

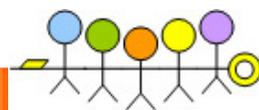


workflow 概念与模型



主要内容

- 为什么需要 workflow 中间件?
- workflow 历史
- 基本概念
- workflow 模型
- workflow 现状与趋势



为什么需要工作流中间件？

■ 场景：

1. 某电信公司需要上马“服务开通系统”项目
2. 系统性能要求：每天处理40万笔业务
3. 上线时间要求：4~6个月开通上线
4. 未来可能会根据市场需要陆续开通更多新的业务

服务开通是由流程驱动的，上万个产品，产品开发到运营、服务都需要流程支持。并且需要通过配置流程来开发新产品



两种可选方案

■ 方案一：传统方式（不采用工作流中间件）

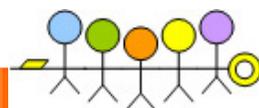
◆ 方案1.1：集成商自带“引擎”

◆ 方案1.2：无引擎：

状态流转与每个活动的处理逻辑完全交织在一起

■ 方案二：基于商业工作流中间件

我们推荐**第二种方案**！



不采用 workflow 中间件 ——存在的问题

■ 集成商自带“引擎”

◆ 功能太弱

- 抽象于OA或某些业务系统，往往只考虑某个问题域的方案，无法满足市场需求变化的要求

◆ 适应能力弱

- 无法满足业务开通系统对一些复杂功能的要求，比如异常处理、业务化定制等

◆ 开放性不够

- 因定制开发，在产生新的需求时，需要进行大量的流程和页面开发工作。
- 无法快速实现对现有流程的调整与优化

◆ 稳定性不够

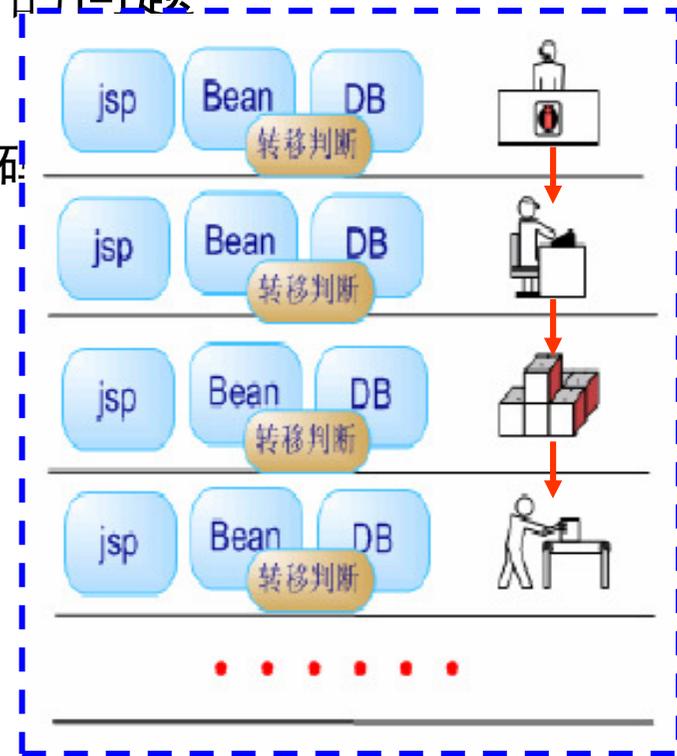
- 因投入不足，无法满足服务开通系统每日处理40万笔业务的需求



不采用 workflow 中间件 ——存在的问题

■ 无引擎

- ◆ 也存在上述“集成商自带引擎”的问题
- ◆ 流转状态固定（硬编码）
 - 一旦需要调整流程，就必须修改代码
- ◆ 参与者固定
 - 业务人员无法调整
- ◆ 流程和页面完全耦合
 - 开发工作量巨大
- ◆ 不能统一管理 and 跟踪流程
- ◆ 无法支持流程优化与改进





如何保障 “服务开通系统” 成功实施？

■ 需要专业的应用架构平台

- ◆ 灵活性强

- ◆ 性能高

■ 需要专业的工作流中间件

- ◆ 高性能

- ◆ 高可扩展性

- ◆ 支持复杂流程，特别是异常处理机制

- ◆ 业务化定制：小的调整业务人员就可以完成



主要内容

- 为什么需要 workflow 中间件?
- workflow 历史
- 基本概念
- workflow 模型
- workflow 现状与趋势



workflows 的历史

- workflow 技术起源
 - ◆ 上个世纪70年代中期办公自动化领域的研究工作。SCOOP, Officetalk系统
- 75-85期间
 - ◆ OA的研究, 迅速发展
- 80s初期
 - ◆ workflow 技术走向商用 (Image, Doc, Email), 但是很少
- 80s后期
 - ◆ OA的研究基本逐渐消失, 代替的是群件 (Groupware) 和 Workflow Management。
- 九十年代以后
 - ◆ 相关的技术条件逐渐成熟, workflow 系统的开发与研究进入了一个新的热潮。 workflow 技术被应用于电讯业、软件工程、制造业、金融业、银行业、科学试验、卫生保健领域、航运业、办公自动化领域。

(from 银狐99)



工作流的历史（续）

■ WfMC（Workflow Management Coalition）

- ◆ 1993.8 Founded
- ◆ 1995.1 Workflow Reference Model (1003)
- ◆ 1998.11 Workflow Process Definition Interchange .WPDL(1016)
- ◆ 2002.10 XPDL1.0
- ◆ 2005.10 XPDL2.0 .Support BPMN.



workflow 标准

■ BPMI (Business Process Management Initiative)

- ◆ 2000 创建,15个创始成员。
- ◆ 2002.9 BPML Published
- ◆ 2004.5 BPMN Published
- ◆ 2004.6 认可BPEL4WS
- ◆ 2005.6 合并到OMG



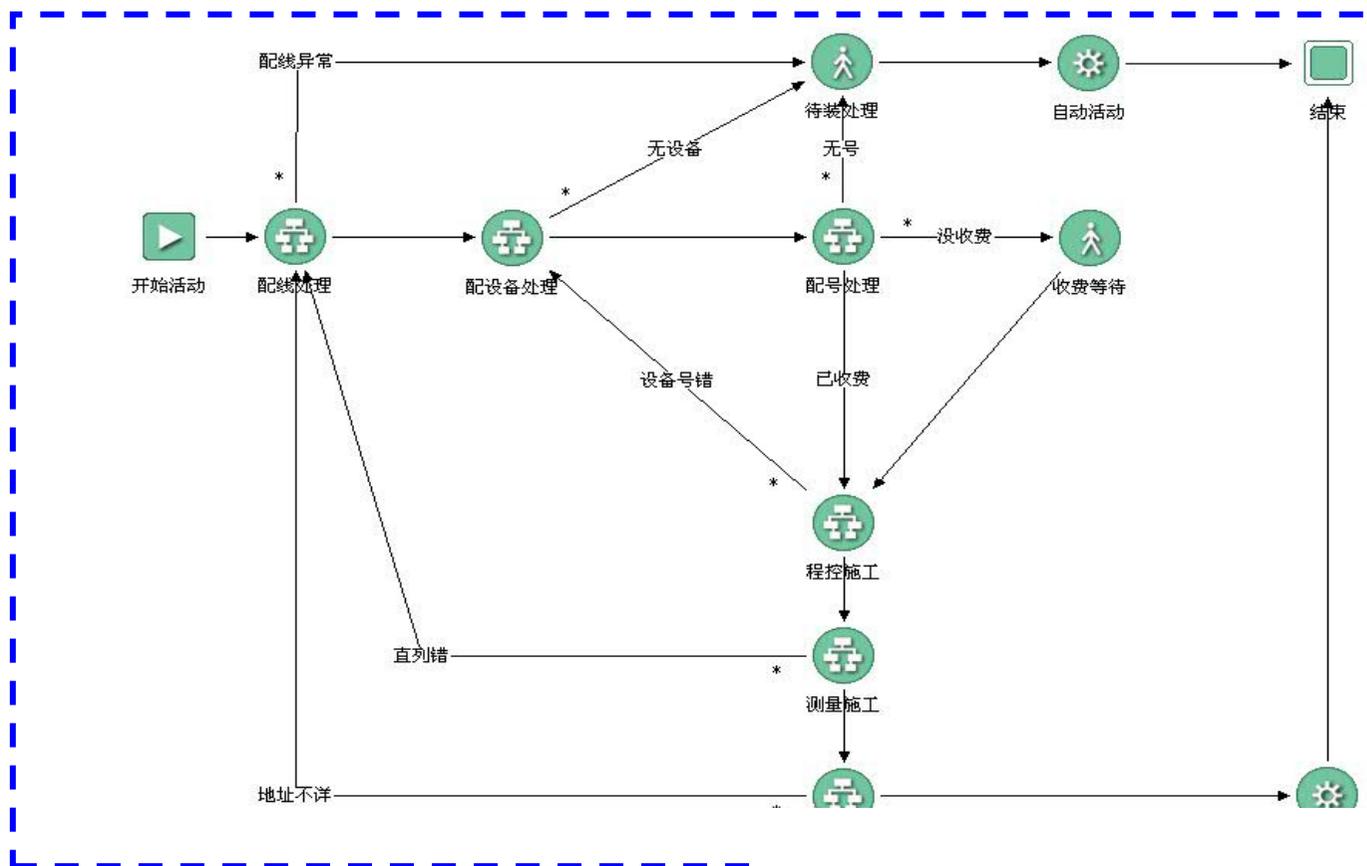
主要内容

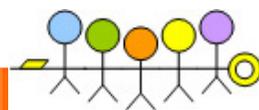
- 为什么需要 workflow 中间件？
- workflow 历史
- 基本概念
- workflow 模型
- workflow 现状与趋势



