



- 1 业务架构驱动的应用架构设计
 - 从模型到软件的转换-业务模型-建模方法发展
- 2 基于活动的过程建模方法
 - 业务任务规划-软件功能设计-活动时序流描述-活动执行控制
- 3 基于数据的过程建模方法
 - 数据分类-数据建模-数据流图-组织建模
- 4 基于状态的过程建模方法
 - DEDS-经典Petri网-高阶Petri网-PNG流程建模-PNG仿真实例
- 5 基于事件的过程建模方法
 - EPC-EPC规则语义-EPC建模规范
- 6 小结

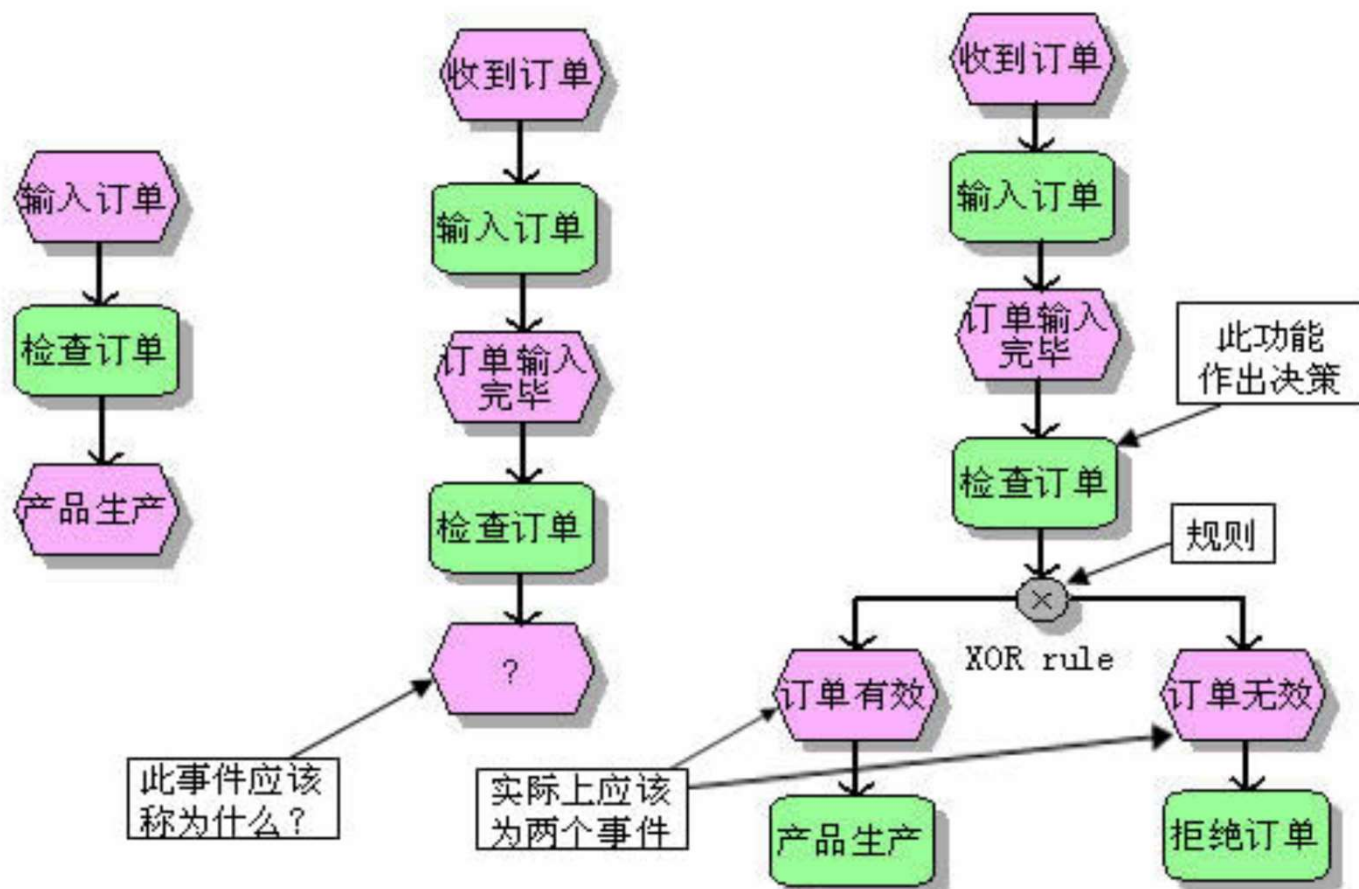
EPC 是一种过程建模方法，全称是Event-driven Process Chain，即事件驱动的过程链。

- EPC出现之前，其他从描述业务组织和过程的方法都太过复杂。
 - 过于面向IT的技术领域；
 - 它们不能提供并行过程的链接；
 - 或者使用了太复杂的符号，也不会提供诸如组织视图这样的观察角度。
- 另一方面，EPC方法从用户、管理人员和咨询顾问的角度出发描绘了业务信息系统，同时加入了其他一些重要特征，如组织结构、功能、数据和信息流，易于理解并且适应实际业务经验表示的符号和语言。
- 因此，EPC被广泛应用ARIS (IDS Prof. Scheer)等软件中。



- EPC强调“过程”是由起始事件和终止事件定义的，包含的基本元素有：
 - 事件(event):事件是指触发某种行为的消息或请求。通常是采用一个“主谓词”形式描述。
 - 能够触发某个流程开始的外部消息（比如，客户订单到达）
 - 流程内部处理状态的改变（比如，产品库存使用完毕）
 - 带有外部影响的最终结果（比如，货物送到了客户）
 - 功能(function):业务流程中的某个操作行为或者完成特定任务的活动。
 - 逻辑连符号: 描述了事件、功能或过程之间的逻辑关系(与，或，异或)。

