

执行BPMN

Presented by
Koen Aers

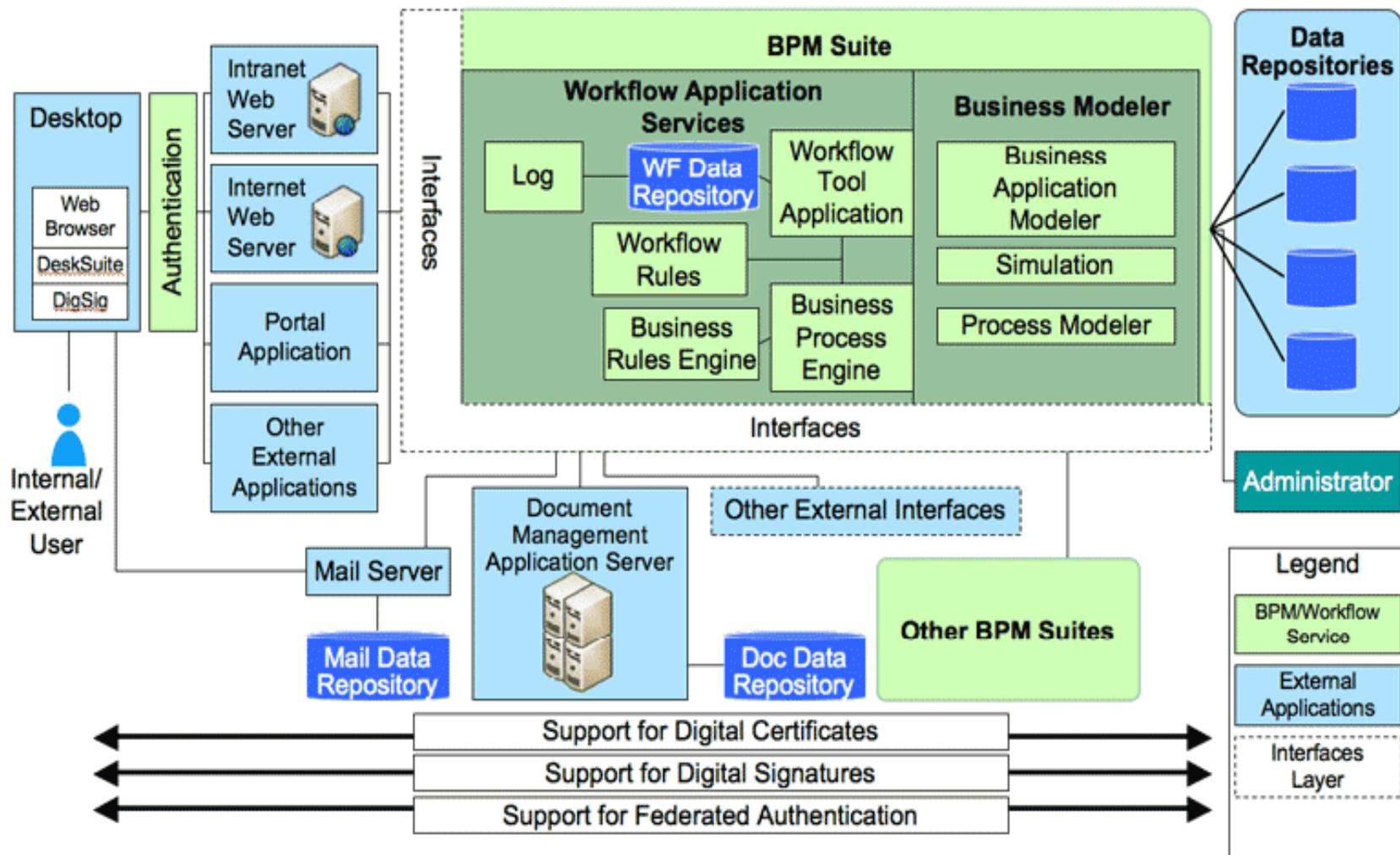
议程

- BPM丛林
- BPMN概览
- BPMN执行场景
- jBPM和jPDL
- 用jBPM执行BPMN

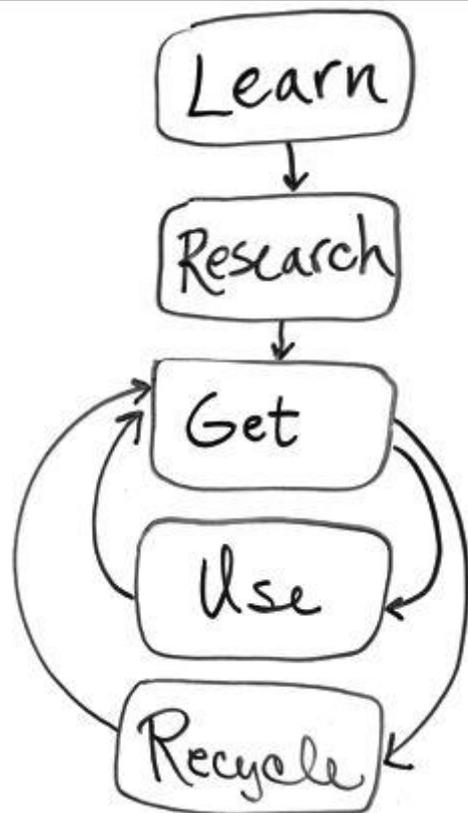
议程

- BPM丛林
- BPMN概览
- BPMN执行语义
- jBPM和jPDL
- 用jBPM执行BPMN

业务流程管理

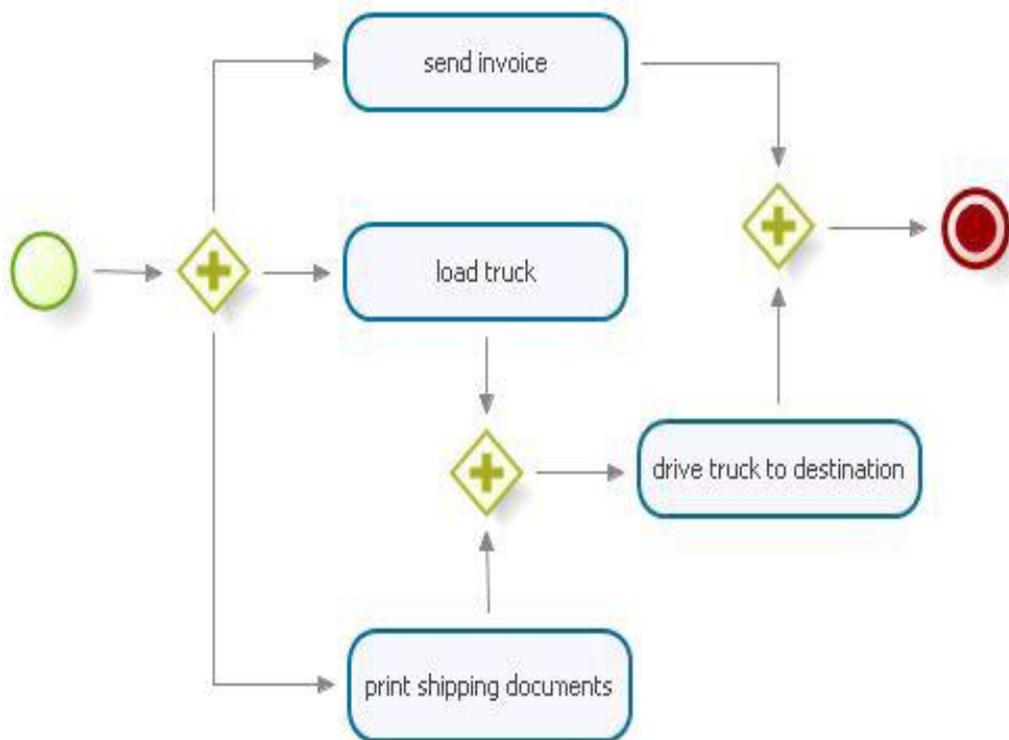


业务流程建模



- 业务流程建模
 - 一个或多个业务流程的模型
 - 定义操作怎样实现
 - 描述工作流和集成
- 摘要
- 工作流== 一系列操作的描述
- (软件?) 工程训练

业务流程执行



■用图来描述的某一方面软件

■执行“流”

■BPMS运行流程执行

■软件

✓领域特定语言

✓在某个系统上的执行

✓集中分发

✓多方参与

✓等待状态

■BUT:流程执行的7个谬论

BPM规范

■ 1995:

- 1 standardization group for workflow models
- Reference model + 5 interface standards
- Size of the average specification : +/- 40 pages

■ 2007:

- 10+ working groups with interest in BPM
- 7+ standards for process models alone
- Size of the average specification: +/- 150 pages

■ OMG:

- CORBA WF, BPMN, BPDM

■ OASIS:

- ebXML (or BPSS), BPEL

■ WfMC:

- XPDL, Wf-XML

议程

- BPM丛林
- BPMN概览
- BPMN执行语义
- jBPM和jPDL
- 用jBPM执行BPMN

事件

Events



An event is something that »happens« during the process. These events affect the flow of the process and usually have a cause (trigger) or an impact (result).
Examples: 'Email received', '3 o'clock', 'Warehouse empty', 'Critical error',...

Event flow Event type	Event flow			Description
	Start	Intermediate	End	
General				The Start Event indicates where a particular process will start. Intermediate Events occur between a Start Event and an End Event. It will affect the flow of the process, but will not start or (directly) terminate the process. The End Event indicates where a process will end.
Message				A message arrives from a participant and triggers the Event. This causes process to (start, continue, end) if it was waiting for a message, or changes the flow if exception happens. End type of message event indicates that a message is sent to a participant at the conclusion of the process.
Timer				A specific time or cycle can be set that will trigger the start of :he Process or continue the process. Intermediate timer can be used to model the time-based delays.
Error				This type of End indicates that a named Error should be generated. This Error will be caught by an Intermediate Event within the Even: Context.

事件

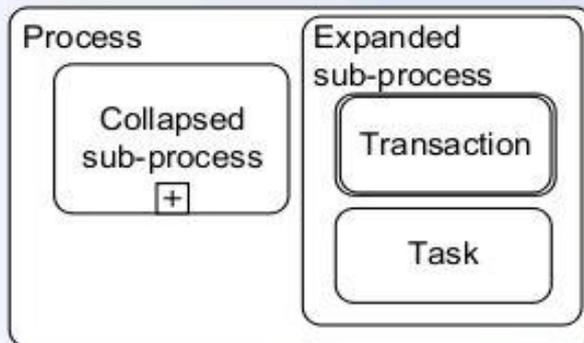
Cancel			This type of Event is used within a Transaction Sub-Process. This type of Event MUST be attached to the boundary of a Sub-Process. It SHALL be triggered if a Cancel End Event is reached within the Transaction Sub-Process.	