工作流的概念

■ “某省电信服务开通”的例子





工作流的概念

■ WfMC的定义

- ◆ **工作流(Workflow)**: 是一个业务流程, 可以部分或全部地自动执行, 并且执行时可以按照一定的过程规则将文档、信息或任务从一个参与者传递给下一个执行的参与者。
- ◆ **工作流管理系统(WfMS)**: 是一个通过使用软件来定义、创建和管理工作流执行的系统。这个软件运行在一个或多个工作流引擎中, 而且能够解释流程定义, 与工作流参与者进行交互, 并在需要的时候调用IT工具或应用。



workflows的概念

■ 业务流程:

“业务流程是有组织的活动， 相互联系， 为客户
创造价值”

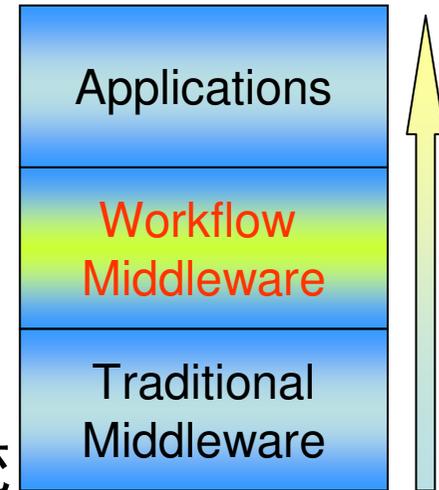
——Hammer教授



workflow 概念

■ workflow 中间件

- ◆ 也是一种 workflow 管理系统
- ◆ 介于应用和普通中间件之间
- ◆ 良好的开放性
 - 可以根据客户需要建立不同的业务系统
- ◆ 高性能



■ workflow 中间件的目的:

- ◆ 在正确的时间将正确的任务按照正确的顺序分配给正确的人员



workflows 的概念

■ workflow 中间件的内涵:

- ◆ 流程定义 (Process Definition)
- ◆ 流程执行环境 (Process Engine)
- ◆ 流程监控与管理 (Administration & Monitor)



工作流的概念

- 活动 (Activity)
- 参与者 (Participant)
- 转移线 (Transition)
- 转移条件 (Transition Condition)
- 时间限制 (Time Limit)
- 相关数据 (Relevant Data)



workflow 概念 ——活动 (Activity)

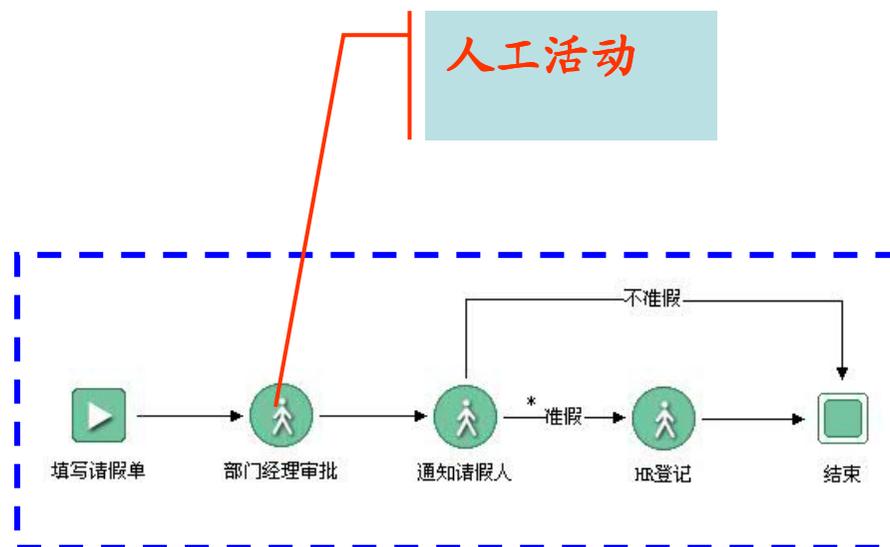
■ 概念

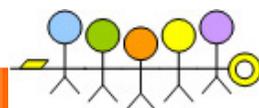
- ◆ 又叫节点、环节或者任务， workflow 是由一个个活动按照一定顺序组成的

■ 分类

- ◆ 开始活动、结束活动
- ◆ 人工活动、自动活动
- ◆ 路由活动
- ◆ 子流程活动

■ 前驱活动与后继活动





workflow 概念 ——活动的属性

- 名称
- 参与者
- 时间限制
- 触发事件
- 激活策略
- 聚合模式
- 分支模式



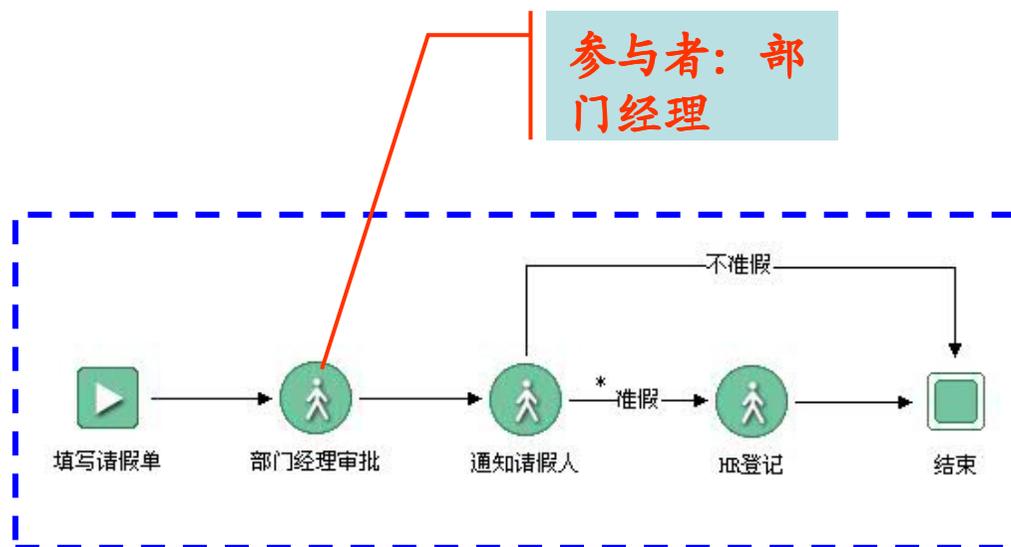
workflow 概念 ——参与者 (Participant)

■ 概念

- ◆ 表示执行某个活动的资源

■ 分类

- ◆ 人
- ◆ 组织
- ◆ 角色
- ◆ 系统（或应用）
- ◆ 动态计算





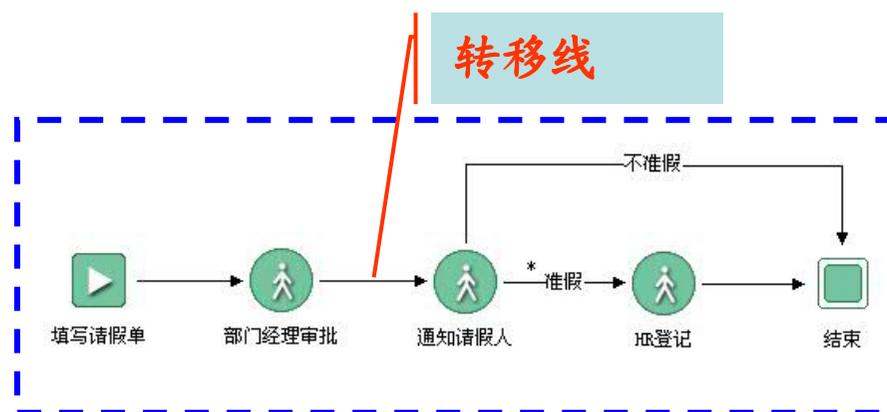
workflow 概念 ——转移线 (Transition)

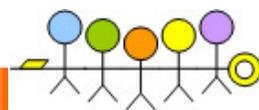
■ 概念

- ◆ 两个活动之间的连接线，表面活动之间的状态迁移
- ◆ 转移线上可以定义转移条件 (Transition Condition)