一个典型的EPC例子



(a) 只有功能的流程

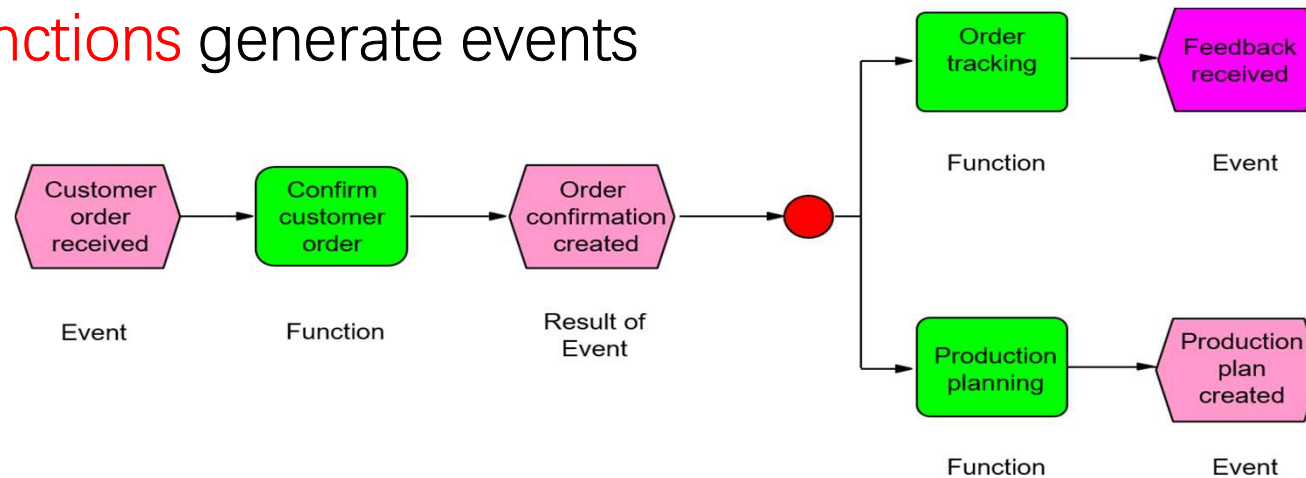
(b) 添加了事件的流程

(c) 正确建模



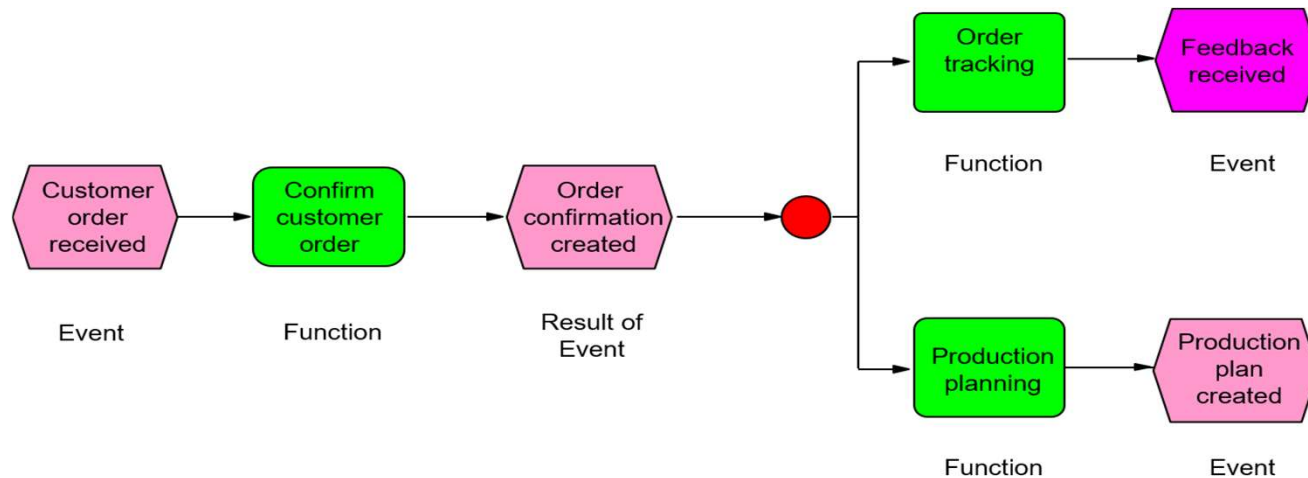
- **功能**是为达到一个或多个企业目标而作用在（信息）对象上的一个任务，操作或活动。
- 一个功能可以由时间和成本来定义特性。