Compensation			This is used for compensation handling—both setting and performing compensation. It calls for compensation if the Event is part of a Normal Flow. It reacts to a named compensation call when attached to the boundary of an activity. Very useful for modelling roll-back actions within the transaction.	
Rule			This type of event is triggered when the conditions for a rule become true. Rules can be very useful to interrupt the loop process, for example: 'The number of repeats = N'. Intermediate rule is used only for exception handling.	
Link				A Link is a mechanism for connecting the end (Result) of one Process to the start (Trigger) of another. Typically, these are two Sub-Processes within the same parent Process. It can be used, for example, when the working area (page) is too small – go to another page.
Multiple				This type of event indicates that there are multiple ways of triggering the Process. Only one of them will be required to {start, continue, end} the Process.
Terminate				This type of End indicates that all activities in the Process should be immediately terminated. This includes all instances of Multi-Instances. The Process is terminated without compensation or event handling.

活动

Activities



An activity is a generic type of work that a company performs. An activity can be atomic (task) or compound (process, sub-process).
Examples: 'Send a letter', 'Write a report', 'Calculate the interests',...



A task is used to represent the activity on the lowest abstraction level.



More information about the transaction and compensation attribute can be found under »Compensation Association«.

Task/Subprocess special attributes

Looping		The task or sub-process is repeated.
Ad Hoc		The tasks in the sub-process can not be connected with sequence flows at design time.
Multiple instances		Multiple instances of task or sub-process will be created.
Compensation		The symbol represents a compensation task or sub-process.

网关

Gateways



A gateway is used to split or merge multiple process flows. Thus it will determine branching, forking, merging and joining of paths. Examples: 'Condition true? – yes/no', 'Choose colour? – red/green/blue',...

Gateway control types

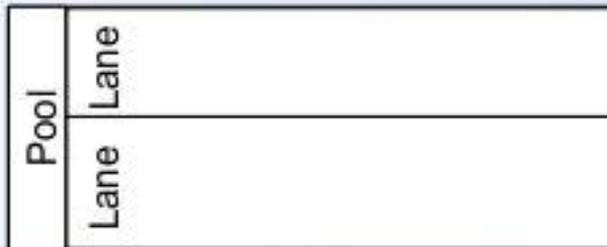
XOR (DATA)	 	Data based exclusive decision or merging. Both symbols have equal meaning. See also Conditional flow. 
XOR (EVENT)		Event based exclusive decision only.
OR		Data based inclusive decision or merging.
COM- PLEX		Complex condition (a combination of basic conditions)
AND		Parallel forking and joining (synchronization).

