPdl定义NetBPM流程的参考，如果要把该流程投入现实中使用显然还需要做很多优化。其中的代码量还是很大的。

NetBPM下设计的流程图如下：



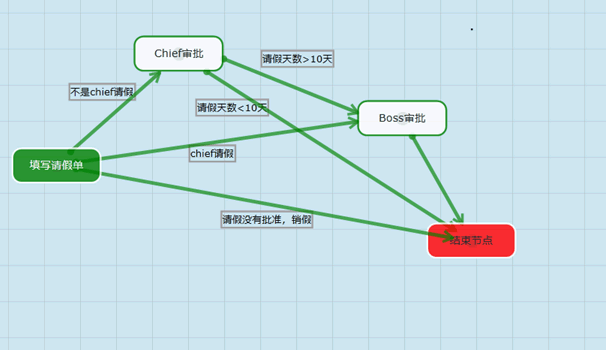
CC Flow

ccflow是一款国产开源工作流。支持SQLServer、Oracle、Access、MySQL数据库，支持群集计算、支持多国语言。流程设计、表单设计都是可视化的，所见即所得。 ccflow提供了强大的数据分析功能：流程运行的各种报表、图形、挖掘、赚取，可以对实（时）效性、成本分析（人力、时间、财物），进行全方位的分析、监控。 Ccflow更可与手机+手机短信+短信猫+电子邮件无缝连接，让您的工作第一时间沟通，第一时间处理。

使用CC Flow创建工作流过程：

1.    在web容器中安装好程序后，打开流程设计器,建立请假流程,即可生成填写请假单和结束节点；  
2.    拖动Chief审批节点、 Boss审批节点，添加连线以及标签注明；  
3.    设置表单：邮件选择傻瓜型表单或者自由类型表单，设置表单后，设置每个节点的工作岗位；  
4.    设置流程跳转方向条件，如判断情人是谁，判断请假天数等，选择的数据来源自表单数据。  
5.    点击运行即可运行流程；可打开windows service，即可使用自带的消息提醒以及邮件发送功能；

CC Flow设计的流程图如下：



综上所述，三款的工作流区别如下表：