Functions generate events



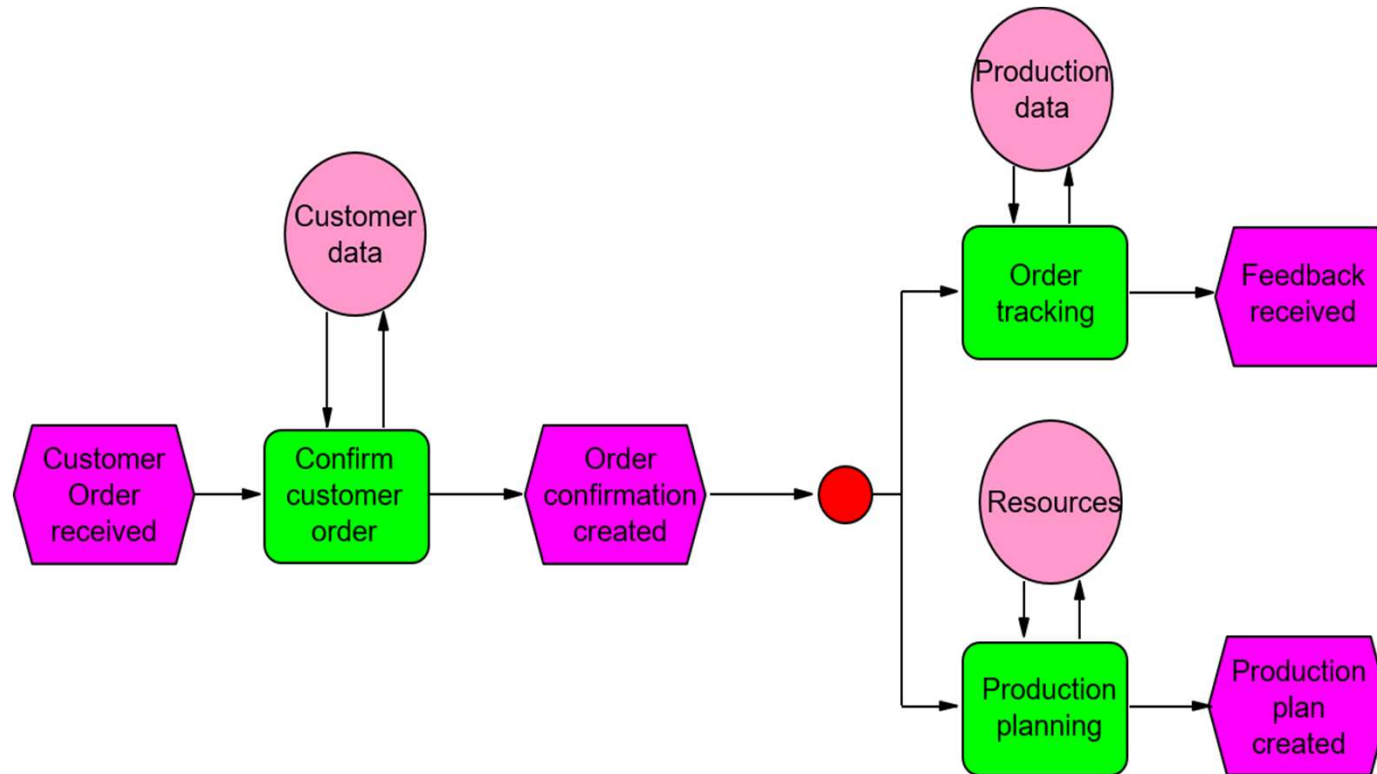


Events initiate functions

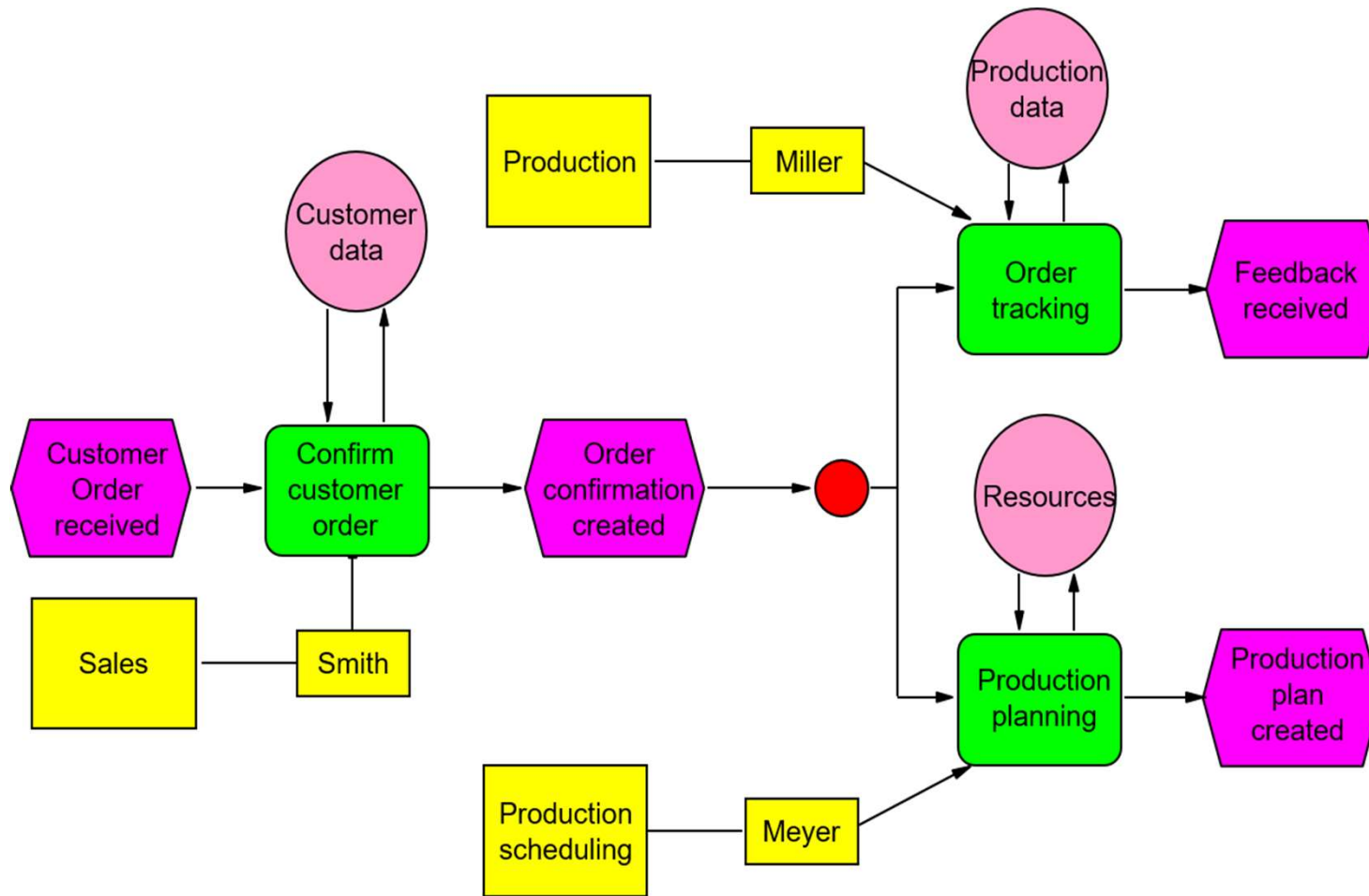


- **事件**描述与业务相关的信息对象的状态。
- 某种行为的消息或请求，也可理解为某种状态的改变
- 这个状态可包括功能的pre-conditions及 post-conditions，可能控制或影响业务过程的运行。

- Data are processed in functions



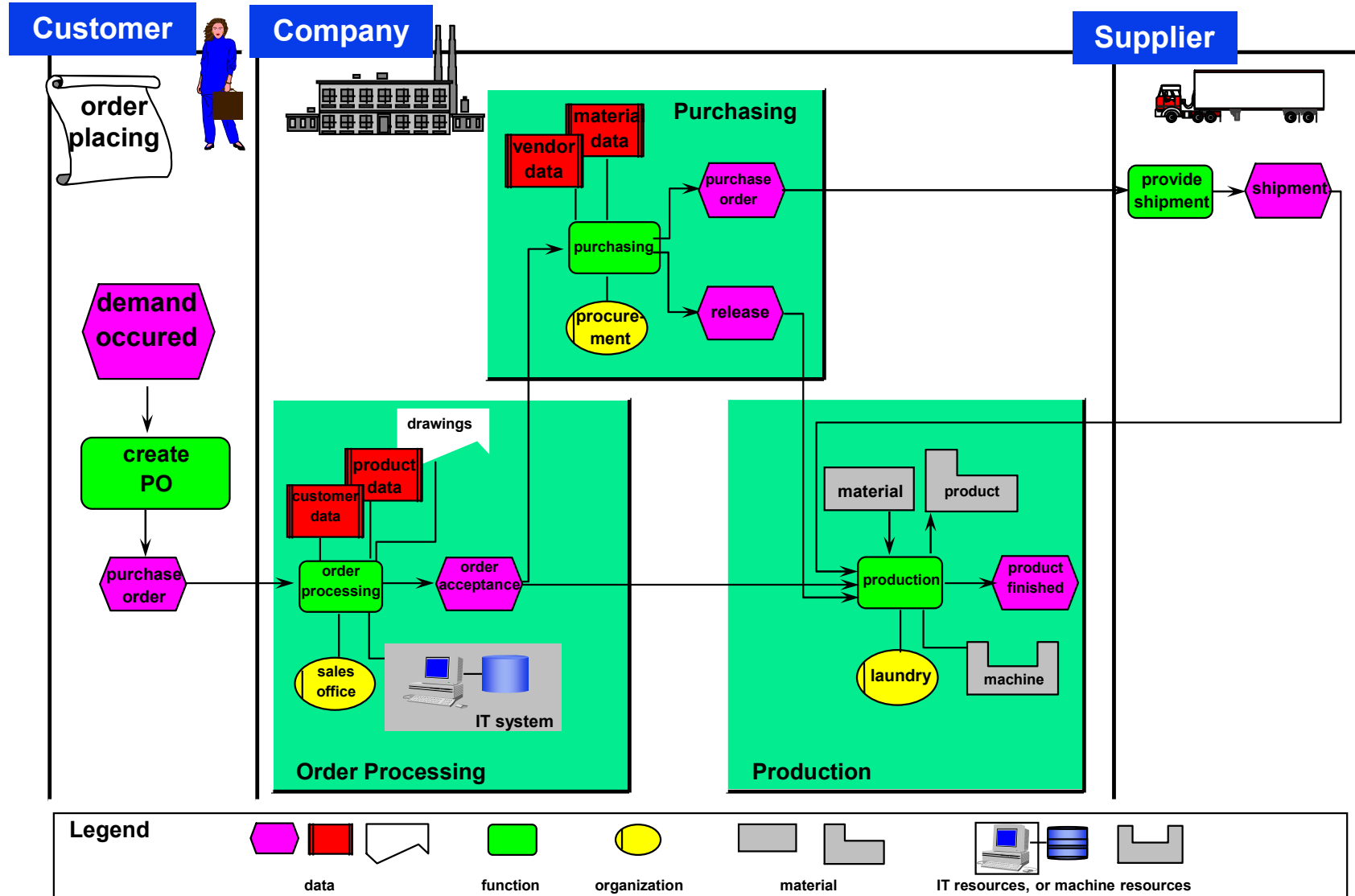
- Employees/Persons are in charge of functions