泳道

Swimlanes



Pools and lanes are used to represent organizations, roles, systems and responsibilities. Examples: 'University', 'Sales division', 'Warehouse', 'ERP system',...



A Pool **MUST** contain 0 or 1 business process.

A Pool can contain 0 or more lanes.

Two pools can only be connected with message flows.

A **Pool** represents a participant in a process. It contains a business process and is used in B2B situations.

A **Lane** is a sub-partition within a pool used to organize and categorize activities.

人工制品

Artefacts

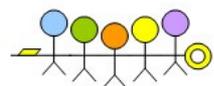
Artefacts are used to provide additional information about the process. If required, modellers and modelling tools are free to add new artefacts.



Examples of data objects: 'A letter', 'Email message', 'XML document', 'Confirmation',...

Set of standardized artefacts

Data object	 [state]	Data objects provide information about what activities are required to be triggered and/or what they produce. They are considered as Artefacts because they do not have any direct effect on the Sequence Flow or Message Flow of the Process. The state of the data object should also be set.
Group		Grouping can be used for documentation or analysis purposes. Groups can also be used to identify the activities of a distributed transaction that is shown across Pools. Grouping of activities does not affect the Sequence or Message Flow.
Annotation	[Description]	Text Annotations are a mechanism for a modeller to provide additional information for the reader of a BPMN Diagram.



图形化的连接对象

Graphical connecting objects



There are three ways of connecting **Flow objects (Events, Activities, Gateways)** with each other or with other information – using sequence flows, message flows or associations.

Graphical connecting objects

Normal
sequence flow



A Sequence Flow is used to show the order in which the activities in a process will be performed.

Conditional
sequence flow



A Sequence Flow can have condition expressions which are evaluated at runtime to determine whether or not the flow will be used.

Default
sequence flow



For Data-Based Exclusive Decisions or Inclusive Decisions, one type of flow is the Default condition flow. This flow will be used only if all other outgoing conditional flows are NOT true at runtime.

Message flow



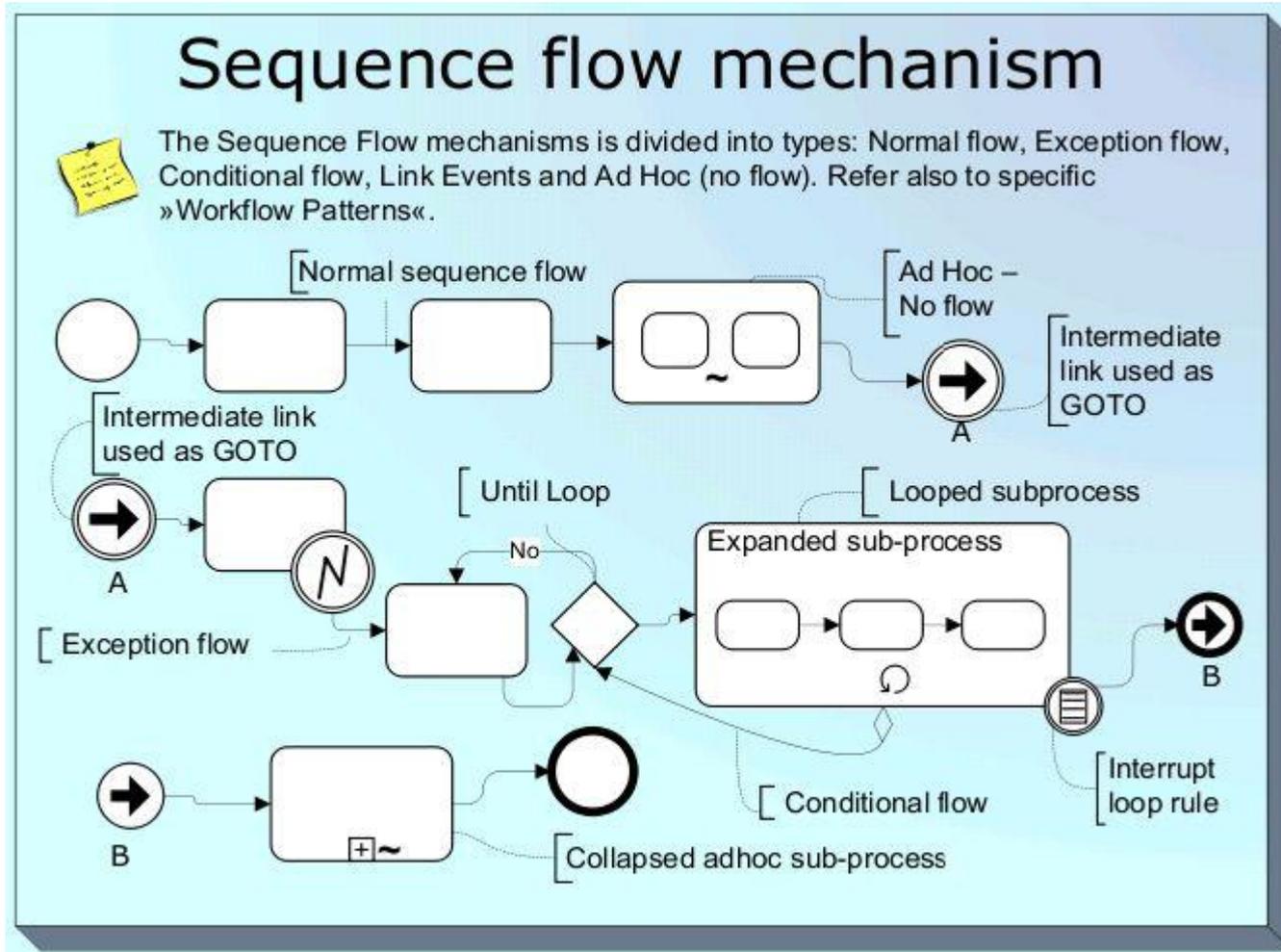
A Message Flow is used to show the flow of messages between two participants that are prepared to send and receive them. In BPMN, two separate Pools in a Diagram can represent the two participants.

Association



An Association (directed, non-directed) is used to associate information with Flow Objects. Text and graphical non-Flow Objects can be associated with Flow objects.