■ 分类

- ◆ 无条件转移
- ◆ 有条件转移
- ◆ 缺省转移线





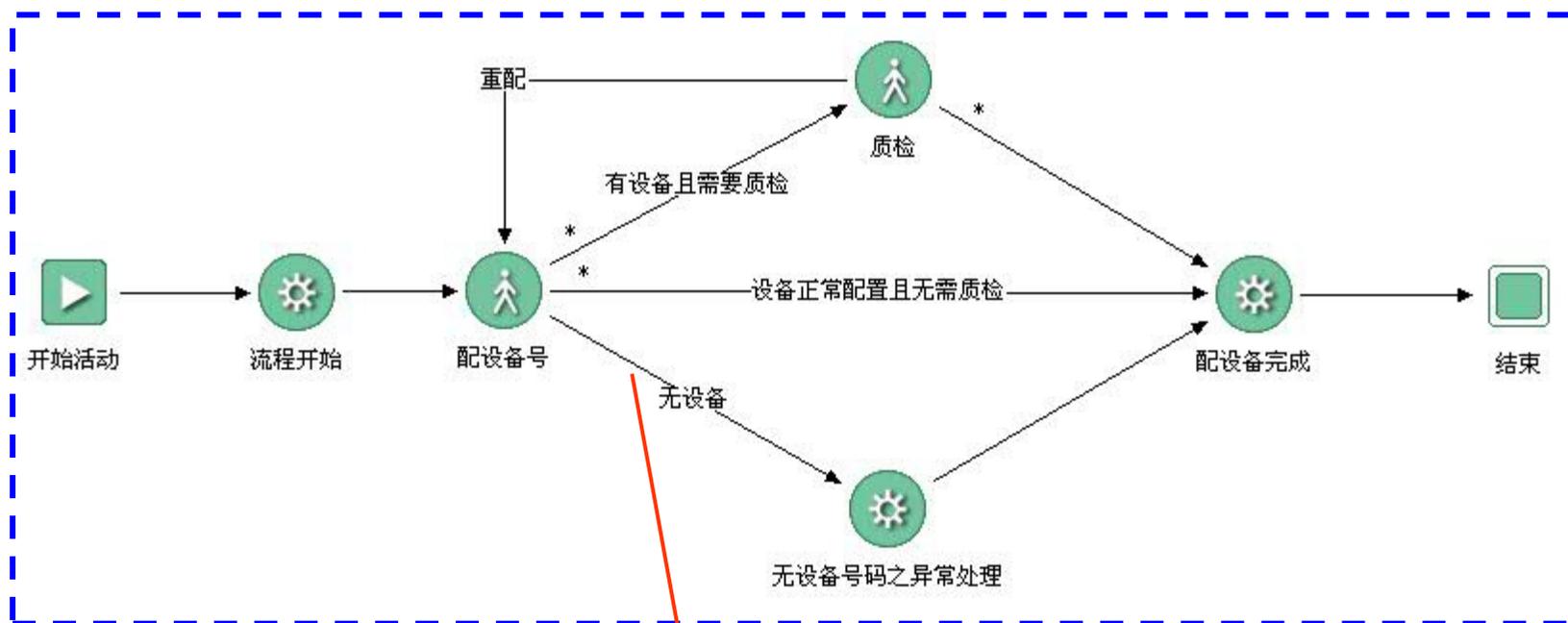
workflow 概念 —— workflow 相关数据

- workflow 相关数据（Workflow Relevant Data）
主要指与业务流程相关的数据。
- WfMC 的定义：
 - ◆ workflow 管理系统通过 workflow 相关数据来确定流程实例状态转移条件，并选择下一个将执行的活动。这些数据可以被 workflow 应用程序访问并修改。因此，workflow 管理软件需要在活动实例之间传递 workflow 相关数据。



workflow 相关数据 —— 举例

■ 配设备:



workflow 相关数据:
有无设备、是否需要



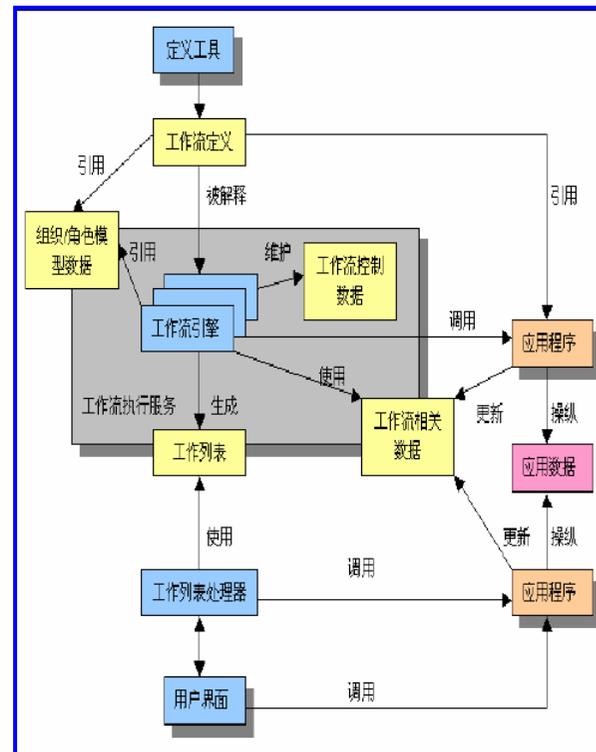
主要内容

- 为什么需要 workflow 中间件？
- workflow 历史
- 基本概念
- workflow 模型
- workflow 现状与趋势



workflow 模型

- 路由模型
- 激活模型





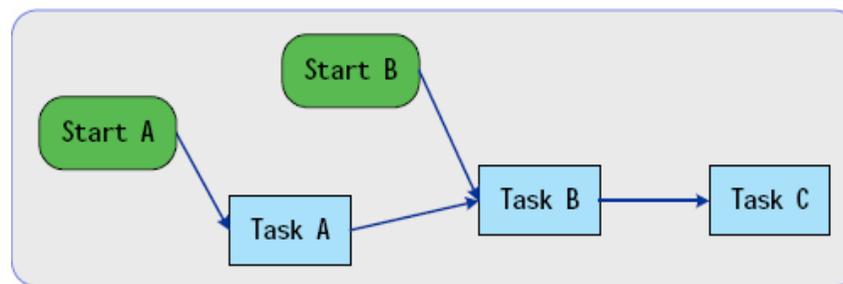
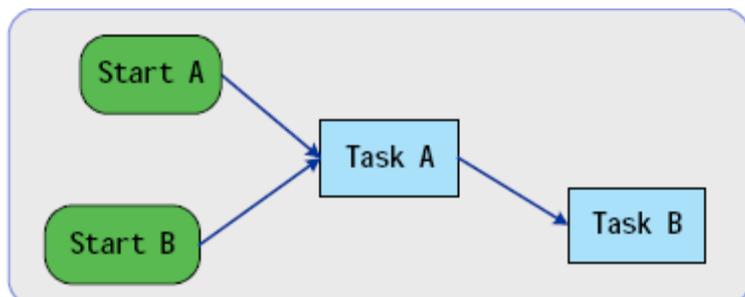
workflow 模型 —— 起点模型

■ 单起点

- ◆ 流程只有一个开始活动

■ 多起点

- ◆ 流程有多个开始活动



(此图from 银狐99)



工作流模型 ——路由模型

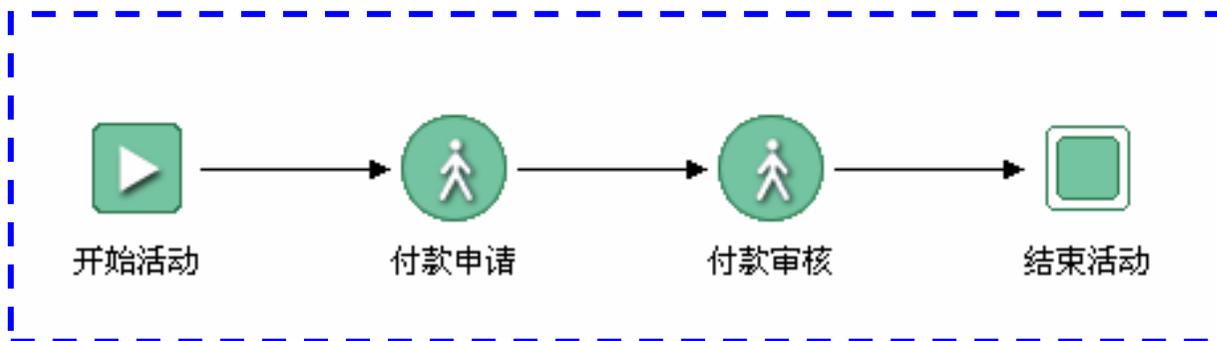
- 基本路由
- 分支路由
- 聚合路由
- 循环路由
- 嵌套路由
- 任意路由