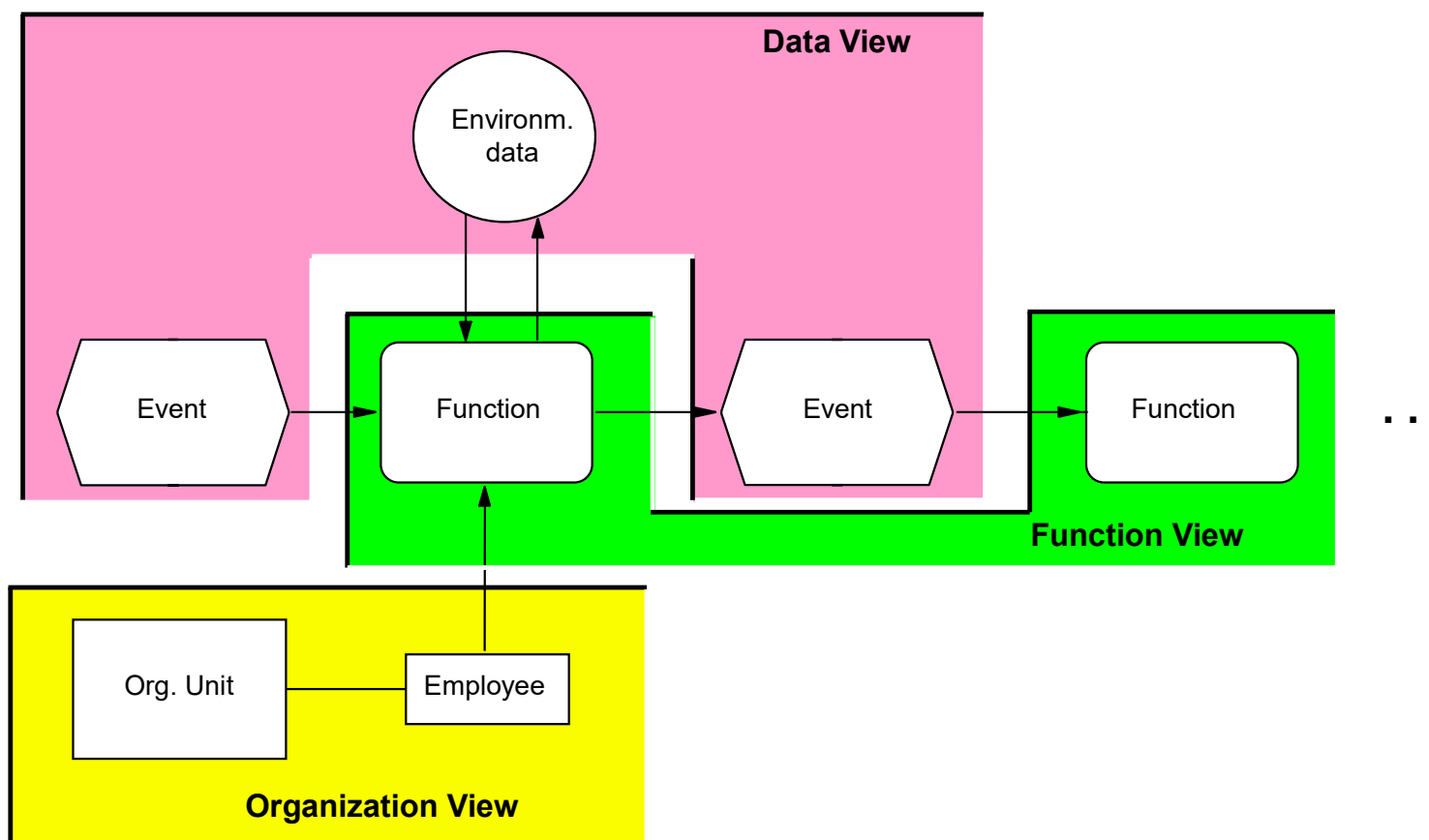


- EPC元素概念在计算机建模中的表述：
- 功能（圆角四边形）：为达到一个或多个企业目标而作用在（信息）对象上的一个**任务，操作或活动**
 - 事件：描述与业务相关的信息对象状态及状态变化的**消息**
 - 组织单元：组织可以是一个公司、部门或员工、岗位
 - 信息：信息可作为一个过程的**输入或输出**数据

