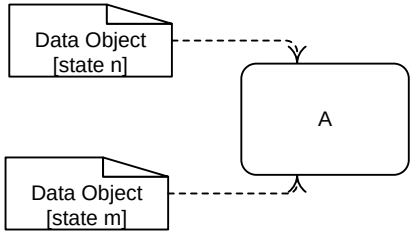
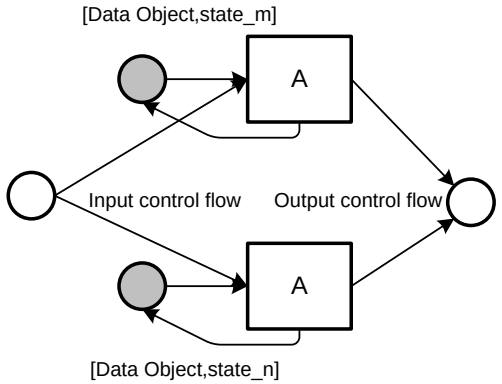
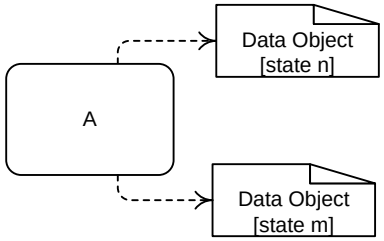
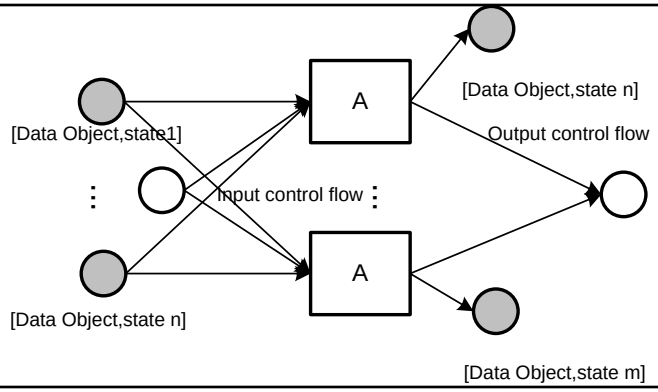
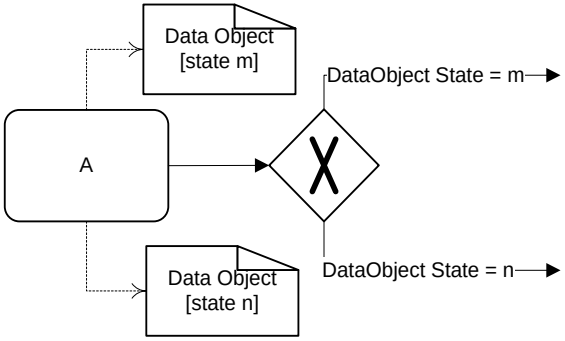
Formalization of Data Access Semantics