序列流机制

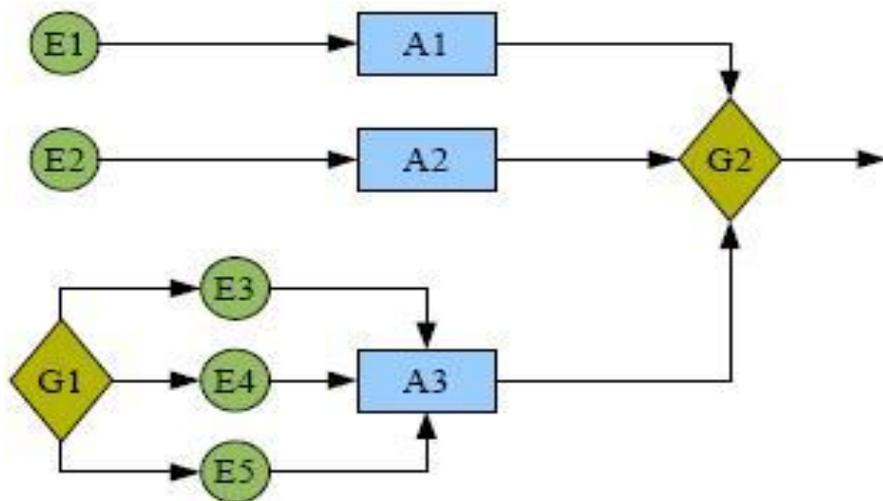


议程

- BPM丛林
- BPMN概览
- **BPMN**执行语义
- jBPM和jPDL
- 用jBPM执行BPMN

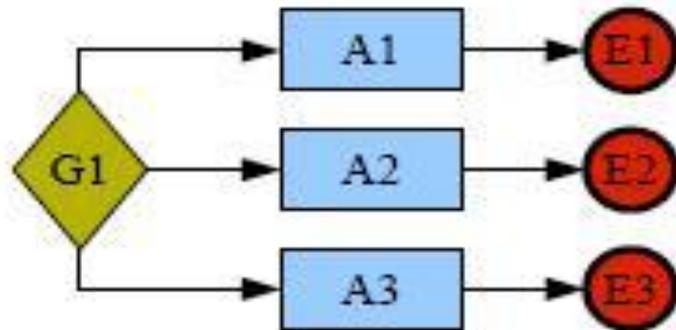
流程初始化

- 当一个开始事件发生时
- 经过基于事件的网关没有传入序列流
- 开始事件的每次发生都在输出序列流上创建一个令牌



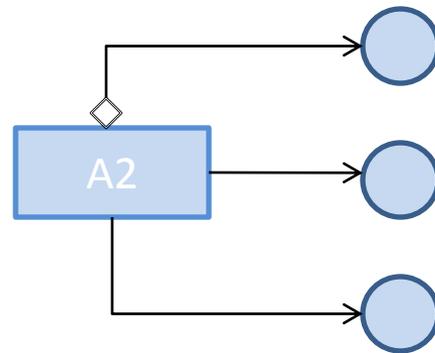
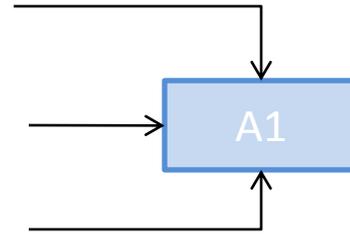
流程终止

- 所有的开始节点都被访问
- 流程实例没有遗留的令牌
- 没有激活的活动
- 所有的令牌必须到达结束节点

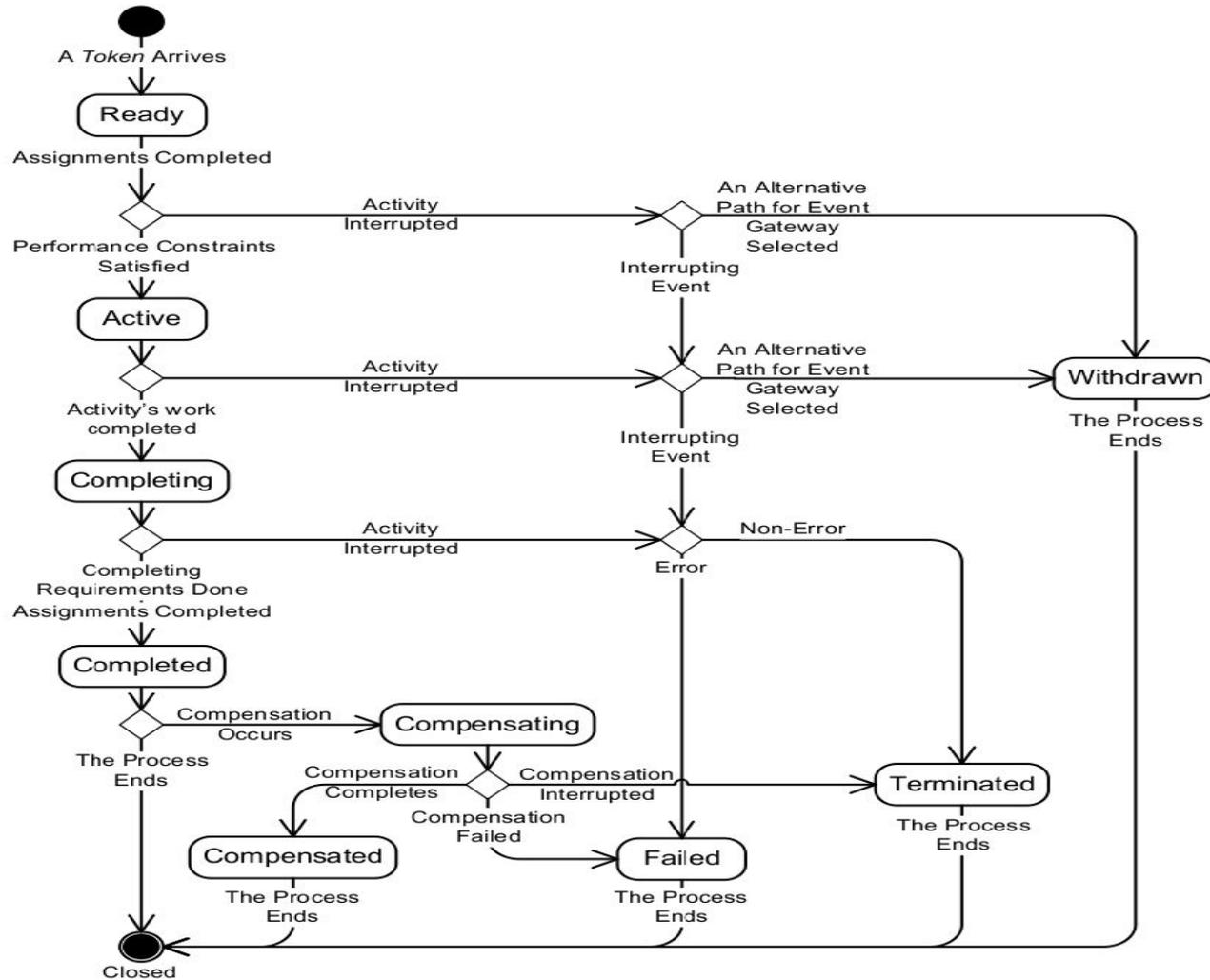


顺序流

- 多个输入序列流
 - 独占的网关
- 无输入序列流的活动
 - 当流程实例创建时被初始化
- 多个输出序列流
 - 为每个流创建一个令牌
 - 条件化的序列流