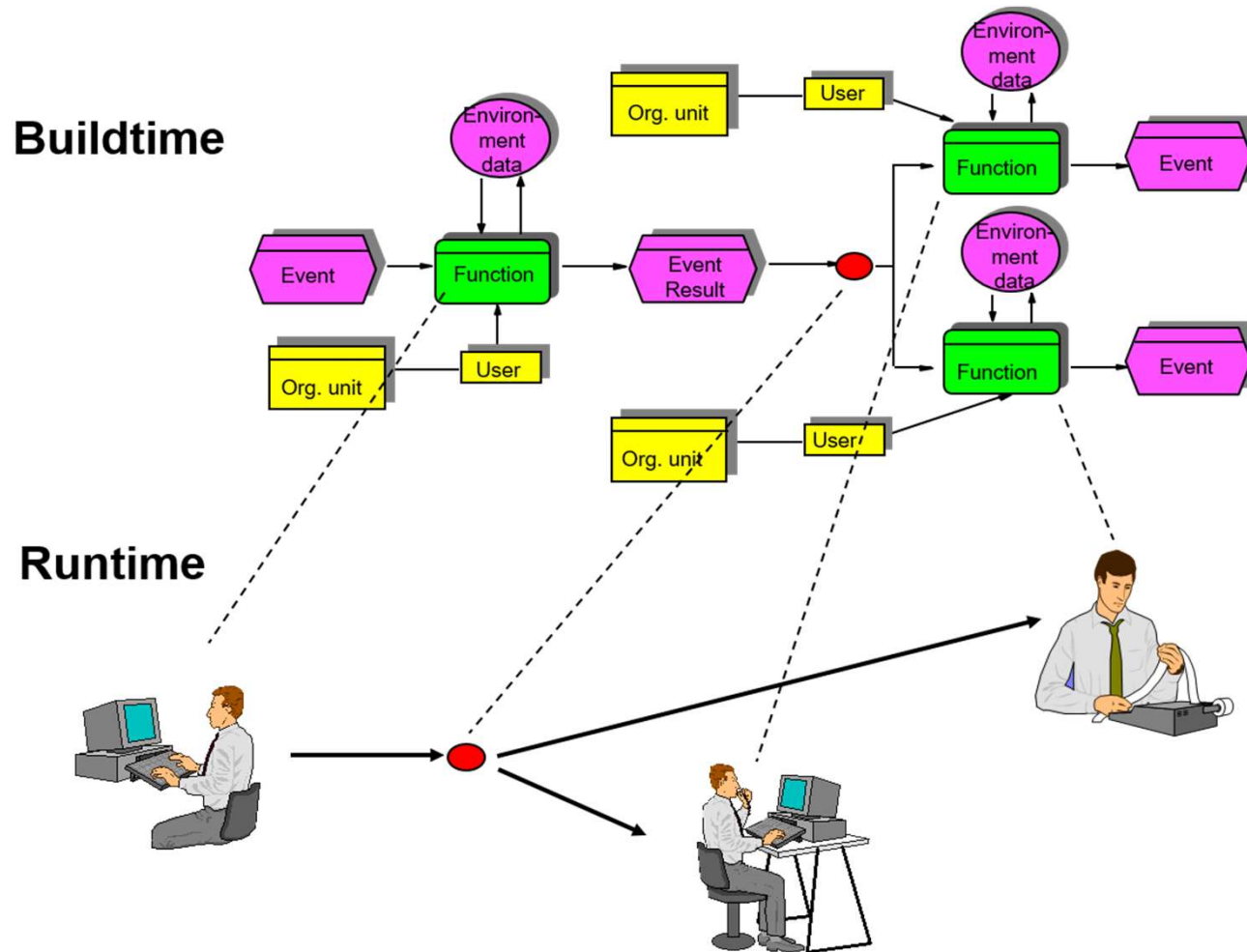
EPC特点1：适合业务人员建模



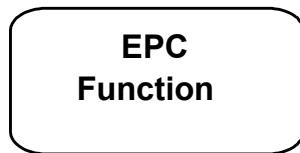
- 一个流程可以被分为多个视图，可以分别表述、分别建模，使得建模的复杂性降低。



从EPC模型到运行程序的映射



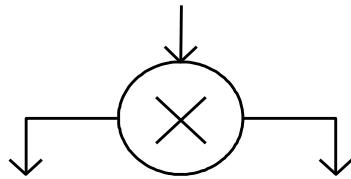
5.2 事件过程链的规则语义



Functions:
activities of the process

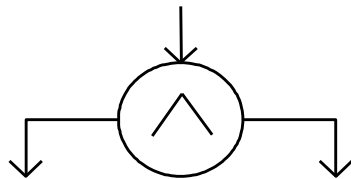


Events: pre- and post-
conditions of functions



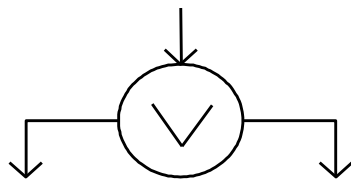
**XOR
Split**

XOR split:
defines a choice to activate one of multiple subsequent
branches.



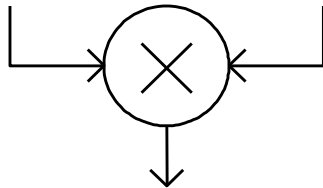
**AND
Split**

AND split:
activates all subsequent branches in concurrency



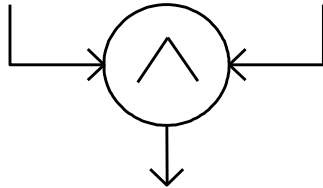
**OR
Split**

OR split:
triggers one, two or up to all of multiple subsequent
branches.



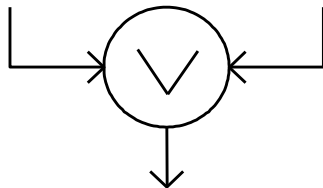
**XOR
Join**

- XOR join:
continue when one of alternative
branches has completed.



