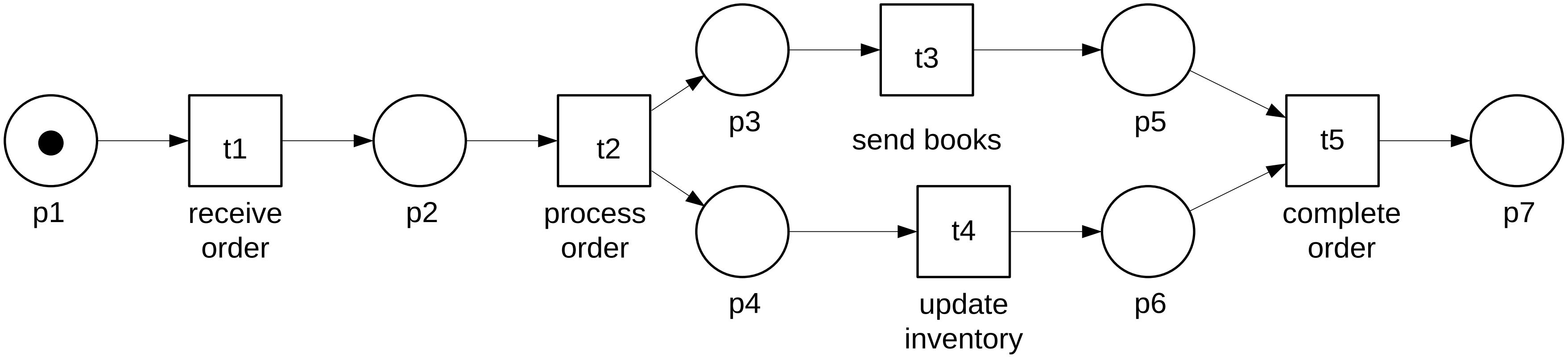
Petri nets

- Idea
  - Formal representation of concurrent systems
  - Formal model and graphical representation are equivalent in classical Petri nets
- Remarks
  - Developed by Carl Adam Petri, 1962
  - Many variations and extensions
  - Here: Application for modeling business processes and their analysis

Petri nets

- Characterization
  - Petri net is a directed graph consisting of places, transitions and directed edges between them
  - Petri nets are bipartite graphs, i.e., edges between two places and edges between two transitions are not allowed

- Example Petri net



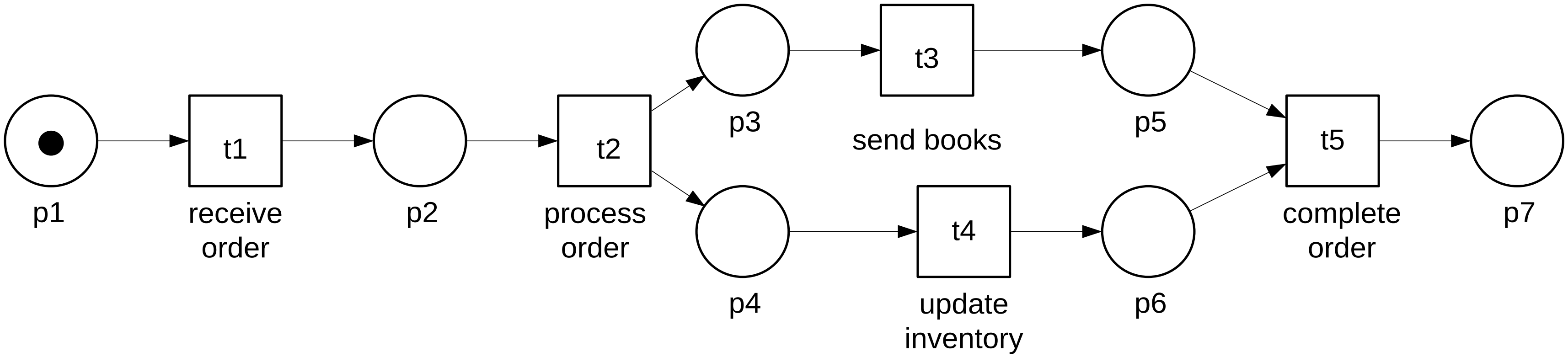
from M. Weske: Business Process Management, © Springer-Verlag Berlin Heidelberg 2007