任务



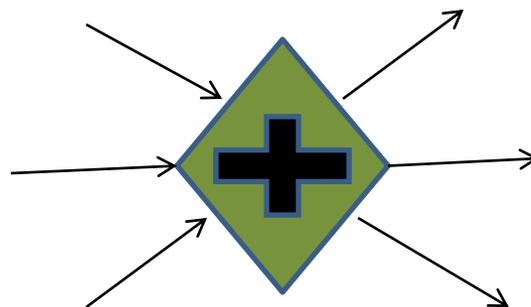
网关

■ 并行网关

- 同步并行分支
- 产生新的并发执行路径

■ 语义

- 每个输入流上至少有一个令牌
- 为输入流确切地消费一个令牌，为输出流正确地产生一个令牌



事件

- 开始事件
 - 启动流程
- 介入事件
 - 等待直到事件发生
- 介入边界事件
 - 消费事件并且执行事件处理器
- 结束事件
 - 中止没有正常结束的流程
 - 其它时间执行他们自己的行为（例如发送消息），并且如果满足下述条件结束流程
 - 所有的开始节点都被访问过了
 - 没有遗留的令牌了

议程

- BPM丛林
- BPMN概览
- BPMN执行语义
- **JBPM和jPDL**
- 用**JBPM**执行BPMN

jBPM和PVM（流程虚拟机）

- jBPM是一个流程语言平台
- 流程虚拟机（PVM）是一个通用的底层基础
- jPDL、BPEL、XPDL基于其上实现



jBPM的进化

产品

jBPM4

- 活动的可插入性的API
- 基于命令模式的服务
- 多语言
- 许多的执行模式
- 数据库分割
- 数据库进化
- 完全的可嵌入性

jBPM3

- 首次可插入性的活动
- 首次多语言
- 首次企业级的嵌入性

jBPM2

- 标准的java嵌入
- Hibernate状态机

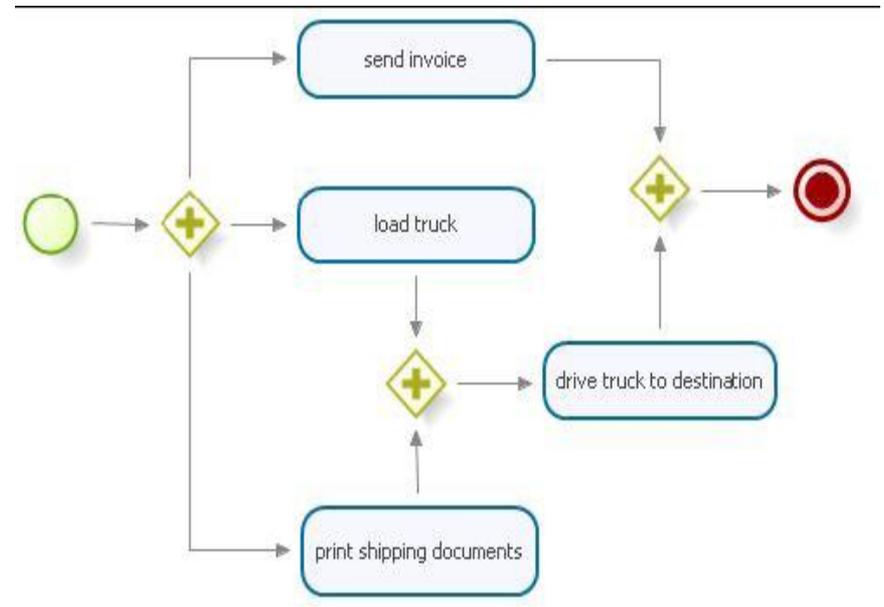
jBPM1

- EJB状态机

项目

jPDL示例

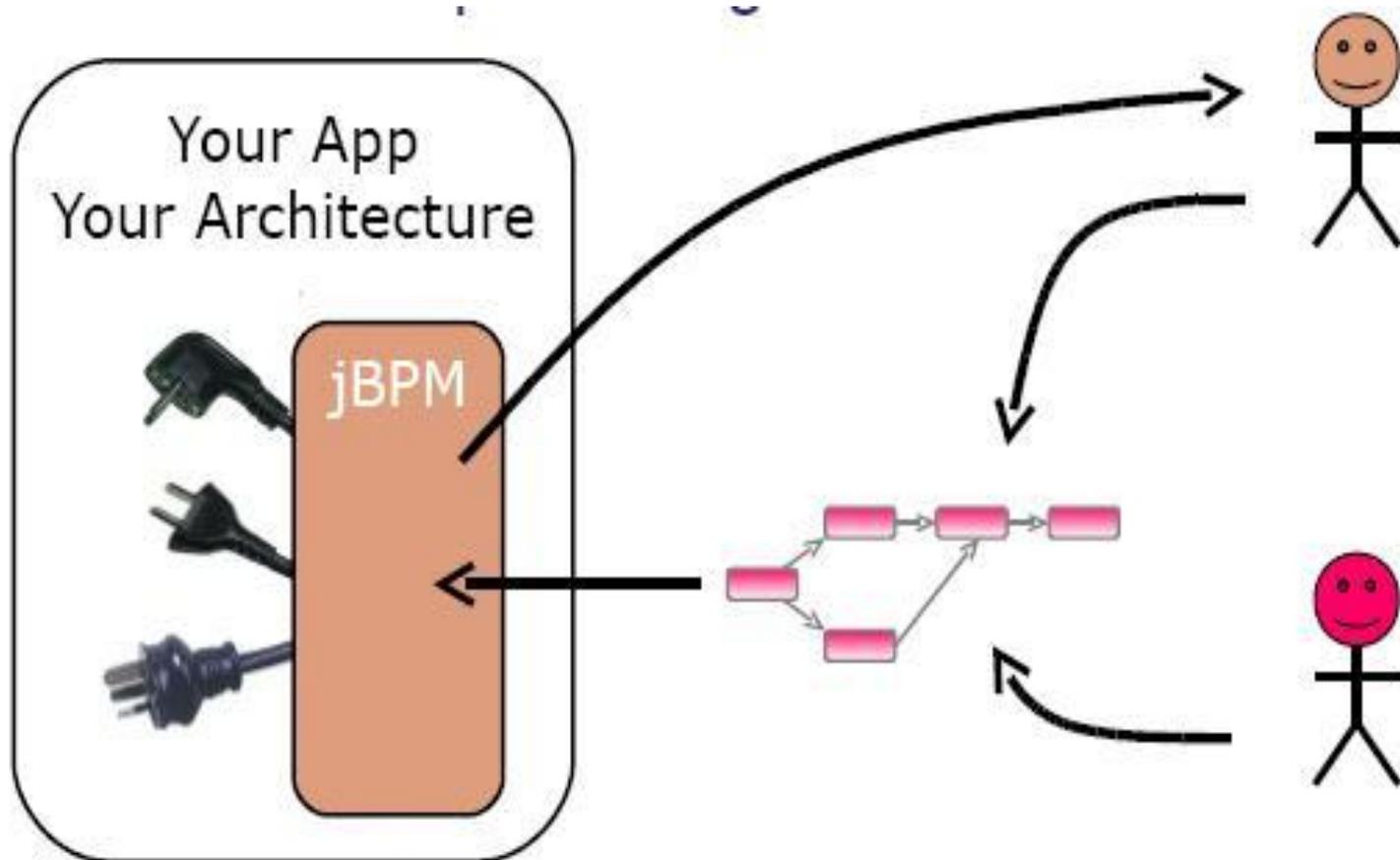
- <process name="Order" >
- <start>
- <transition to="fork"/>
- </start>
- <fork name="fork">
- <transition to="send invoice"/>
- <transition to="load truck"/>
- <transition to="print shipping documents"/>
- </fork>
- <state name="send invoice">
- <transition to="final join"/>
- </state>
- ...