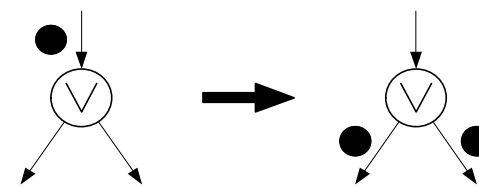
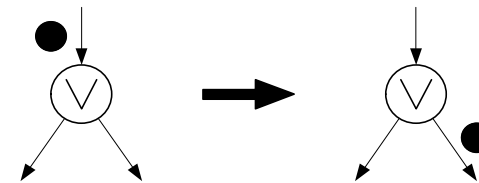
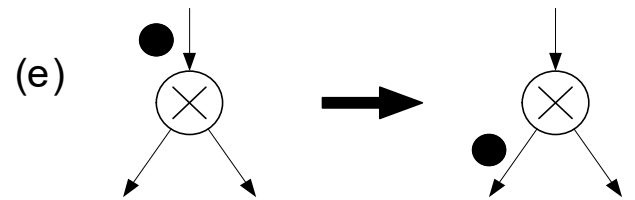
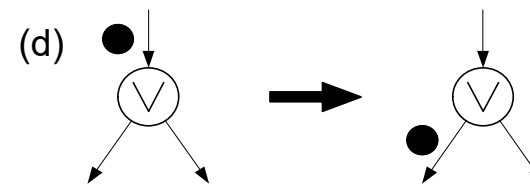
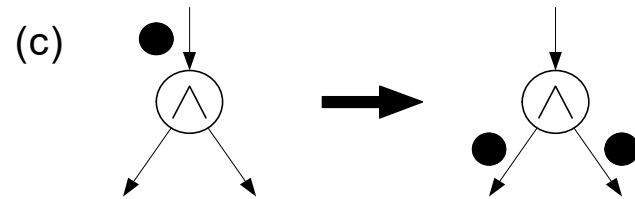
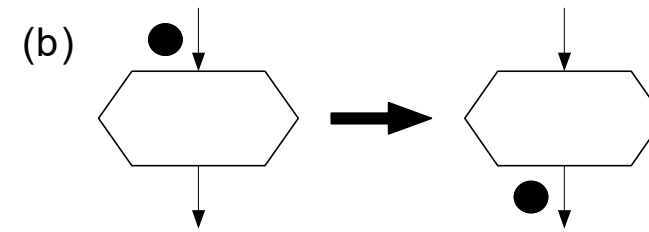
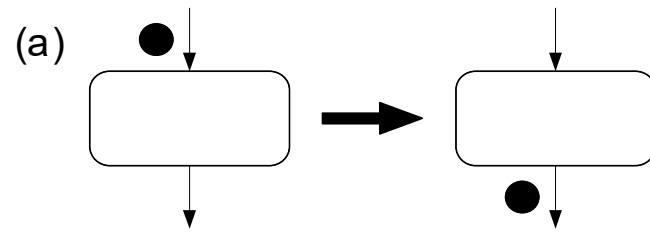
**AND
Join**

- AND join:
waits for all incoming branches to
complete.



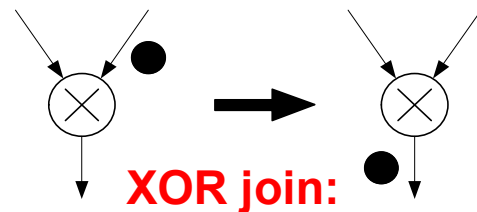
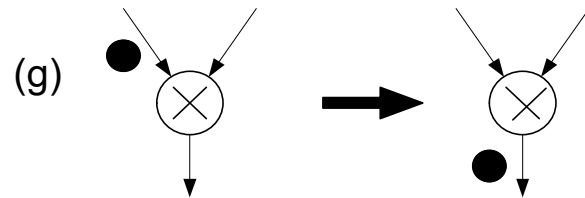
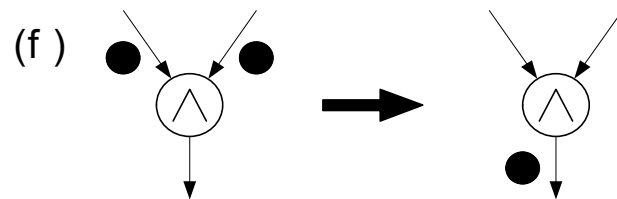
**OR
Join**

- OR join :
waits for all active branches to
complete.



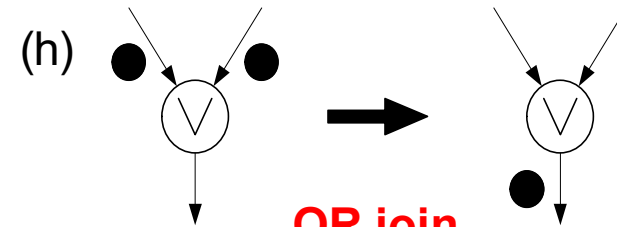
AND join:

waits for all incoming branches to complete.



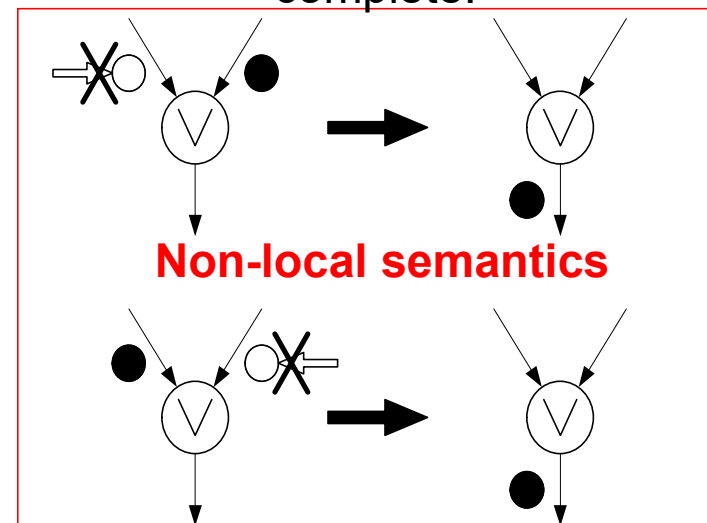
XOR join:

continue when one of alternative branches has completed.

