子流程驳回处理

# 概要

在BPMX3目前的实现中，我们没有实现从子流程驳回到主流程，主要的原因是我们驳回的实现采用的是临时修改流程定义，将流程的流转动态进行改变，没有办法从子流程画一条线到主流程。

现在转换了一下思路，还是采用画线的方式来实现。即从子流程节点画一条线指向前面的节点。

# 实现方式

## 实现原理

在子流程中，流程节点在第一个节点进行驳回，这个时候我们就往上找到子流程实例节点，再在子流程节点上画一条线指向之前的节点。

## 核心代码

com.hotent.platform.service.bpm.cmd.CallActivityRejectCmd

|  |
| --- |
| ProcessDefinitionEntity processDefinition = (ProcessDefinitionEntity)  ((RepositoryServiceImpl) repositoryService).getDeployedProcessDefinition(superEnt.getProcessDefinitionId());  ActivityImpl curAct= processDefinition.findActivity(superEnt.getActivityId());  curAct.getOutgoingTransitions().clear();  //创建一个连接  ActivityImpl destAct= processDefinition.findActivity(targetNode);  TransitionImpl transitionImpl = curAct.createOutgoingTransition();  transitionImpl.setDestination(destAct);  superEnt.take(transitionImpl); |

## 修改代码

|  |  |  |
| --- | --- | --- |
| 代码类 | 动作 | 备注 |
| com.hotent.core.bpm.model.ProcessCmd | 修改 | 增加了jump跳转注释 |
| com.hotent.core.bpm.util.ActivitiUtil | 增加 | 获取命令执行器工具类。使用方法：  CommandExecutor commandExecutor=ActivitiUtil.getCommandExecutor();  CallActivityRejectCmd cmd=new CallActivityRejectCmd(taskId);  cmd.setTargetNode(nodeId);  return commandExecutor.execute(cmd); |
| com.hotent.platform.controller.bpm.TaskController | 修改 | 做了一些代码的调整。 |
| com.hotent.platform.controller.mobile.MobileTaskController | 修改 | 代码做了一点重构。 |
| com.hotent.platform.dao.bpm.ExecutionStackDao | 修改 | 增加了一个方法  public void delByParentfId(Long parentId) |
| com.hotent.platform.event.def.CallActivitiBackEvent | 增加 | 子流程驳回时抛出的事件。  在类CallActivityRejectCmd中进行抛出。 |
| com.hotent.platform.event.listener.CallActivityBackListener | 增加 | 监听CallActivitiBackEvent事件，然后进行处理。 |
| com.hotent.platform.model.bpm.ExecutionStack | 修改 | 增加前面和后面的堆栈。 |
| com.hotent.platform.service.bpm.cmd.AllowBackCallActivitiCmd | 增加 | 判断这个是否属于子流程驳回cmd。 |
| com.hotent.platform.service.bpm.cmd.CallActivityRejectCmd | 增加 | 外部子流程指令，使用任务ID作为构造参数。 |
| com.hotent.platform.service.bpm.cmd.TaskAllowRejectCmd | 增加 | 判断任务是否允许驳回。 |
| com.hotent.platform.service.bpm.IBpmActService | 修改 | 增加接口方法  /\*\*  \* 驳回到某个节点。  \* @param taskId  \* @param nodeId  \*/  boolean reject(String taskId,String nodeId );  /\*\*  \* 根据任务ID判断流程是否允许驳回。  \* @param taskId  \* @return  \*/  boolean isTaskAllowBack(String taskId);  /\*\*  \* 判断允许子流程实例是否允许驳回。  \* <pre>  \* 判断当前是否为子流程实例，并且当前节点位于第一个节点。  \* </pre>  \* @param taskId  \* @return  \*/  boolean allowCallActivitiBack(String taskId); |
| com.hotent.platform.service.bpm.impl.BpmActService | 修改 | 实现上类方法。 |
| com.hotent.platform.service.bpm.listener.CallSubProcessEndListener | 修改 | 增加在驳回时修改子流程节点状态。。 |
| com.hotent.platform.service.bpm.thread.TaskThreadService | 修改 | 添加方法。  public static void putVariables(Map<String,Object> vars\_) |
| com.hotent.platform.service.bpm.BpmService | 修改 | 重构跳转方法。  transTo(TaskEntity task, String toNode) |
| com.hotent.platform.service.bpm.BpmTaskExeService | 修改 | 做了部分重构。 |
| com.hotent.platform.service.bpm.BpmTaskUtil | 增加 | 增加转办判断方法。 |
| com.hotent.platform.service.bpm.ExecutionStackService | 修改 | 重构堆栈部分方法。 |
| com.hotent.platform.service.bpm.ProcessRunService | 修改 | 重构代码 |
| com.hotent.platform.service.util.ServiceUtil | 修改 | 重构  去掉无效代码。 |
| ExecutionStack.map.xml | 修改 | 增加根据父id删除方法。 |
|  |  |  |

## 数据库脚本修改

alter table BPM\_EXE\_STACK add constraint FK\_SELF\_PARENTID

foreign key (STACKID) references bpm\_exe\_stack (STACKID) on delete cascade;