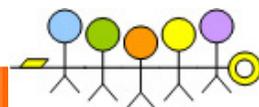


路由模型 ——基本路由

■ 串行路由

- ◆ 每个活动都只有一个前驱和后一个后继活动，其中开始活动只有一个后继活动，结束活动只有一个前驱活动。





路由模型 ——分支路由

■ 分支路由概念（Split）

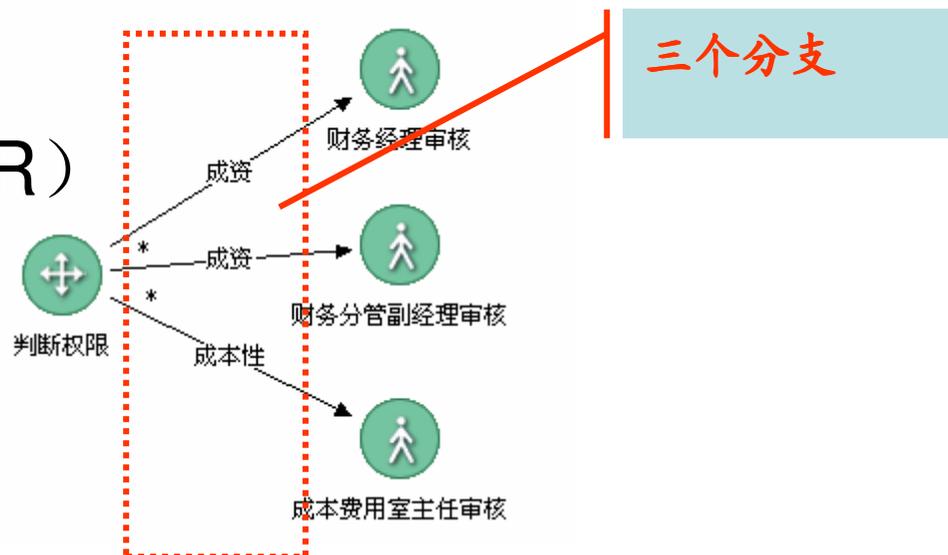
◆ 指当一个活动存在多个后继活动时的路由策略：

■ 分支路由分类：

◆ 并行路由（AND）

◆ 独占式选择（XOR）

◆ 多路分支（OR）

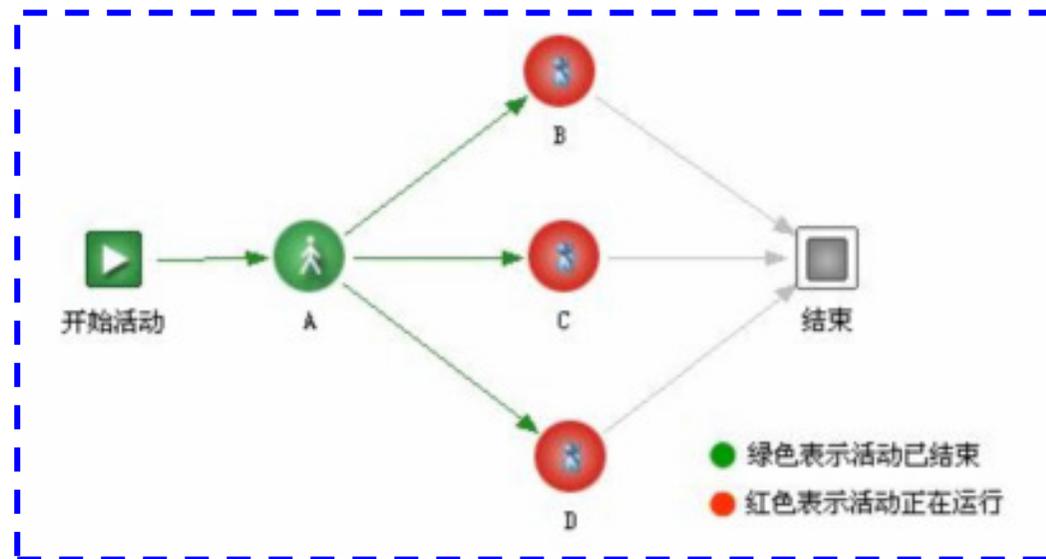




分支路由 ——并行

■ 并行路由

- ◆ 表示当前活动结束后它的**所有后继**活动将同时被执行

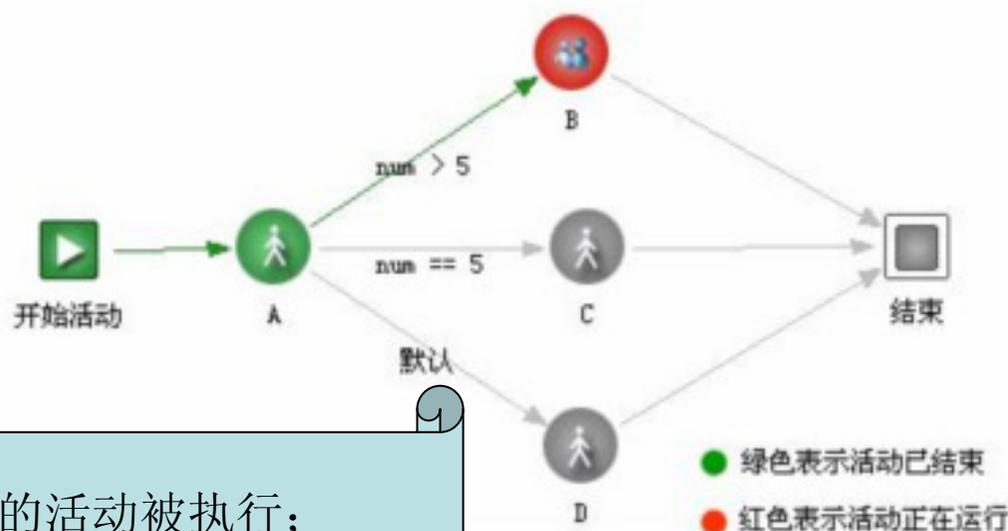




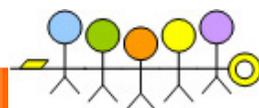
分支路由 ——独占式选择

■ 独占式分支

◆ 根据迁移条件自动选择一个后继活动执行：



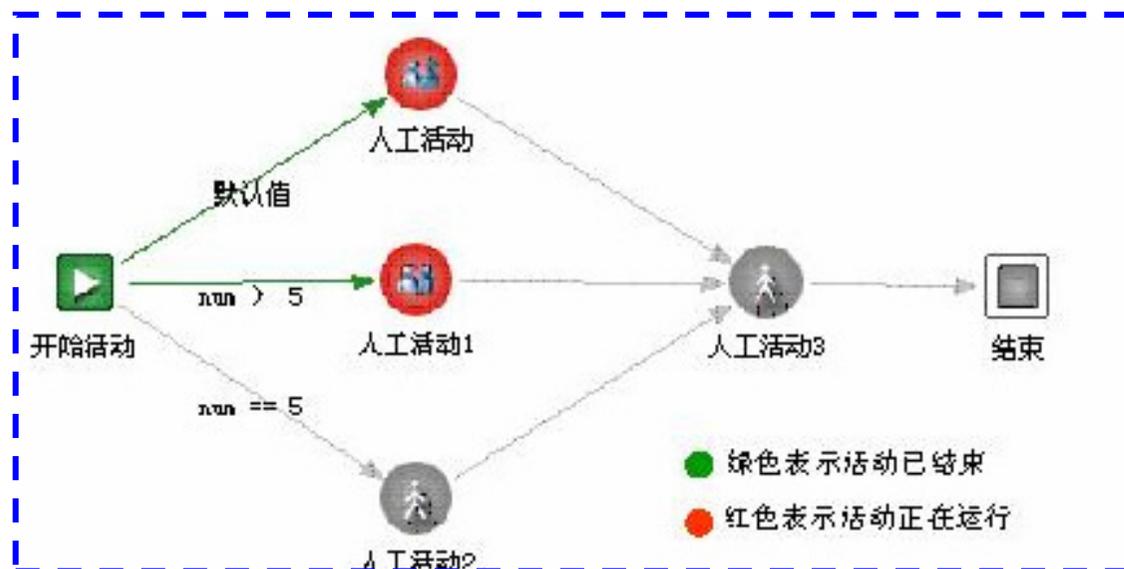
- 1) 只有一个满足条件的活动被执行；
- 2) 如果有多个条件满足，则优先级高活动将被执行；
- 3) 如果条件都不满足，则执行“默认”路由。