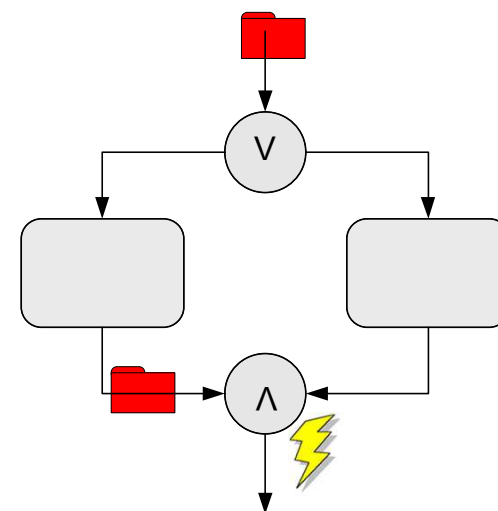
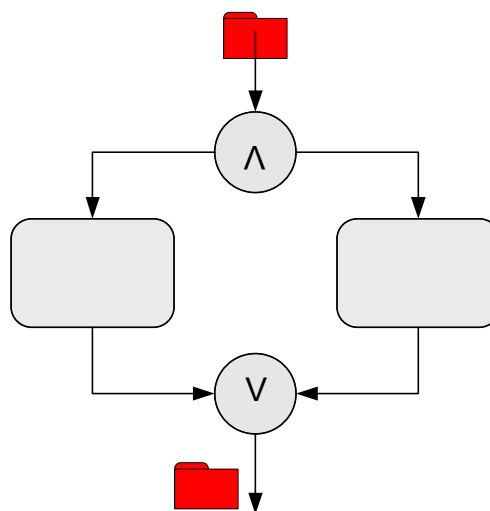
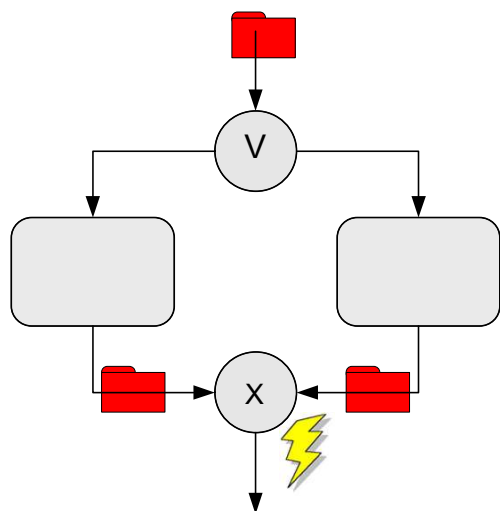
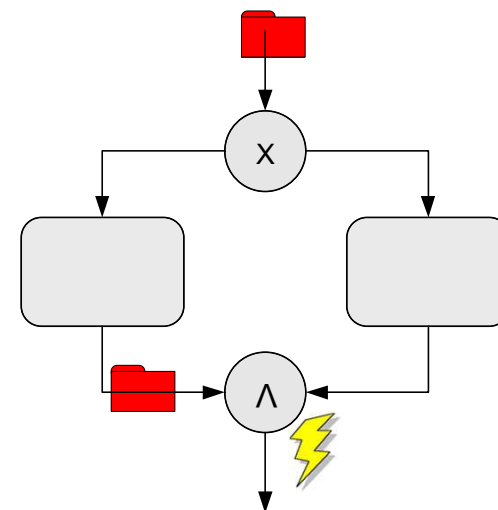
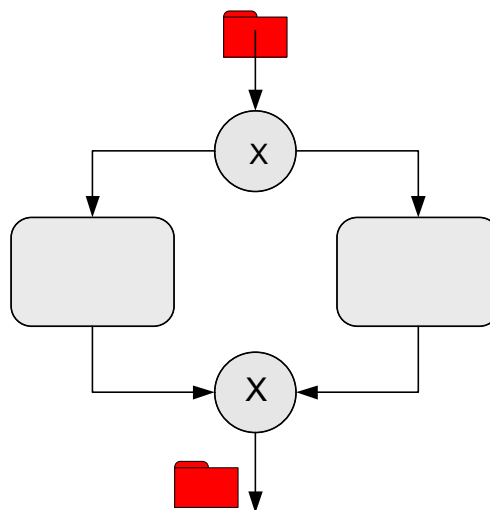
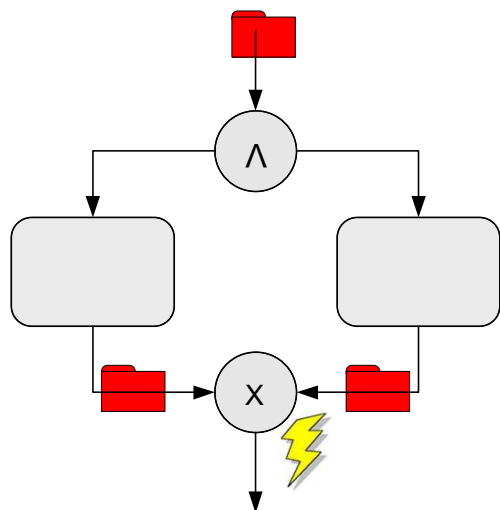


OR join

waits for all *active branches* to complete.