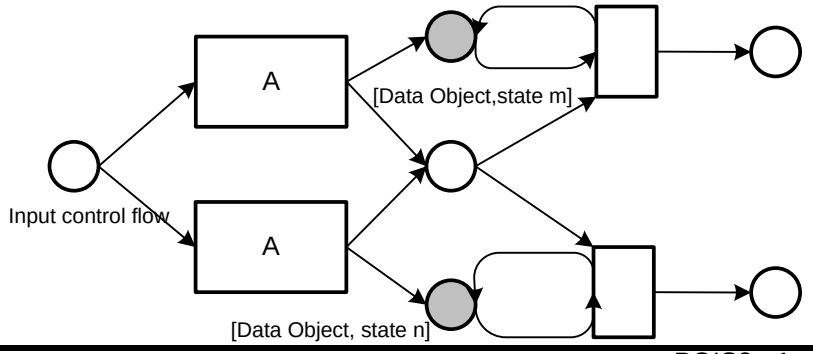




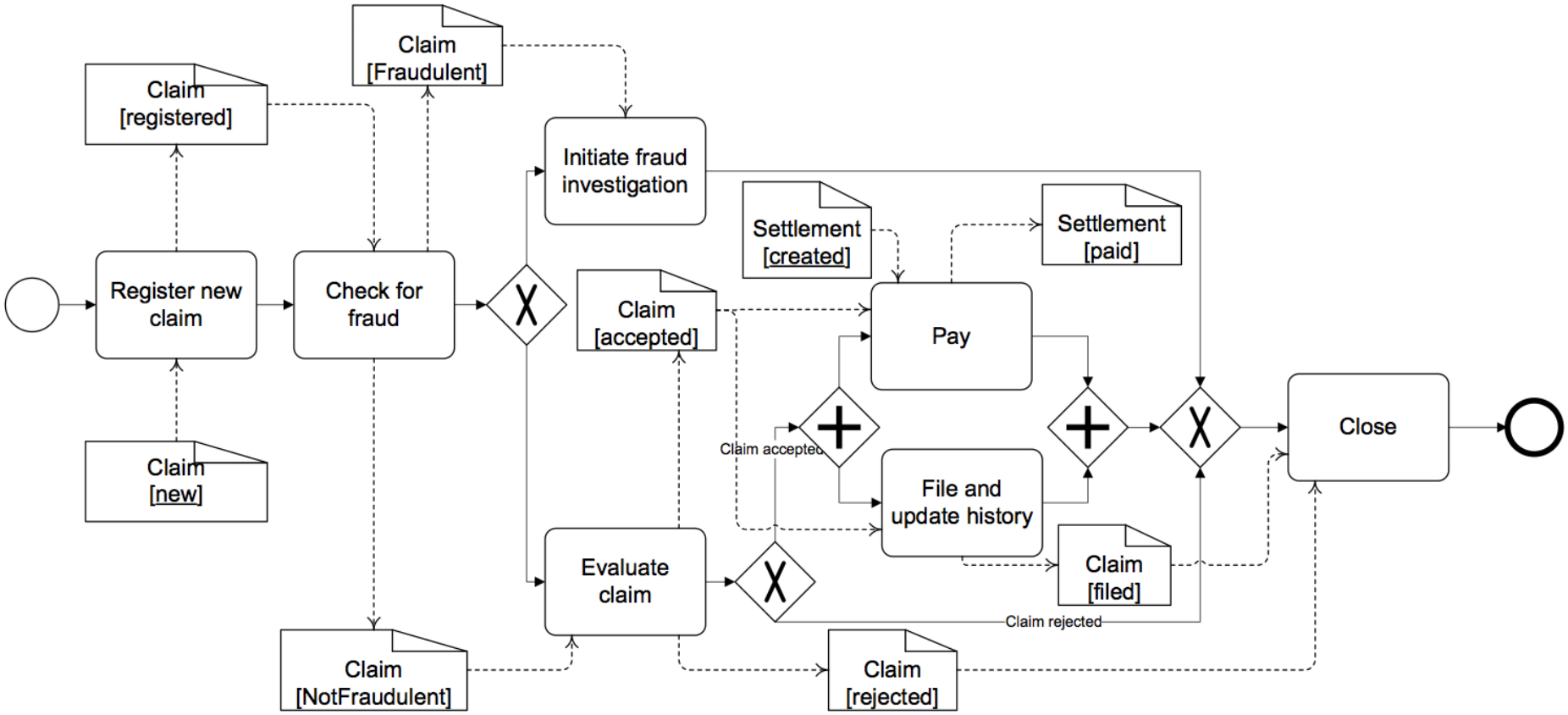




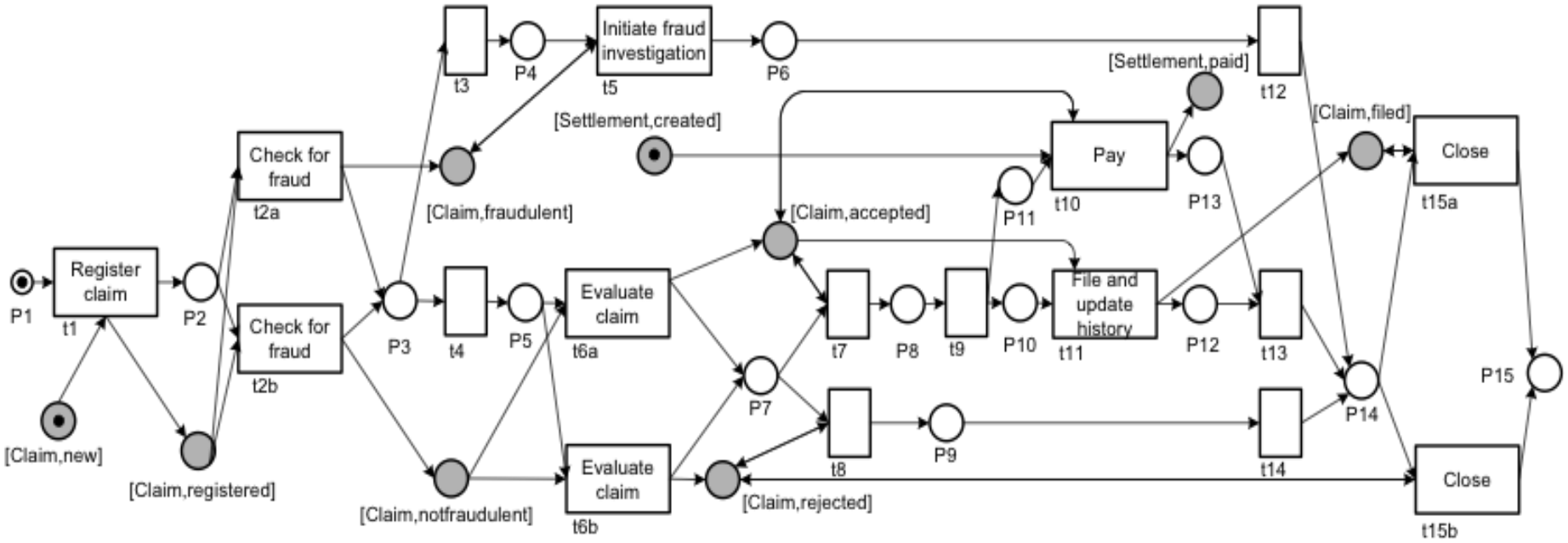




Let's map this to Petri nets



Resulting Petri net

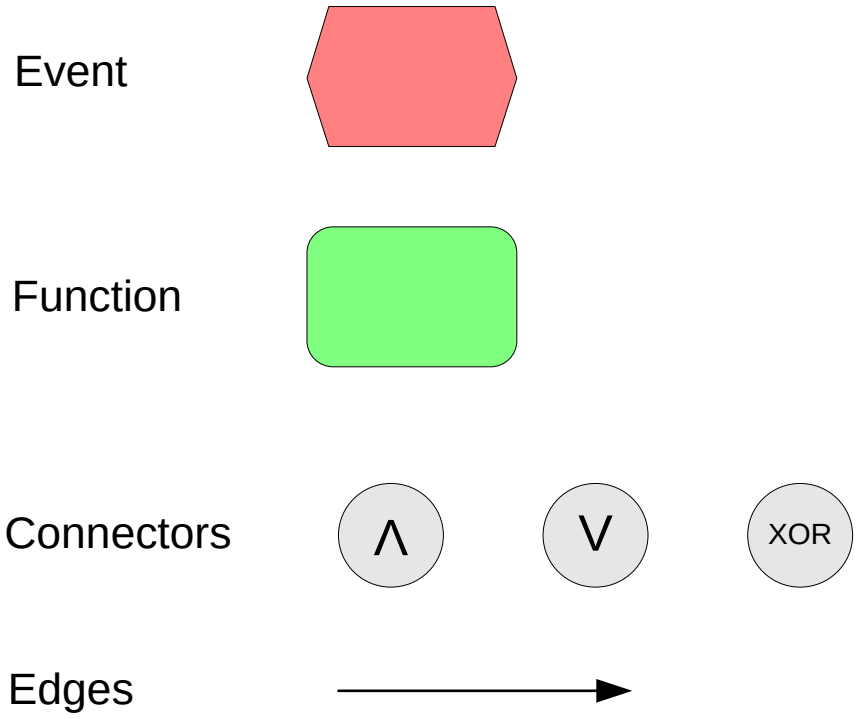


Event-driven Process Chains

- Semi-formal process modeling technique
  - Developed by A.W. Scheer, in 1992
  - ARIS software product developed by IDS Scheer AG
  - Easy to use by domain experts in the foreground, less formal investigations or software implementation of business processes
- Event-driven Process Chains (EPC)
  - Process modeling: Functions, Events, Connectors