jPDL特性

- 开放性
 - 基本的流程语言可以被扩展
 - 运行期行为的API
- 充实java
 - 抽取状态管理
 - 流程==结构化
 - 代码装饰
- 可嵌入性
 - 库
 - 数据库和应用服务器的无关性
 - 标准的企业级的java

分析-开发人员的桥梁



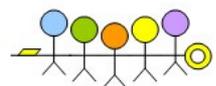
议程

- BPM丛林
- BPMN概览
- BPMN执行语义
- jBPM和jPDL
- 用jBPM执行BPMN

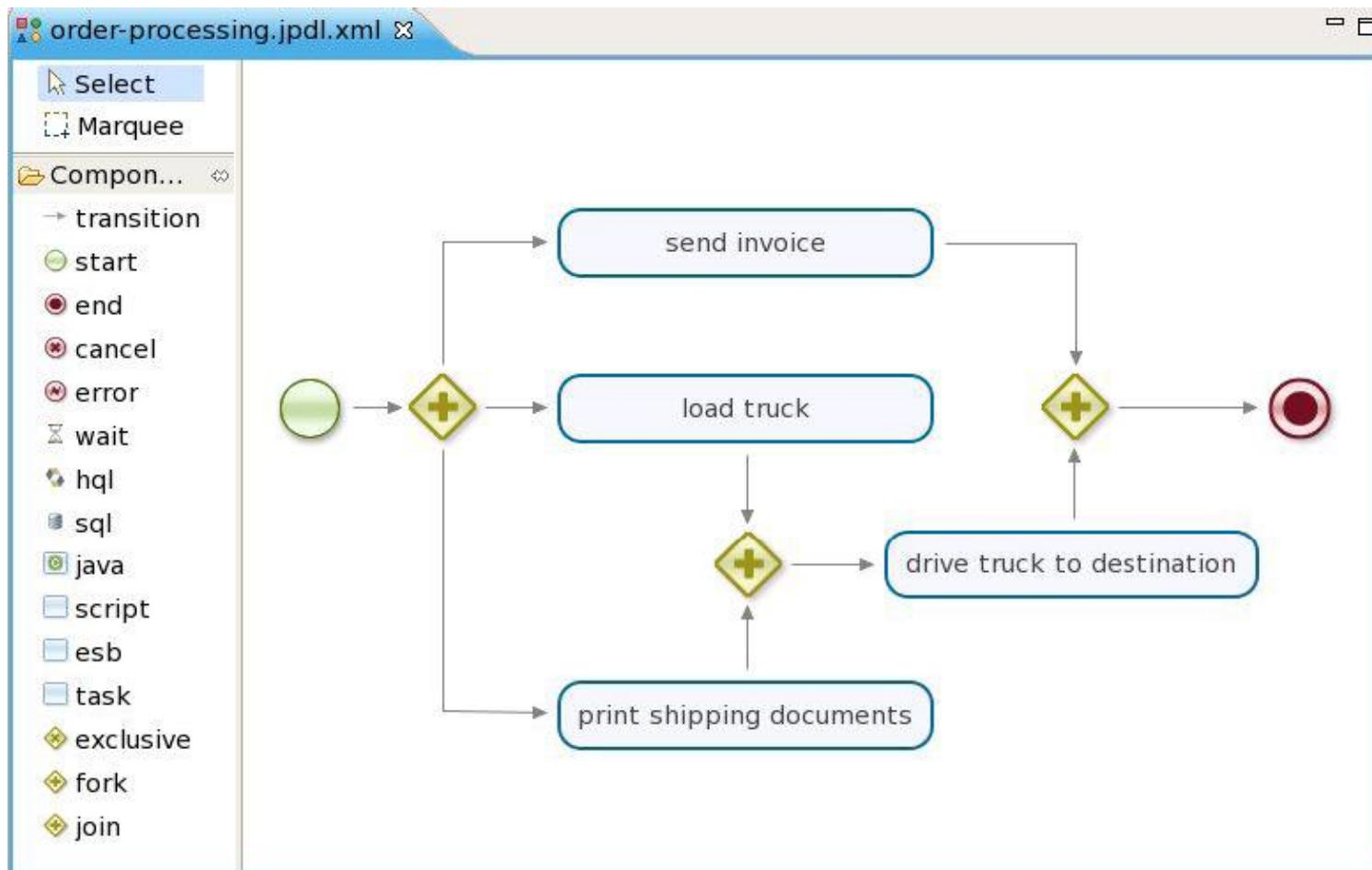
jBPM3

■ BPMN到jPDL的转换