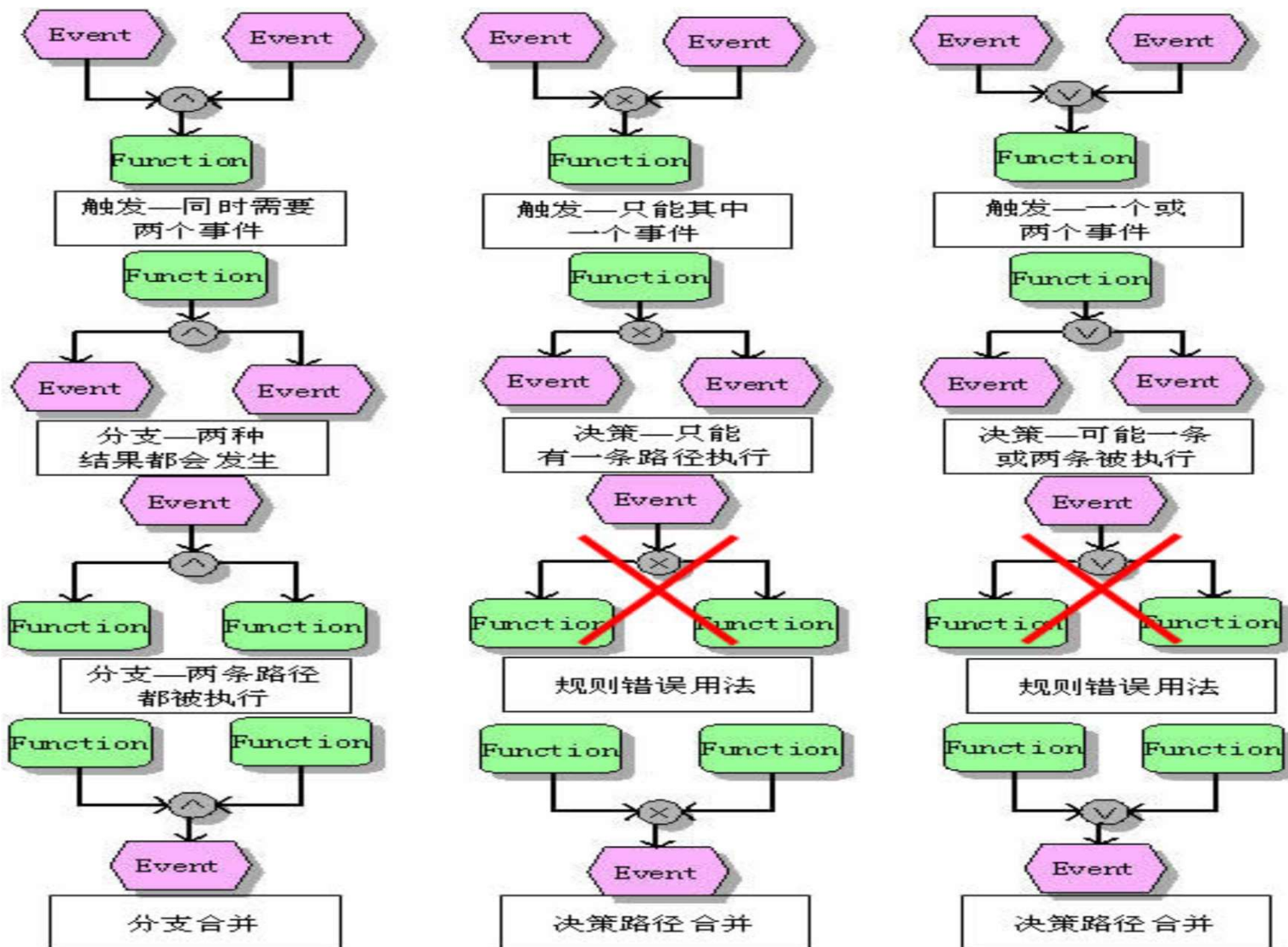
不正确的匹配



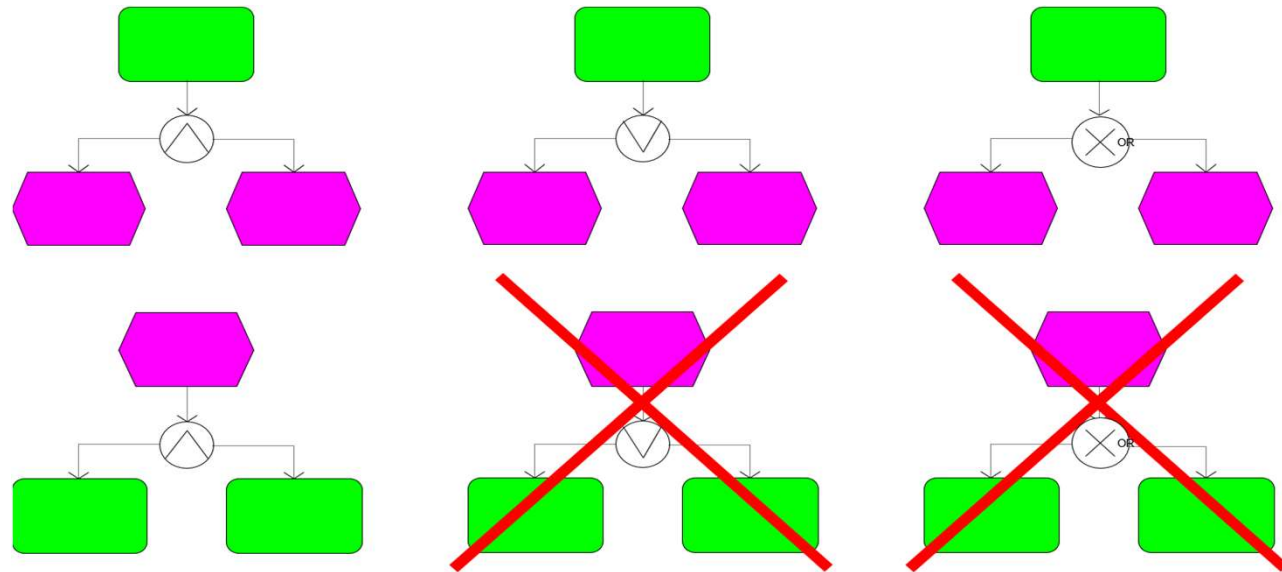
► EPC的规则体系：

操 作 符	在功能之前（单输入多输出）	在功能之后（多输入单输出）
OR	或决策, 在一个决策之后有一个或多个可能的结果路径	或事件, 功能有一个或多个触发事件
XOR	异或决策, 在某一时刻有且只有一个可能的路径	异或事件, 在某一时刻有且只有一个可能的触发事件
AND	与分支, 流程被分成两个或多个并行的分支	与事件, 所有的事件要同时满足才能触发功能

基于规则组合的复杂流程建模



- 每一个模型必须至少包含一个开始事件和一个结束事件。
- 功能与事件交替着出现。
- 事件和功能永远只有一个输入和一个输出连接。
- 流程路径使用规则进行分离与合并。
- 事件是静态的，无法决定决策，决策必须是由功能作出，不要在事件后使用带有决策的规则连接。



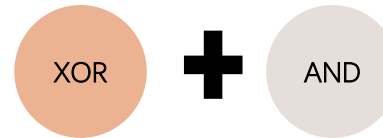


面向流程的执行，EPC有七条建模规范：

- G1: Use as few elements in the model as possible
- G2: Minimize the routing paths per element.
- G3: Use one start and one end event.
- G4: Model as structured as possible.
- G5: Avoid OR routing elements.
- G6: Use verb-object activity labels.
- G7: Decompose the model if it has more than 50 elements.

- G1: Use as few elements in the model as possible.
 - Larger models tend to be more difficult to understand and have a higher error probability than small models.
- G2: Minimize the routing paths per element.
 - higher the degree of an element, harder to understand, higher the modeling errors.
- G3: Use one start and one end event.
 - Low error probability.
 - Workflow engines.

- G4: Model as structured as possible.
 - as formulas with balanced brackets
- G5: Avoid OR routing elements.



- G6: Use verb-object activity labels.
 - Verb-object style(“Inform complainant”) VS. Action-noun labels (e.g. “Complaint analysis”).
- G7: Decompose the model if it has more than 50 elements.
 - For models with more than 50 elements the error probability tends to be higher than 50%.



- 1 业务架构驱动的应用架构设计
 - 从模型到软件的转换-业务模型-建模方法发展
- 2 基于活动的过程建模方法
 - 业务任务规划-软件功能设计-活动时序流描述-活动执行控制
- 3 基于数据的过程建模方法
 - 数据分类-数据建模-数据流图-组织建模
- 4 基于状态的过程建模方法
 - DEDS-经典Petri网-高阶Petri网-PNG流程建模-PNG仿真实例
- 5 基于事件的过程建模方法
 - EPC-EPC规则语义-EPC建模规范
- 6 小结



- 应用架构设计的核心是确定软件的业务功能，进而描述动态交互行为。
- 以流程模型为核心，可以建立基于任务、数据、事件、状态等四种信息建模方法，也对应着执行流、数据流、事件流、以及状态转移等动态描述。
 - 基于任务的流程建模方面，其核心是时序任务的执行过程；
 - 基于数据的过程建模方法，其核心是信息处理流程；
 - 基于事件的流程建模方法，其核心是基于消息的信息交换过程；
 - 基于状态的过程建模方法较为抽象，适用于复杂并发的应用场合。
- 从业务建模驱动技术架构的出发，阐述了应用架构设计的主要方法。