- Extended EPC (eEPC)
  - Modeling organisational units
  - Input and output behavior of functions included

Elements

- Events
  - Representing occurrence of an economically relevant state
  - Events trigger functions or are triggered by termination of functions, pre/post conditions
  - Events do not make decisions (XOR, OR)
- Functions
  - Represent units of work
  - Transform input after outputs
  - Can make decisions
- Connectors and edges



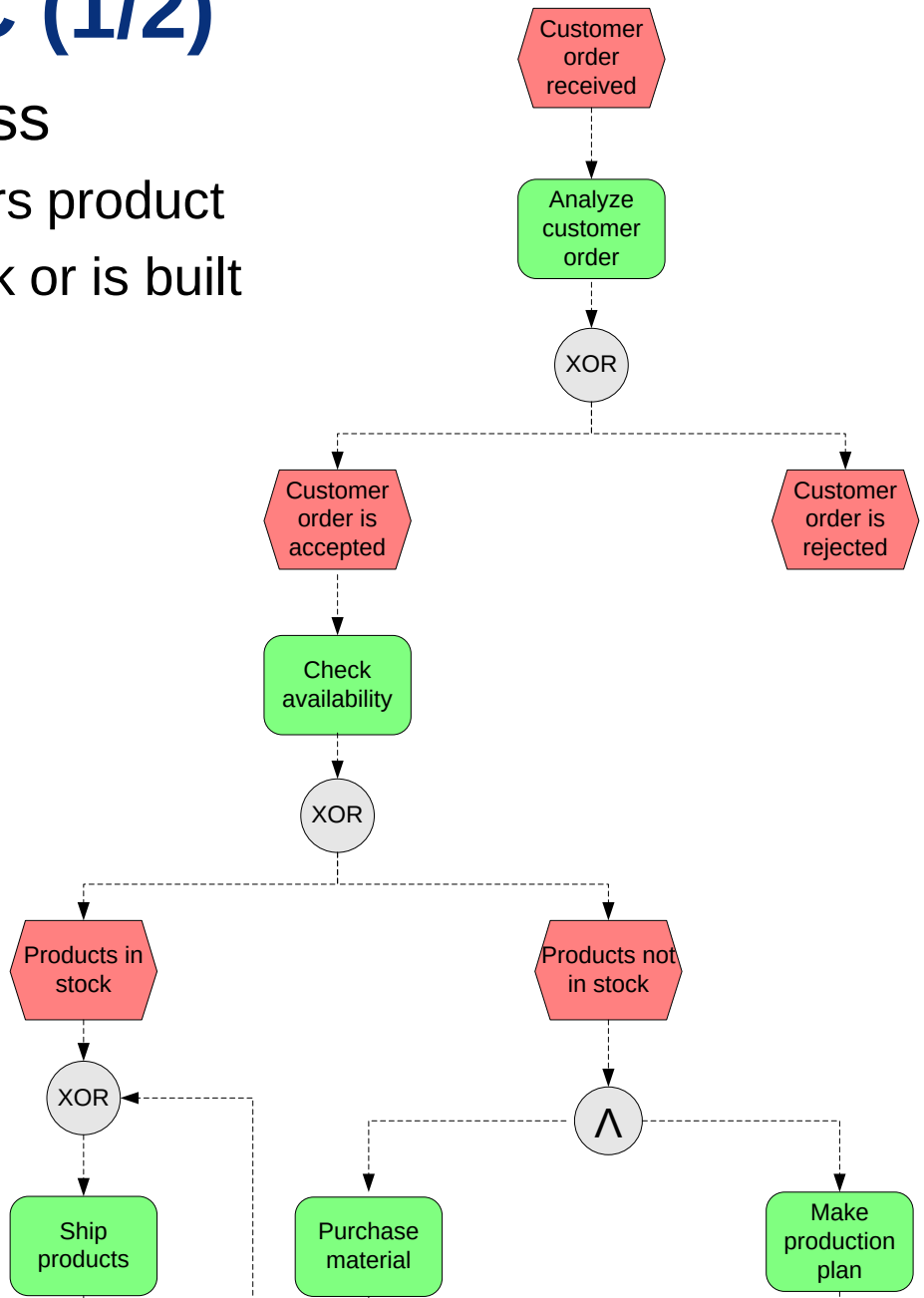
from M. Weske: Business Process Management, © Springer-Verlag Berlin Heidelberg 2007