分支路由 ——多路分支

■ 多路分支

- ◆ 所有满足条件的后继活动都被执行
- ◆ 如果没有满足条件的活动，则执行缺省活动





路由模型

——聚合路由

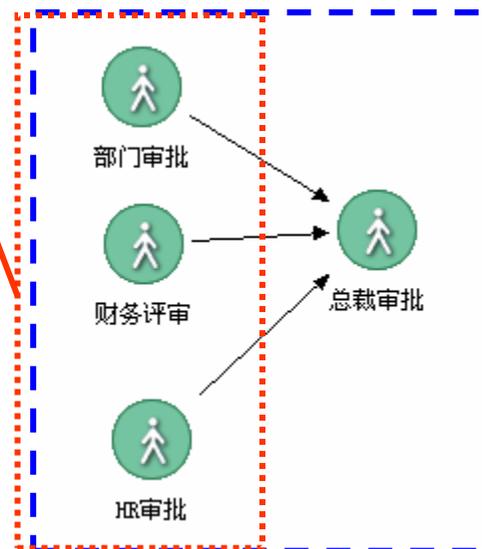
■ 聚合路由概念（Join）

- ◆ 当一个活动存在多个前驱活动会聚时的路由策略

■ 聚合路由分类

- ◆ 全部聚合（AND）
- ◆ 单一聚合（XOR）
- ◆ 多路聚合（OR）

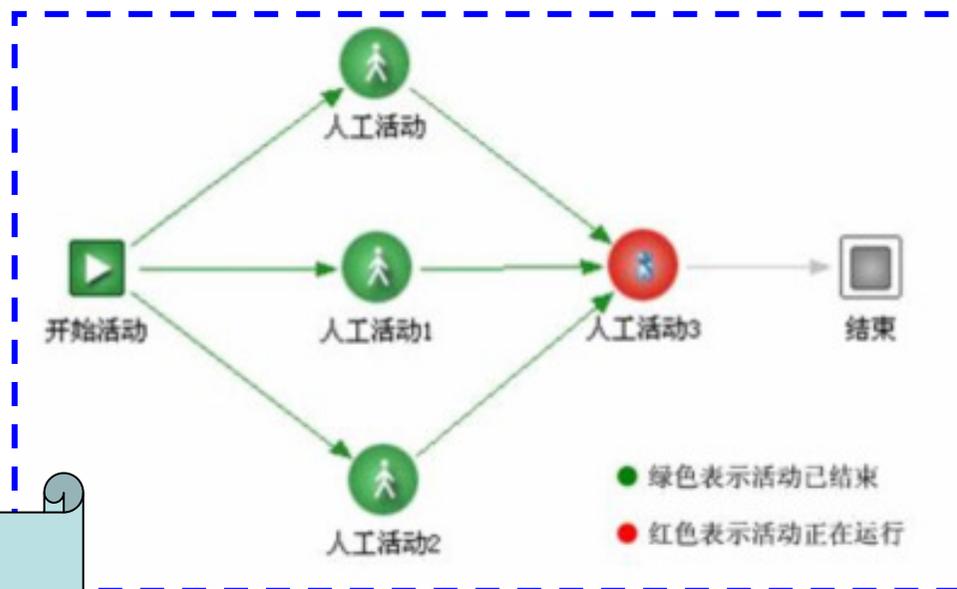
“总裁审批”
的前驱活动



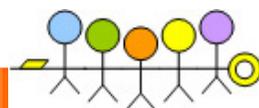


聚合路由 ——全部聚合

- 该活动必须等到它的所有前驱活动全部完成才可以被执行

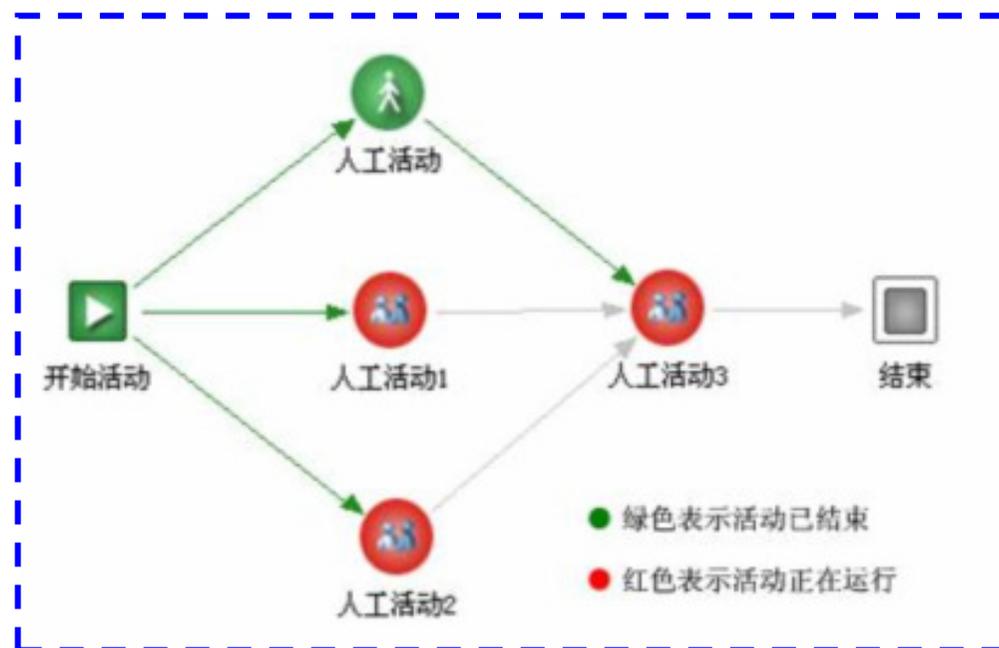


常用于并联审批



聚合路由 ——单一聚合

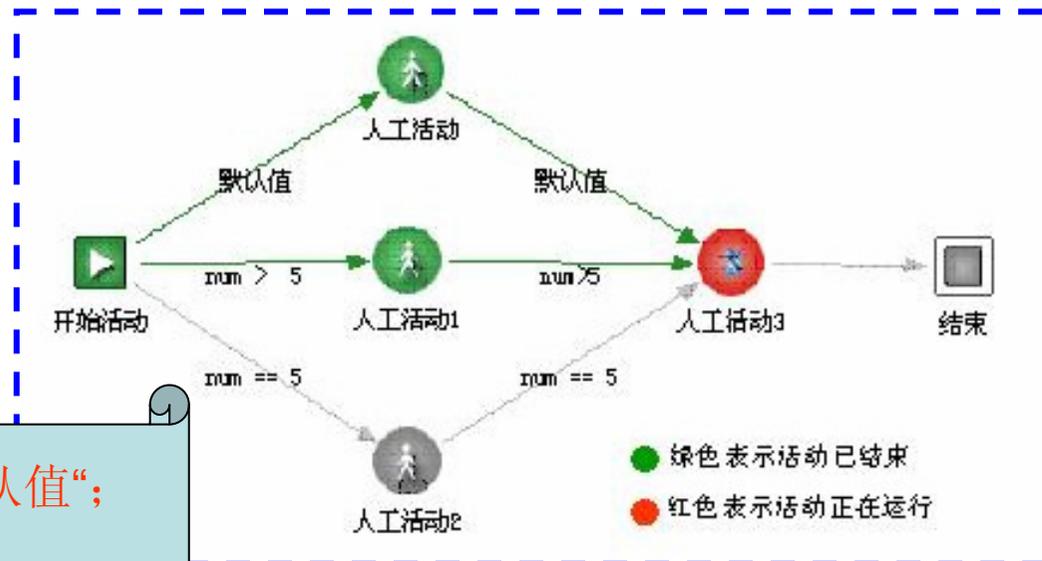
- 某活动的若干前驱活动中，**只要有一个正在执行的活动完成**，下一活动即可被执行





聚合路由 ——多路聚合

- 某活动必须等到它的所有满足条件的正在执行的前驱活动全部完成才可以执行



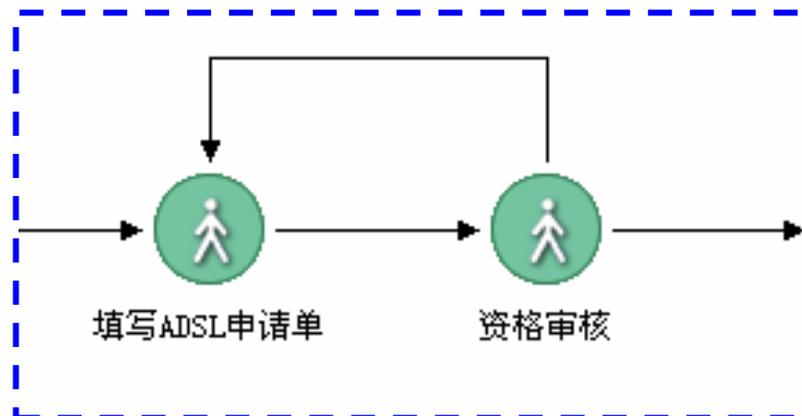
- 1) 与该活动的连线是“默认值”;
- 2) 与该活动连线上条件为“true”;



路由模型 ——循环路由

■ 概念

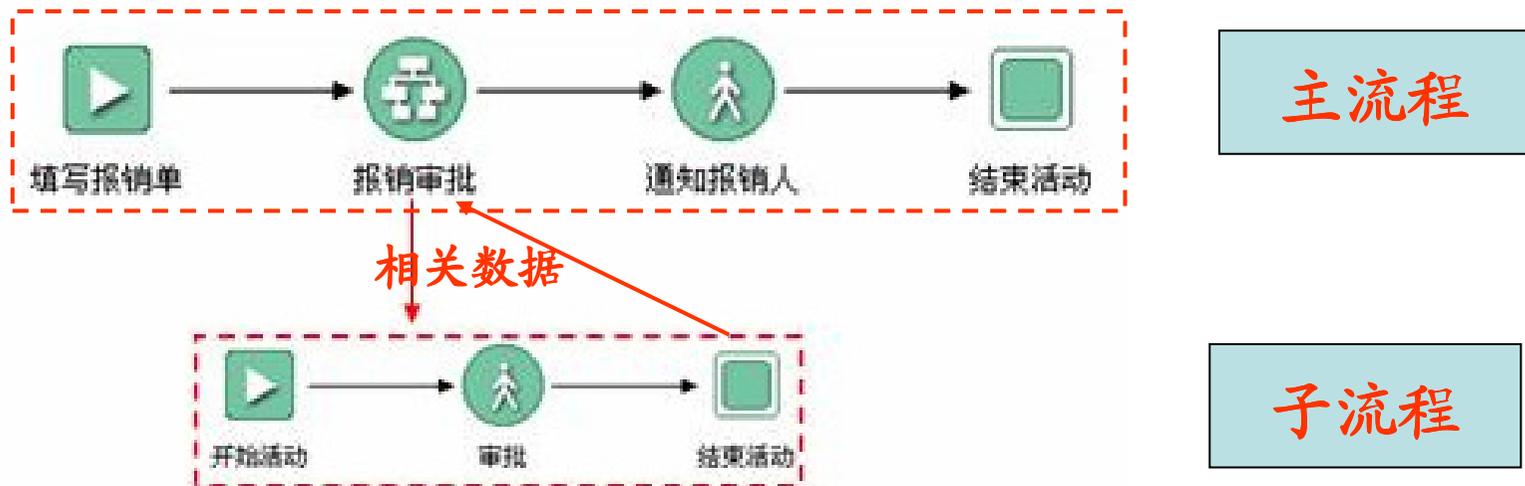
◆ 表示某个活动在一定条件下会被多次重复执行





路由模型 ——嵌套路由

- 嵌套路由指在一个工作流中嵌套执行另一个工作流。又叫“子流程路由”