Definition Petri net

Definition 4.1 A *Petri net* is a tuple (P, T, F) with

- a finite set P of places
- a finite set T of transitions such that $T \cap P = \emptyset$
- a flow relation $F \subseteq (P \times T) \cup (T \times P)$
- A place $p \in P$ is an input place of a transition $t \in T$, if and only if there exists a directed arc from p to t, i.e., if and only if $(p, t) \in F$. The set of input places for a transition t is denoted $\bullet t$.
- A place p is an output place of a transition t, if and only if there exists a directed arc from t to p, i.e., if and only if $(t, p) \in F$. The set of output places for a transition t is denoted $t\bullet$.
- $p\bullet$ and $\bullet p$ denote the set of transitions that share p as input places or output place, respectively.

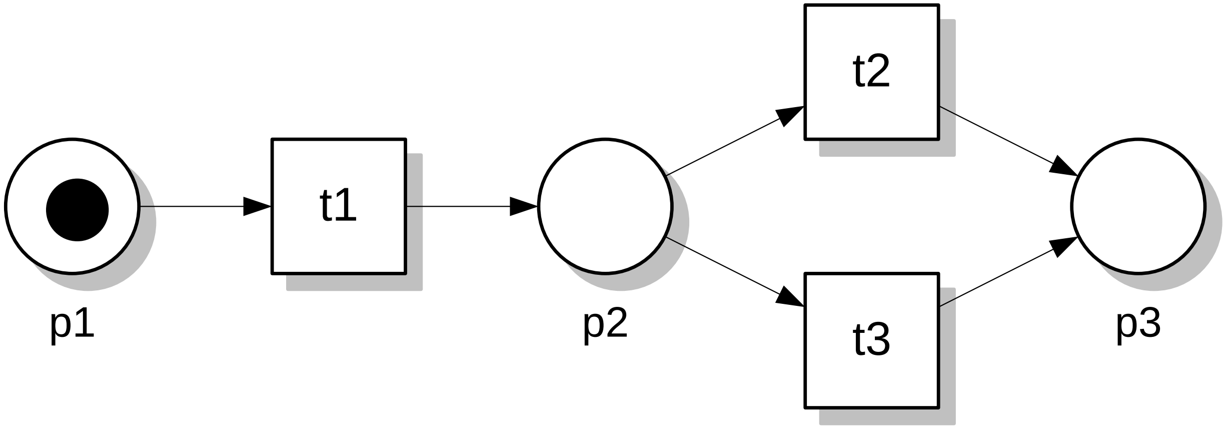
Example



$$P = \{p1, p2, p3, p4, p5, p6, p7\},$$
$$T = \{t1, t2, t3, t4, t5\},$$
$$F = \{(p1, t1), (t1, p2), (p2, t2), (t2, p3), (t2, p4), (p3, t3), (p4, t4), (t3, p5), (t4, p6), (p5, t5), (p6, t5), (t5, p7)\}.$$

Marking

- The dynamic behavior of a system is represented by tokens in the Petri net
- The state of a Petri net (marking) is described by the distribution of tokens on places