Definition 4.7 A tuple $A = (E, F, V, m, C)$ is an *event driven process chain*, if

- E is a non empty set of events
- F is a non empty set of functions
- V is a set of connectors
- $m : V \mapsto \{and, or, xor\}$ is a mapping that assigns to each connector a connector type, representing *and*, *or* and *exclusive or* semantics.
- Let $K = E \cup F \cup V$. $C \subseteq K \times K$ is a set of edges connecting events, functions and connectors, such that the following conditions hold:
  - (K, C) is a connected graph
  - Each function has exactly one incoming edge and exactly one outgoing edge.
  - Each start event has exactly one outgoing edge and no incoming edge. Each end event has exactly one incoming and no outgoing edge. There is at least one start event and one end event. All other events have exactly one incoming edge and one outgoing edge.
  - Each event can only be followed (possibly via connectors) by functions, and each function can only be followed (possibly via connectors) by events.
  - There is no cycle in an event driven process chain that consists of connectors only.
  - No event is followed by a decision node, i.e., an *or* split node or *exclusive or* split node.

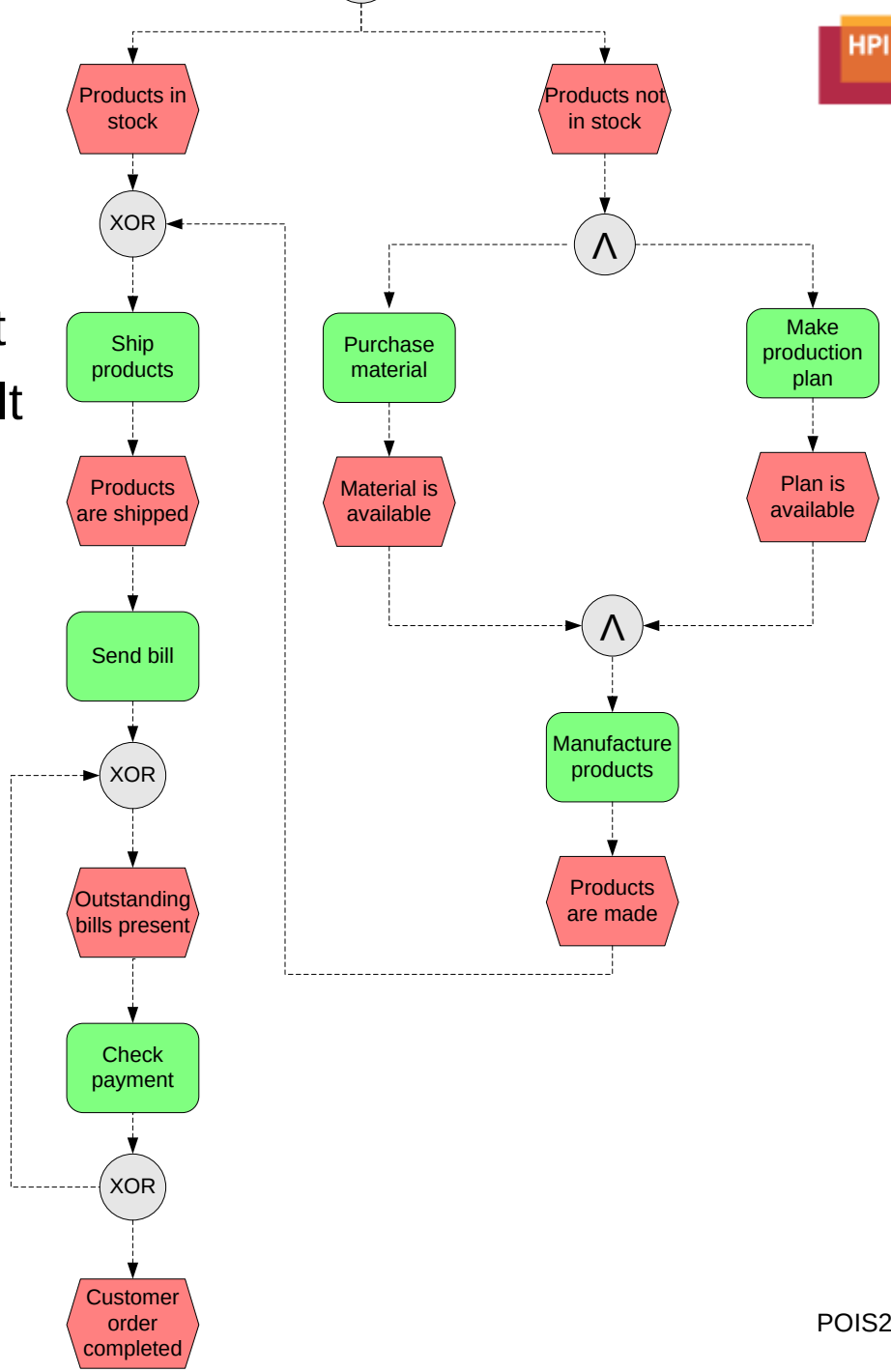
Example-EPC (1/2)