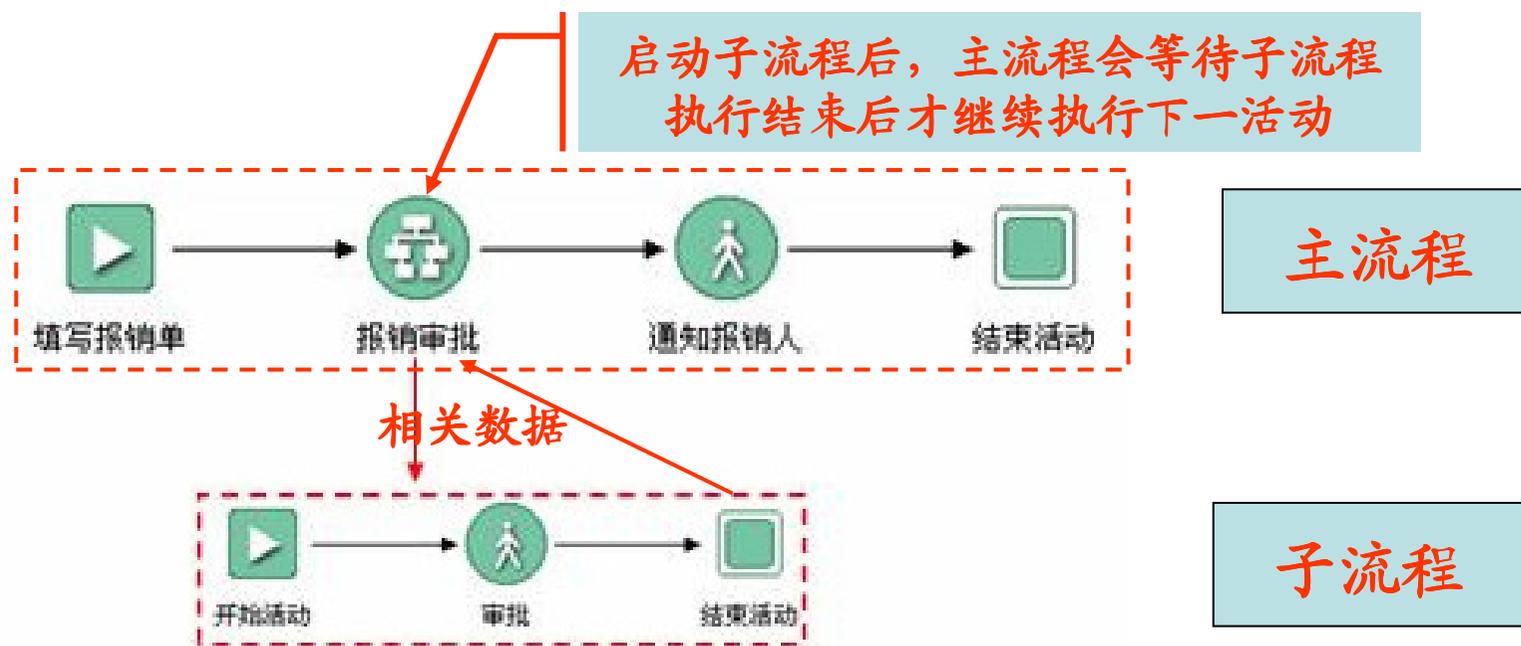




嵌套路由 ——嵌套策略

■ 同步执行

◆ 等待子流程运行完成后，该子流程活动才结束

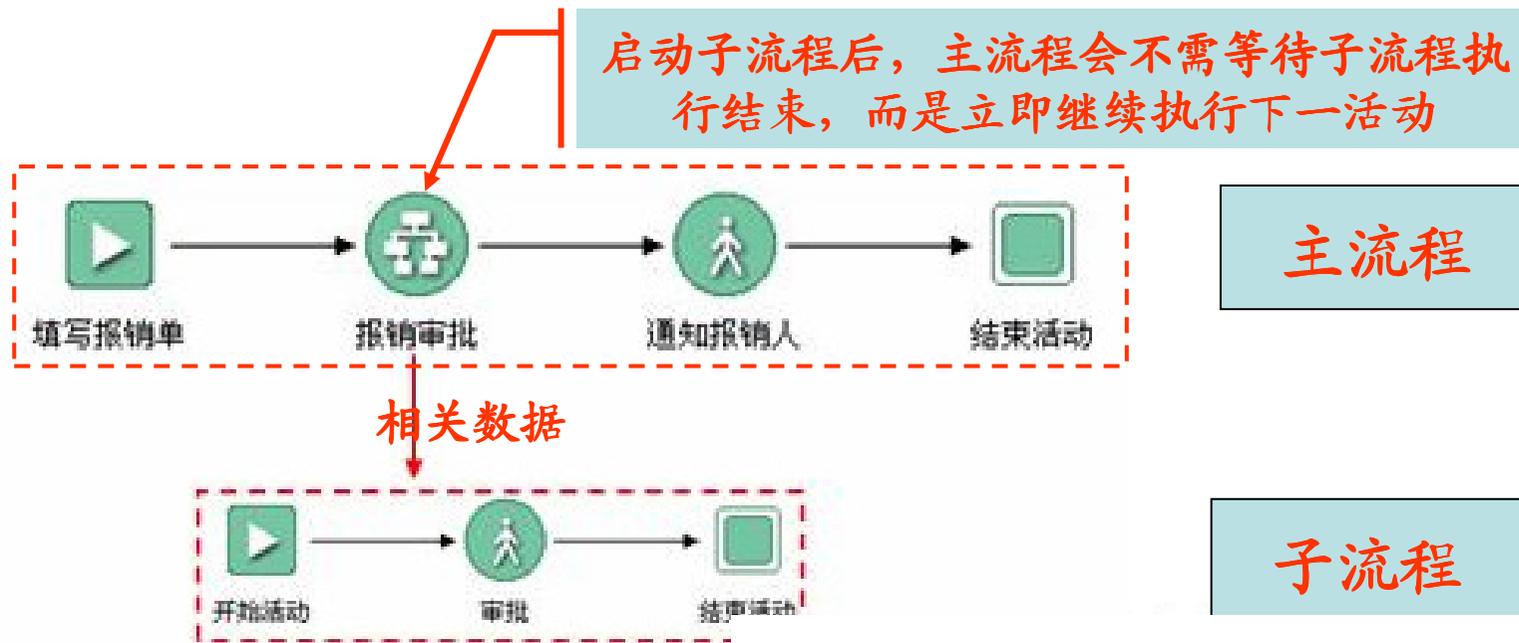




嵌套路由 ——嵌套策略

■ 异步执行

- ◆ 当前活动在启动完子流程后就结束，而无需等待子流程运行完成





工作流模型 ——激活模型

- 流程激活
- 活动激活



流程激活

- 人工激活
- 定时激活
- 消息激活



流程激活

■ 人工激活

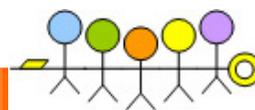
- ◆ 大多数的流程激活，都是因为人为的信息数据输入或产生。比如一个订单处理流程，客户提交了订单信息（订单信息数据产生），则激活了订单处理流程的开始



流程激活

■ 定时激活

- ◆ 在一个特定的时间，因为特定的情况，符合特定的条件，激活某个特定的流程（或任务）。
- ◆ 这种激活方式，在现实中很少单独出现，大多数情况，都因为在某一个流程中，因为在限定的时间内，因某项任务未达到预期的状态，而激活另外的任务或新的处理流程。也就是说，这种方式，是受外来因素影响的，而且大多与一些流程任务（或流程模式）一起出现。
- ◆ 举例：
 - 订单处理流程，限定5天内发货，那么定义在第三天的时候，如果没有接到发货通知，则激活一个催办信息（催办任务）。这样流程系统，会在第三天的时候自动发出催办信息



流程激活

■ 消息激活

- ◆ 多用在两个流程之间，一个流程在某种状态下发出一个消息，激活另一个流程。
- ◆ 举例：
 - ADSL新装业务，如果申请用户没有安装电话，则ADSL新装流程会发出一个消息启动“电话新装”流程