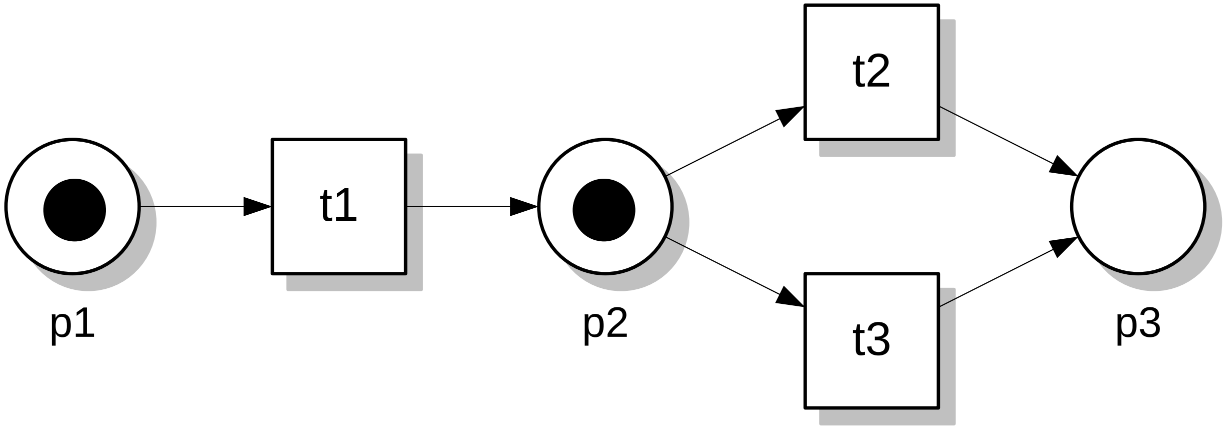
Definition 4.2 The *marking* (or *state*) of a Petri (P, T, F) net is defined by a function $M : P \rightarrow \mathbb{N}$ mapping the set of places onto the natural numbers, where $\mathbb{N}$ is the set of natural numbers including zero. $\diamond$

Condition Event Nets

- Idea
  - Places represent conditions
  - Transitions represent events
- Consequences
  - A condition is fulfilled IFF a token is in the place
  - At any time, any place holds at most one token

Execution Semantics

- Firing transitions
  - Transitions can fire when they are enabled
  - Transitions are enabled when every input place contains one token
  - Firing a transition removes one token from each input place and puts a token on each output place
- Note
  - Different classes of Petri nets have different firing rules



Definition: Condition Event Net

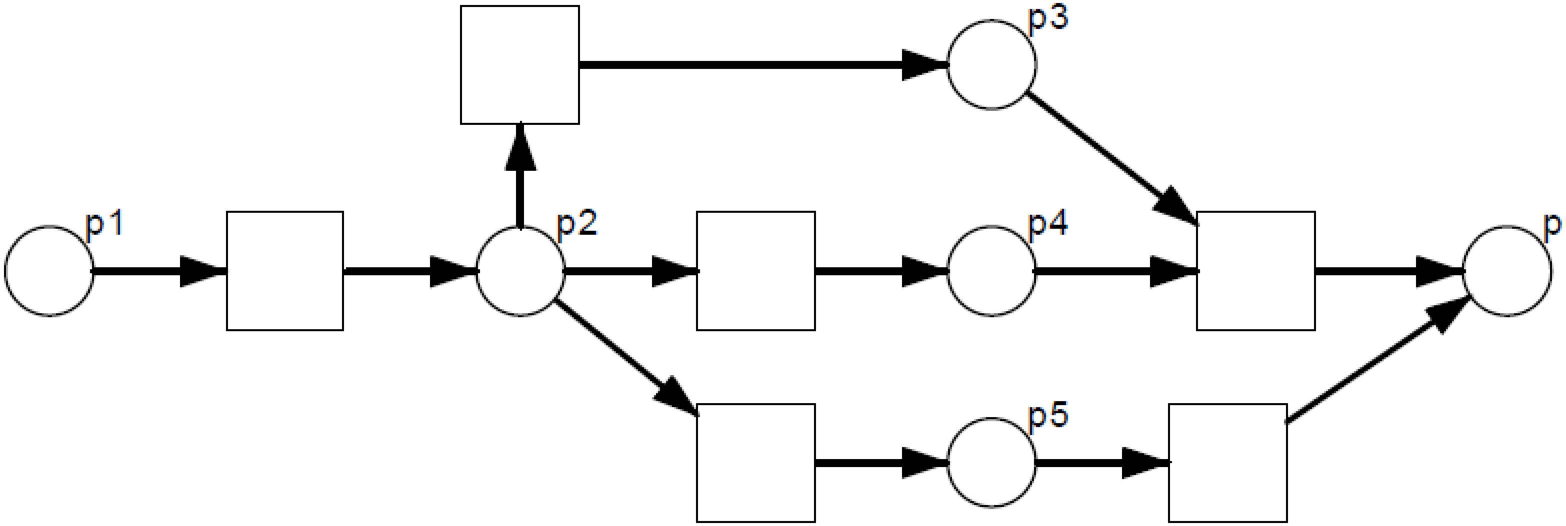
Definition 4.4 A Petri net (P, T, F) is a *condition event net*, if $M(p) \leq 1$ for all places $p \in P$ and for all states M.