- Ordering Process
  - Customer orders product
  - Product in stock or is built
  - Payment



Example-EPC (2/2)

- Ordering Process
  - Customer orders product
  - Product in stock or is built
  - Payment



Correctness Analysis of EPC

- Problem
  - EPC is an informal Notation
  - Correctness analysis is defined only for formal notations
- Approach
  - Transformation from EPC to workflow nets
  - Analysis of workflow nets

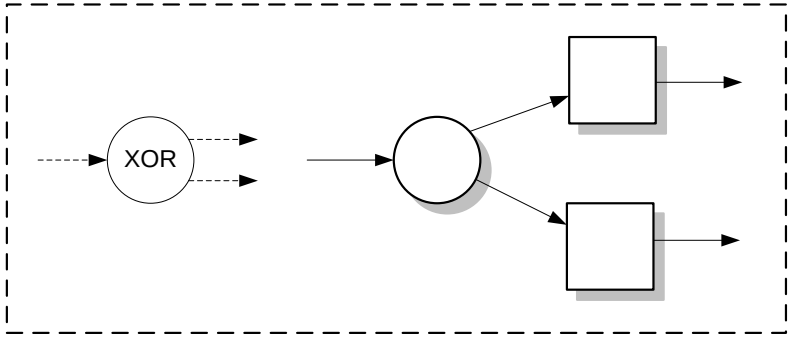
Mapping EPC to a workflow network

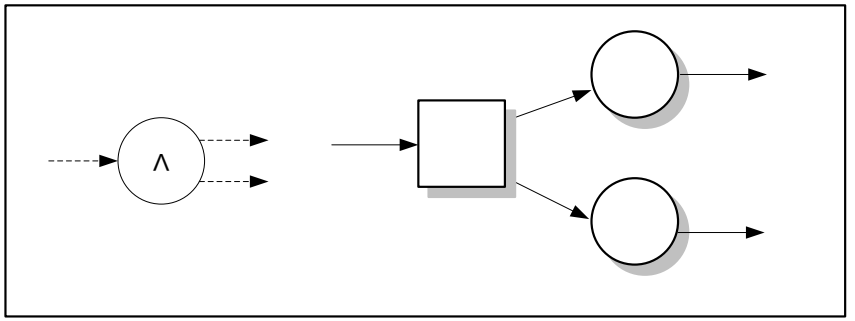
- Step 1a: generation of workflow modules
  - Functions of an EPC are mapped to transitions of a workflow net.
  - Events of an EPC are mapped to places of a workflow net.

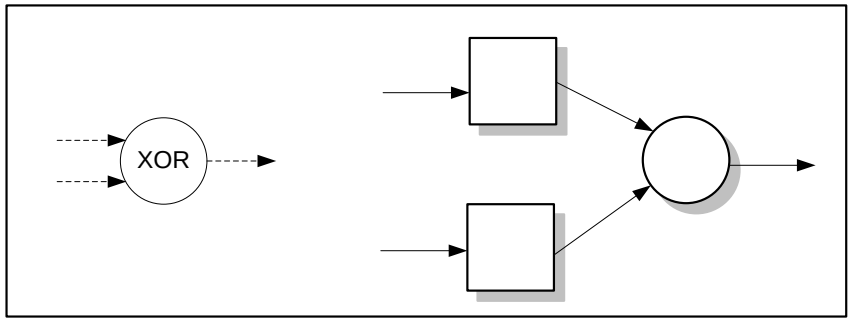
from M. Weske: Business Process Management, © Springer-Verlag Berlin Heidelberg 2007