激活模型

——活动激活

■ 立即激活

- ◆ 现实中大部分都是这种活动

■ 等待激活

- ◆ 暂时不激活等待人工或者通过API激活

■ 条件激活

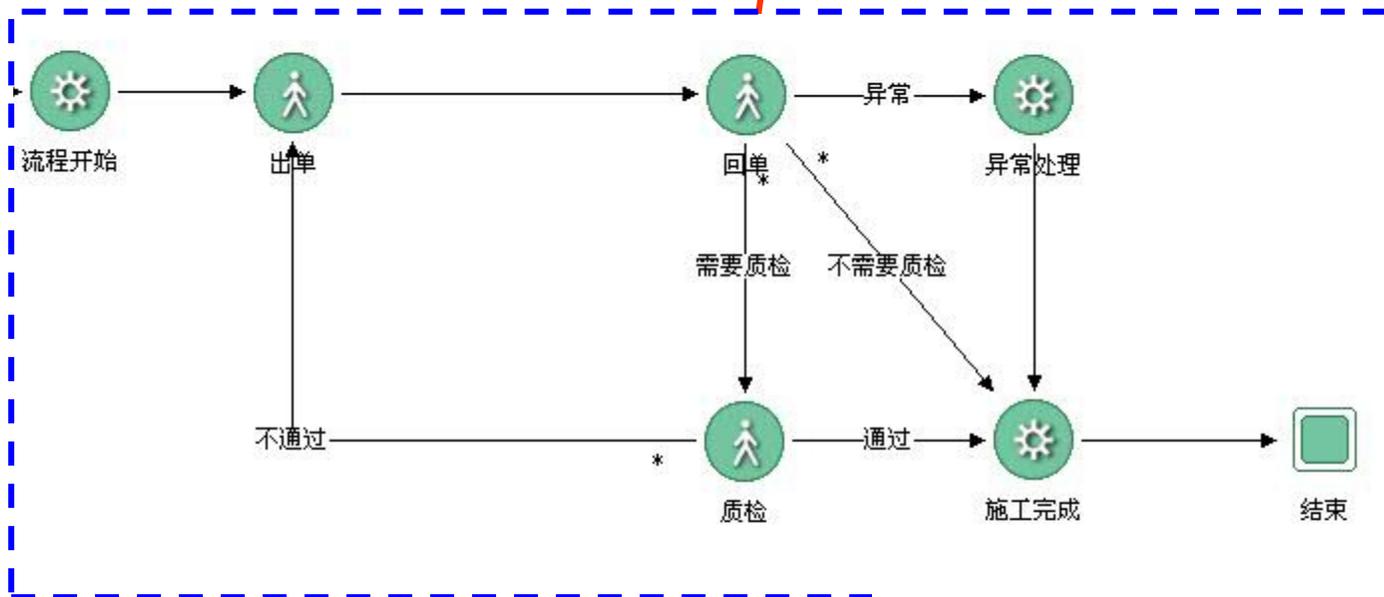
- ◆ 如果满足某个条件则激活，否则不激活



激活模型 ——立即激活例子

立即激活

“出单”活动结束后，会立即执行“回单”活动

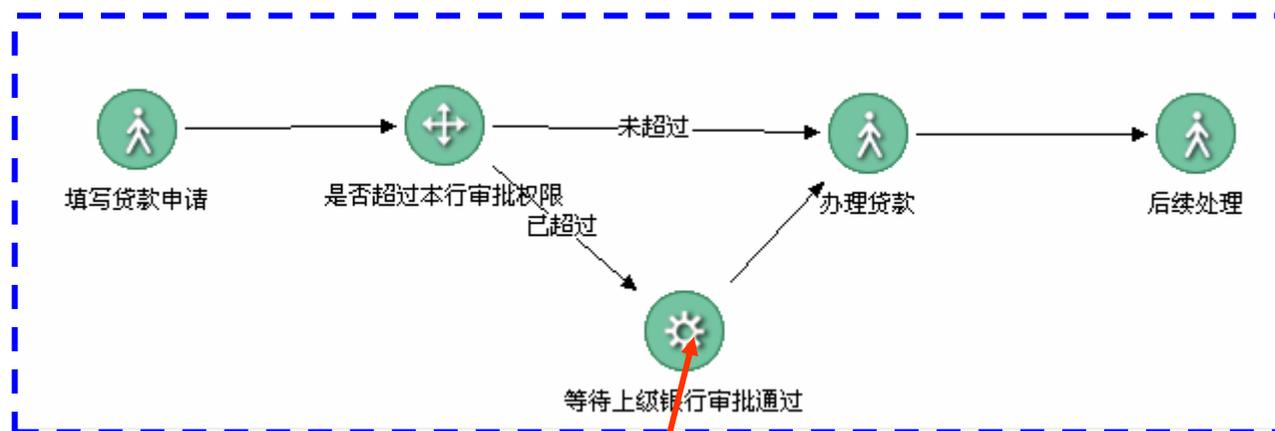




激活模型 ——等待激活例子

■ 等待激活

- ◆ 一般用于需要等待其它流程的处理结果，或者需要等待外部应用的处理结果的情况



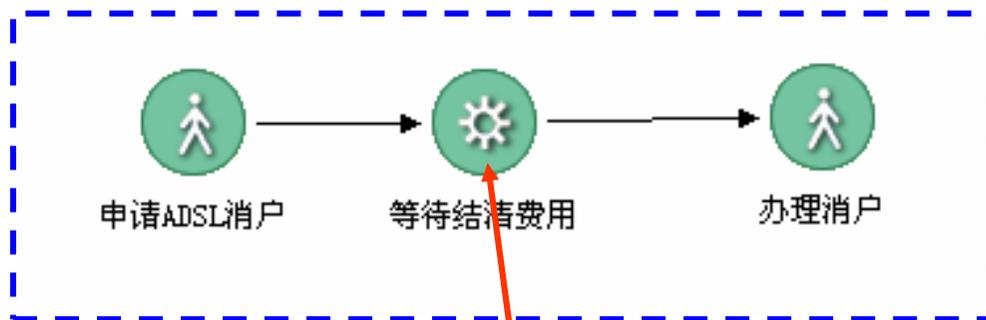
在贷款审批流程中，如果申请额度超过本行的审批权限，则需要上报到更高级别的银行审批，而上级银行的审批流程不在本行，则需要等待上级银行审批后，发送消息或者人工的方式通知继续执行流程