- 基于来自于Eclipse BPMN项目的图形
- 转换为基本的jPDL
- 在jPDL编辑器中加入技术细节

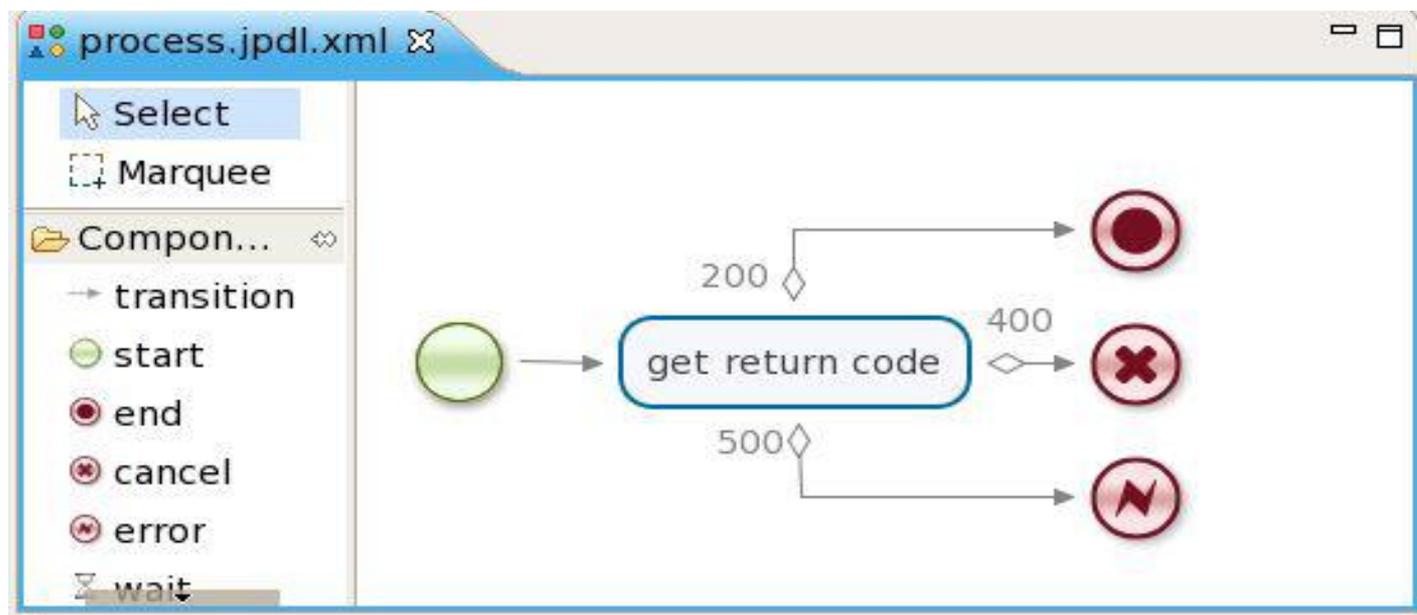


jPDL中的BPMN皮肤展现



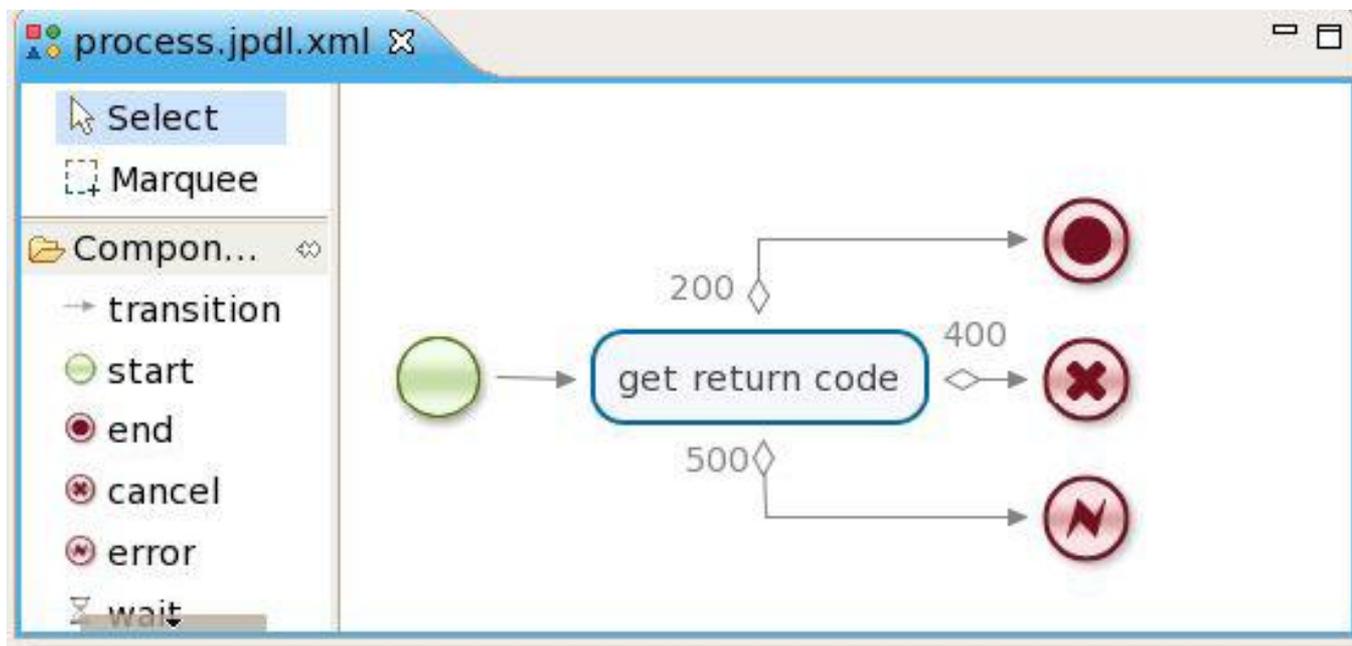
开始和结束流程

- 每个流程仅有一个开始事件
- 可能有多个结束事件
- 当前的结束事件结束整个流程



顺序流

- 一个外向的顺序流总是无条件的
- 多个外向的顺序流
 - 没有隐式的分叉行为
 - 外向的顺序流总是有条件的



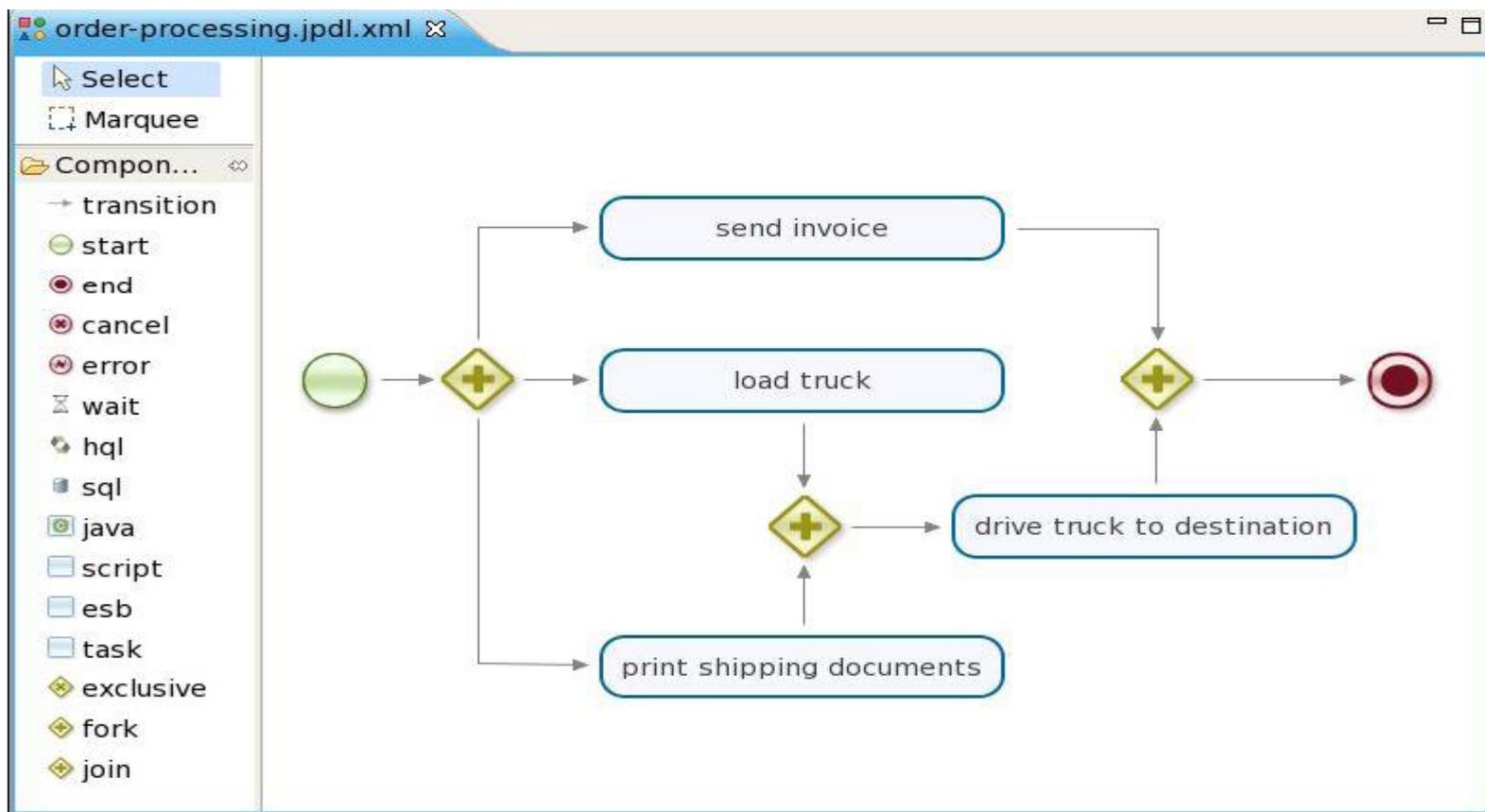
分支与合并