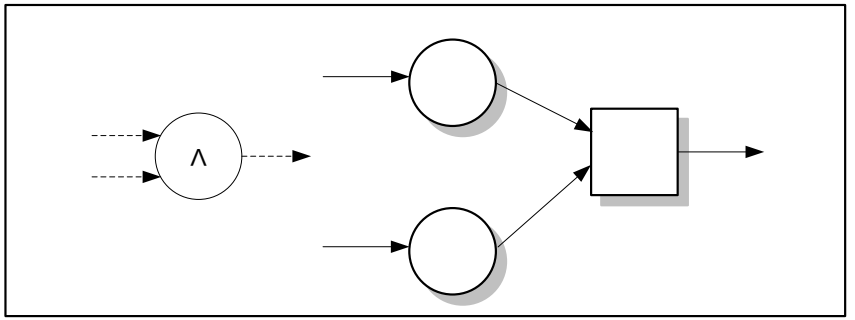
Mapping EPC to a workflow network

- Step 1b: generation of workflow modules
  - Connectors are mapped to sub-workflow nets









Mapping EPC to a workflow network

- Step 2: Combining workflow modules
  - When combined with different input and output elements, they are connected
  - Otherwise they are merged
  - Example
    - Function immediately before a Split-node or
    - Event immediately before a Join-node

Mapping EPC to a workflow network

- Step 3: Reach the structure of workflow net
  - Step 3a: may add an initial place i
  - Step 3b: may add a final place o

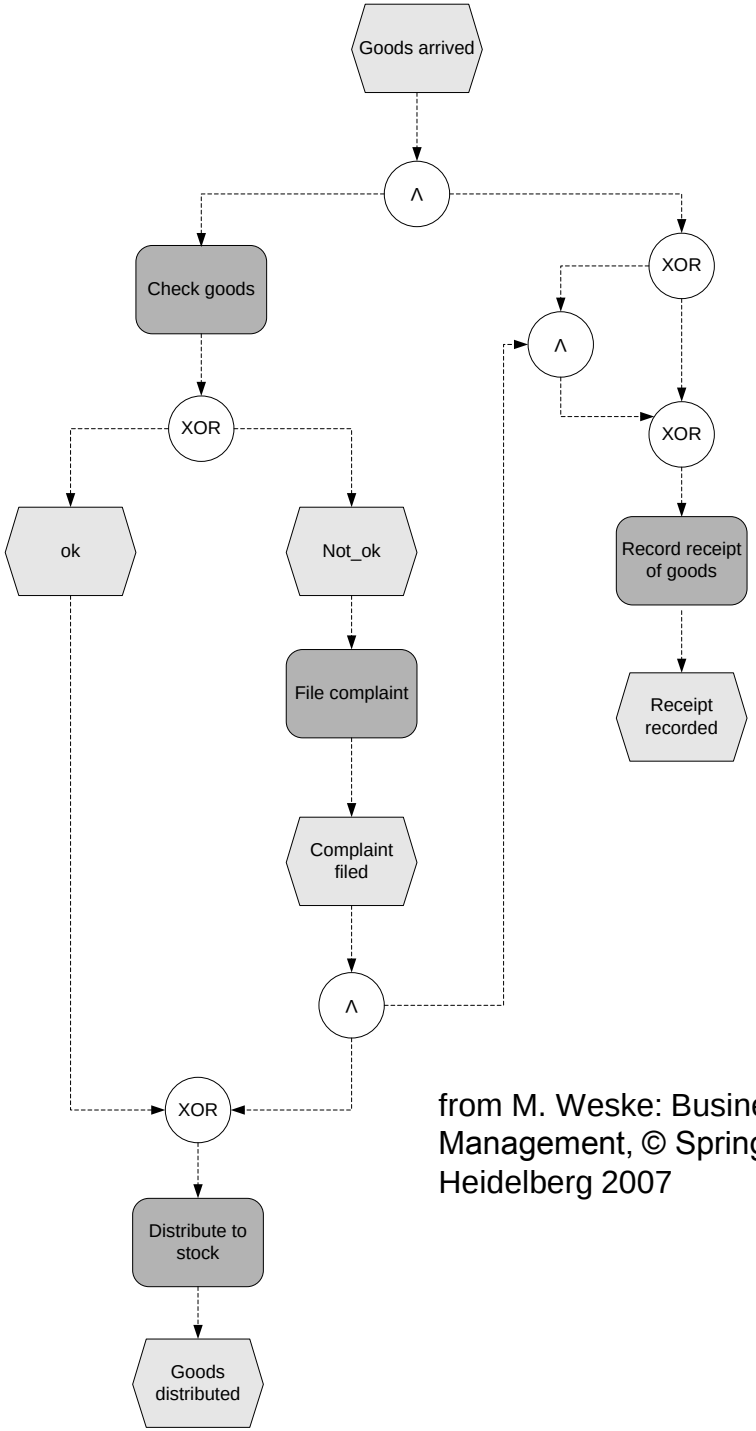
- Step 3a: If multiple places exist without incoming edges: (Assumption: alternative start events)
  - Create a new initial place i
  - For each place p with no incoming edges create transition t and edges from t to p and from i to t