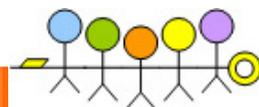
激活模型

——条件激活例子

■ 条件激活



ADSL消户业务中，如果要进行消户，需要先等待用户结清费用，才运行消户。此处可以调用一个API判断是否结清费用，如果没有结清，则该活动处于等待状态。



workflow模型 ——EOS特有的模型

- 任意路由
- 版本控制
- 回退处理
- 自由流
- 代理/代办
- 异常处理



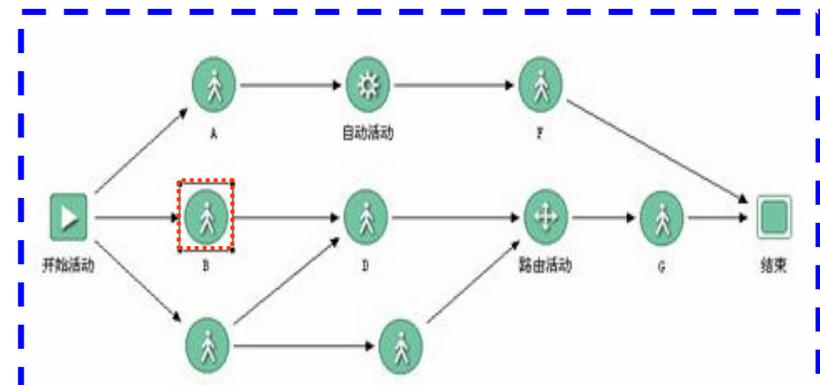


EOS特有的模型 ——任意路由

- 是指当一个活动运行完成后，被启动、运行的后继活动不是由引擎根据转移条件判断，而是由当前活动的参与者动态指定，可以是一个也可以是多个

■ 策略

- ◆ 在流程内任意自由
- ◆ 在指定范围内自由
- ◆ 在后继活动范围内自由

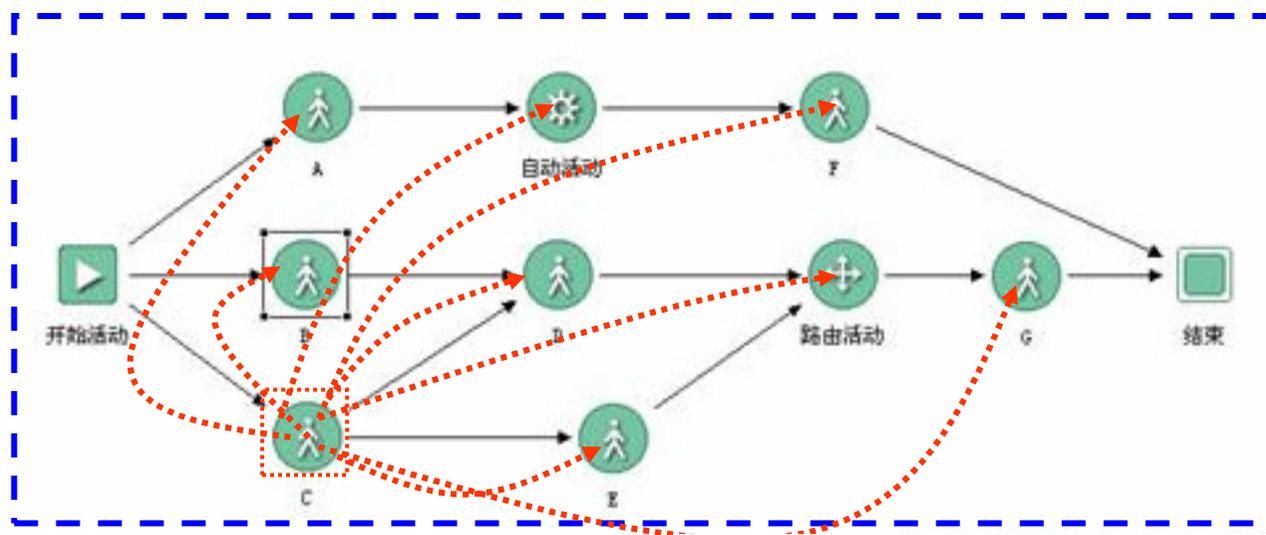




任意路由 ——策略

■ 在流程内任意自由

- ◆ 运行时该活动的后继活动可以是除开始活动和本身以外的流程内的所有活动。

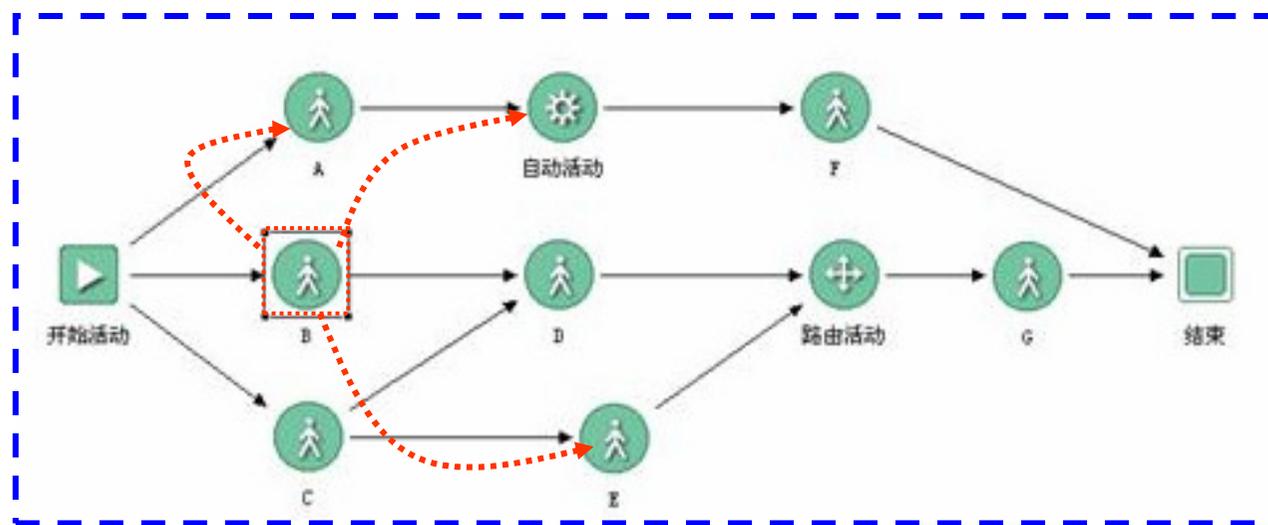




任意路由 ——策略

■ 在指定范围内自由

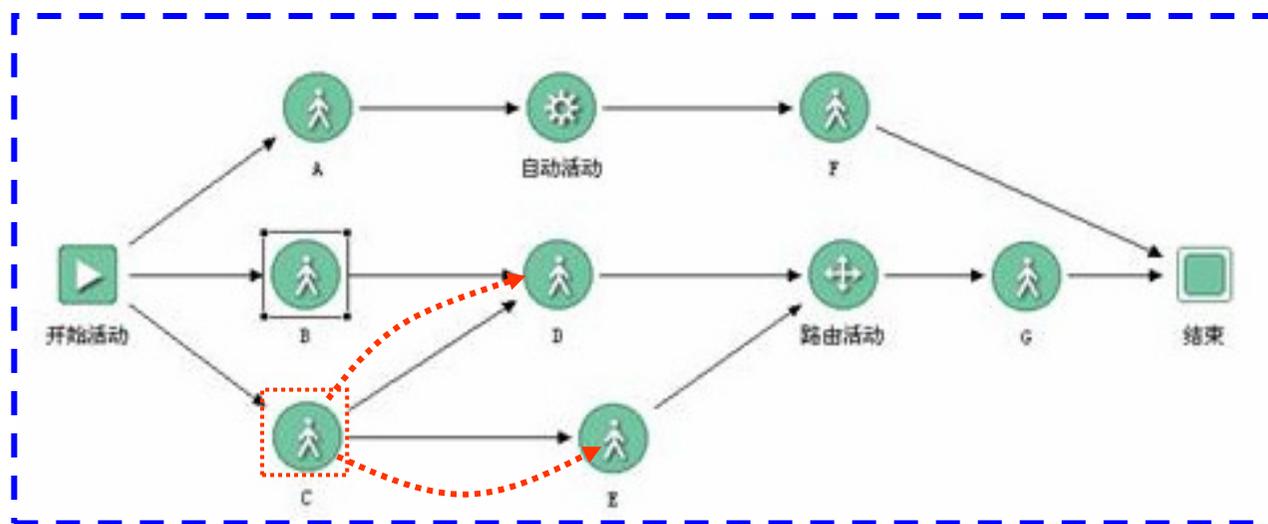
- ◆ 定义时开发人员根据业务需求从流程中圈定若干活动作为运行时可选择的后继活动的范围。

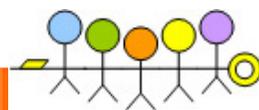




任意路由 ——策略

- 在后继活动范围内自由：
 - ◆ 运行时该活动可选的后继活动范围为它的直接后继活动。





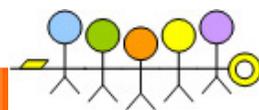
EOS特有模型 ——版本控制

■ 概念

- ◆ 每个流程都允许同时存在多个版本，用户可以根据需要（如业务发生变化或者政策发生变化时）对版本进行升级。

■ 版本控制策略

- ◆ 特事特办型——个别流程实例的流程调整
- ◆ 一刀切型——所有新、旧流程实例都按新的流程运行
- ◆ 分水岭型——未结束的流程实例，仍按旧的流程运行



EOS特有模型 ——回退

■ 概念

- ◆ 回退指的是当流程运行到某个活动时，由于一些原因要将流程退回到已经执行过的某个前驱活动上
- ◆ 回退是一种动态循环
 - 只有在运行过程中才会明确的知道将要回退到哪个活动
- ◆ 回退往往需要执行业务补偿



工作流模型 ——小结

■ 流程路由模型

◆ 串行路由

◆ 分支路由

◆ 聚合路由

◆ 循环路由

◆ 嵌套路由

◆ **EOS特有模型**：回退、自由流、异常处理、代理代办、版本控制。。。

■ 激活模型



主要内容

- 为什么需要 workflow 中间件?
- workflow 历史
- 基本概念
- workflow 模型
- workflow 现状与趋势



workflows的发展现状

■ 厂商情况

◆ 国外 workflow

■ 如IBM、BEA等

◆ 集成商自己开发的 workflow 平台

◆ 商用 workflow 中间件

■ 代表：普元、协同等



国外工作流产品的现状

- 定位问题：
 - ◆ 解决集成问题，需求来自于欧美市场
- 性能差
 - ◆ 堵单严重（江苏电信），跟其定位有关，主要是为了解决集成问题
- 功能弱
 - ◆ 很多中国特色的功能不支持
 - 如回退、异常处理、业务化定制、代理/代办等
 - ◆ 缺乏应用框架的配合
 - ◆ 很多功能只能在应用系统中实现，需求变化时工作量大
- 扩展性不好
 - ◆ 难以实现业务化的定制



国内工作流产品的现状

- 国内集成商或者工作流厂商
 - ◆ 投入不足
 - ◆ 性能问题
 - 不具有支撑每日处理40万笔业务的能力
 - ◆ 缺乏复杂流程支持
 - 如多个流程同步等待、异常处理能力等



普元EOS workflows的特点

- 高性能
- 高可扩展性
- 对复杂流程的支持
- 中国特色
 - ◆ 回退处理
 - ◆ 代理代办
 - ◆ 自由流
 - ◆ 版本控制



workflows的发展趋势

- 市场趋势
- 标准趋势



工作流现状与趋势

——趋势

■ 商业化

- ◆ 产品逐渐由集成商自用向商业产品转变
- ◆ 部分集成商改用商业产品，节省成本

■ 专业化

- ◆ 工作流逐渐由专业厂商提供从产品、服务一条龙服务
- ◆ 随着专业厂商的投入越来越大，工作流产品的功能越来越强、性能越来越高

■ 普及化

- ◆ 市场认可度越来越高
- ◆ 高端应用几乎都会采用工作流平台



workflow 标准的发展趋势

- **WfMC—— workflow 标准的霸主**
 - ◆ 将对OMG的标准进行更多的支持
- **Workflow与BPM逐渐融合**
 - ◆ 相互补充，产品功能逐渐融合
- **BPMI和OMG合并后将会推动主流标准**
 - ◆ BPMN
 - ◆ BPDM
- **实施SOA的最高境界**



总结

■ 为什么需要工作流

- ◆ 业务人员更好的把握业务流程
- ◆ 管理人员统一管理和跟踪流程
- ◆ 利于对流程的持续优化与改进

■ 工作流的概念

- ◆ 4 R: 把正确的任务在正确的时间按照正确的顺序分配给正确的人

■ EOS工作流的优势

- ◆ 高性能
- ◆ 高可扩展性
- ◆ 支持复杂流程
- ◆ 支持业务化定制



■ Q&A



谢谢！！