■ 分支

- 一个输入流
- 多个输出流

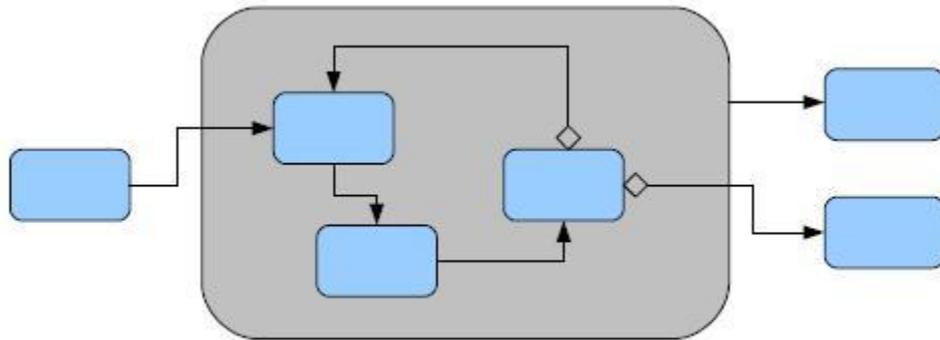
■ 合并

- 多个输入流
- 一个输出流



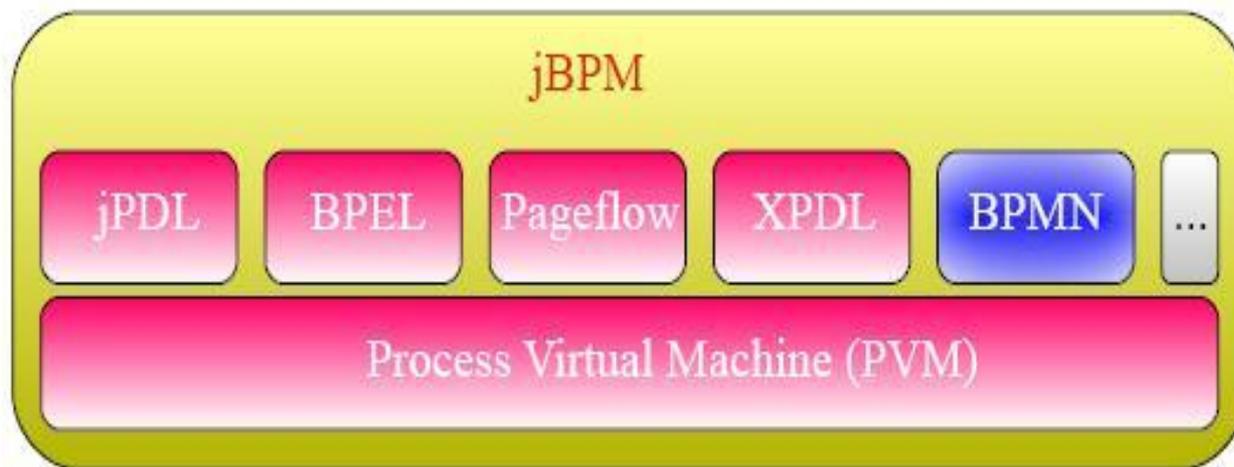
超状态

- 流程中的阶段
 - 特定的任务?
 - 新的人造物?



流程虚拟机之上的BPMN

- 本地执行
- 无映射问题
- jBPM4 GA版本针对实现



结论

- BPM仍旧是一个很复杂的东西
- BPMN是非常有前途的
- 执行语义和一致性仍然是模糊的
- BPMN与jPDL和jBPM建立在相同的概念想法上
- 用jBPM来执行BPMN有两种可能性