Mapping EPC to a workflow net

- Step 3b/1: If multiple places exist without outgoing edges: (Assumption: common end-events)
  - Create a new final place o
  - Create a new transition t
  - For each place p without an outgoing edge create an edge from p to t
  - Create an edge from t to o
- Step 3b/2 (Assumption: alternative end-events)
  - Create a new final place o
  - For each place p without an outgoing edge create a transition t
  - Create edges from p to t and from t to o

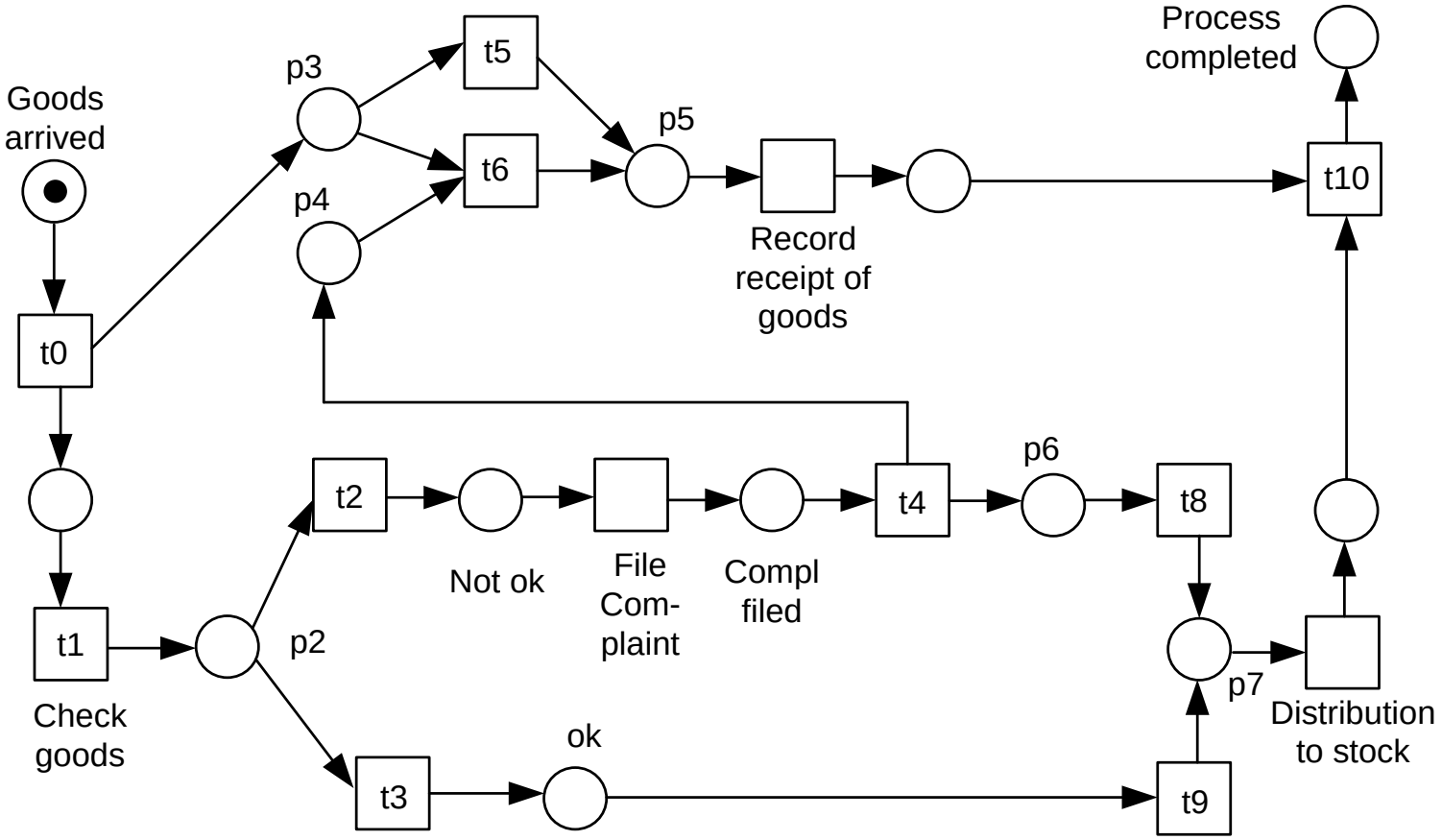
Example

- Step 1
  - Functions, events and connectors
- Step 2
  - Combining the modules
- Step 3
  - New end event

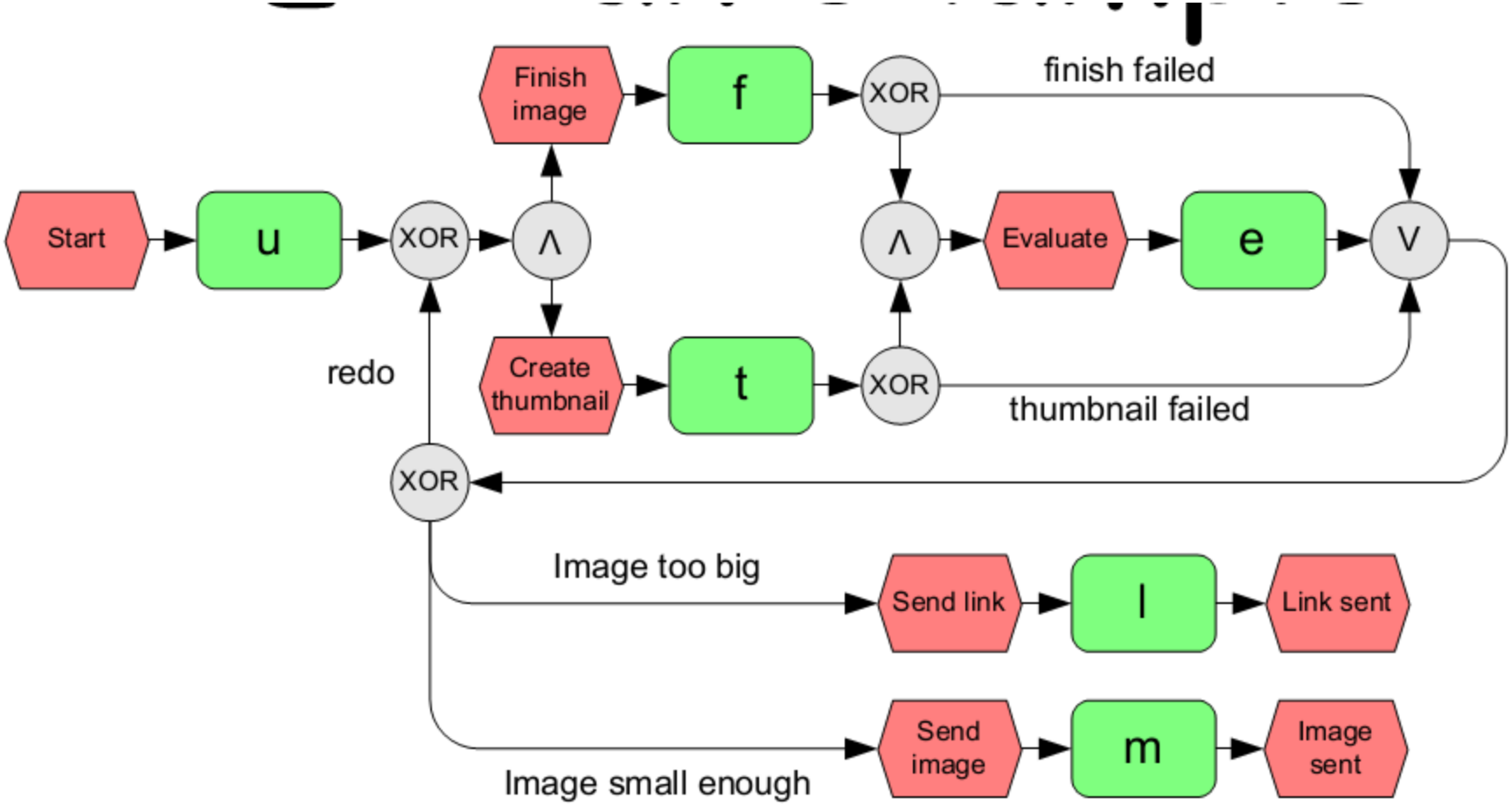


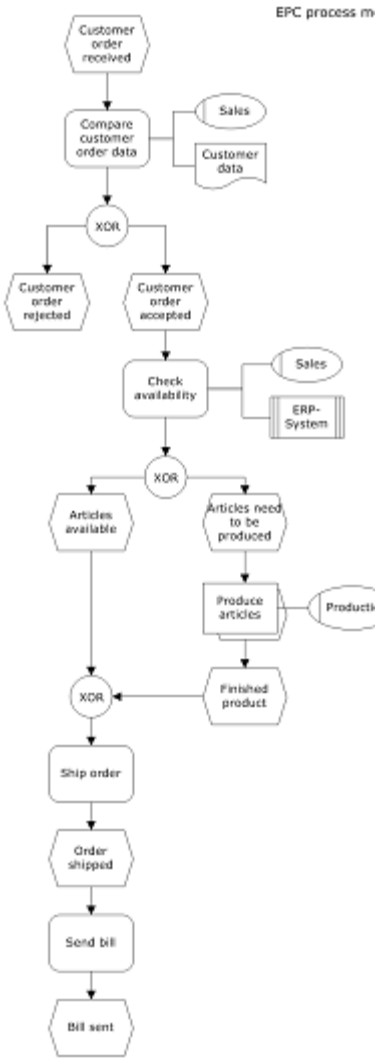
from M. Weske: Business Process Management, © Springer-Verlag Berlin Heidelberg 2007

Resulting Workflow-net: Sound?



from M. Weske: Business Process Management, © Springer-Verlag Berlin Heidelberg 2007


Start
u
XOR
Finish image
f
XOR
finish failed
Create thumbnail
t
XOR
thumbnail failed
redo
Evaluate
e
v
XOR
Image too big
Send link
l
Link sent
Image small enough
Send image
m
Image sent


EPC process m
Customer order received
Compare customer order data
Sales
Customer data
XOR
Customer order rejected
Customer order accepted
Check availability
Sales
ERP-System
XOR
Articles available
Articles need to be produced
Produce articles
Finished product
XOR
Ship order
Order shipped
Send bill
Bill sent