- A transition t is *enabled* in a state M if $M(p) = 1$ for all input places p of t and $M(q) = 0$ for all output places q of t that are not input places at the same time.
- The firing of a transition t in a state M results in state M', where

$$(\forall p \in \bullet t) M'(p) = M(p) - 1 \wedge (\forall p \in t \bullet) M'(p) = M(p) + 1$$

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

Enablement and Reachability



Definition: Enablement and Reachability

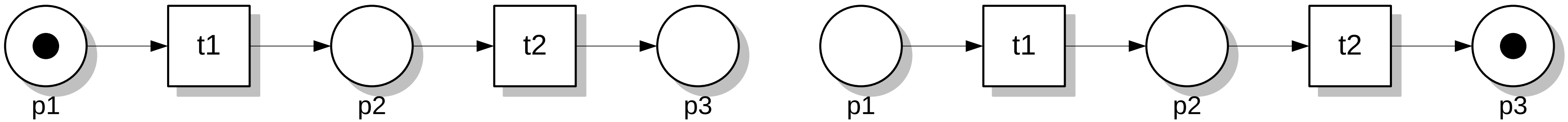
Definition 4.3 Let (P, T, F) be a Petri net and M a marking. The firing of a transition is represented by a state change of the Petri net.

- $M \xrightarrow{t} M'$ indicates that by firing t the state of the Petri net changes from M to M'.

- $M \rightarrow M'$ indicates that there is a transition t such that $M \xrightarrow{t} M'$.
- $M_1 \xrightarrow{*} M_n$ means that there is a sequence of transitions $t_1, t_2, \ldots t_{n-1}$ such that $M_i \xrightarrow{t_i} M_{i+1}$, for $1 \leq i < n$.
- A state M' is *reachable* from a state M, if and only if $M \xrightarrow{*} M'$.

Execution Semantics

- A Petri net system is a pair (PN,M) with
  - Petri net PN = (P,T,F)
  - Initial Marking M
- Let (({p1,p2,p3}, {t1,t2}, {(p1,t1), (t1,p2), (p2,t2), (t2,p3)}),(1,0,0)) be a Petri net system
  - $M_1 \xrightarrow{o} M_3$ where $o=(t_1 t_2)$ transfers Petri net from Marking M_1 to Marking M_3

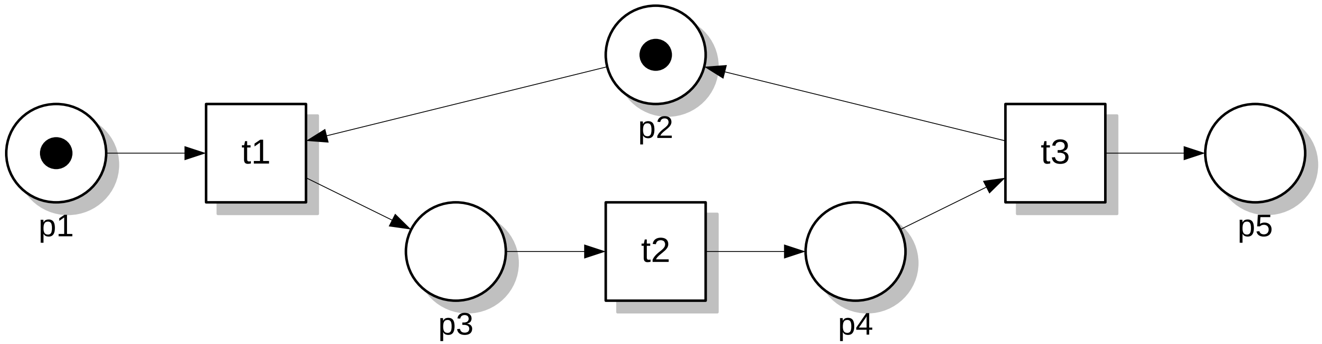


M_1 : (1,0,0)

M_3 : (0,0,1)

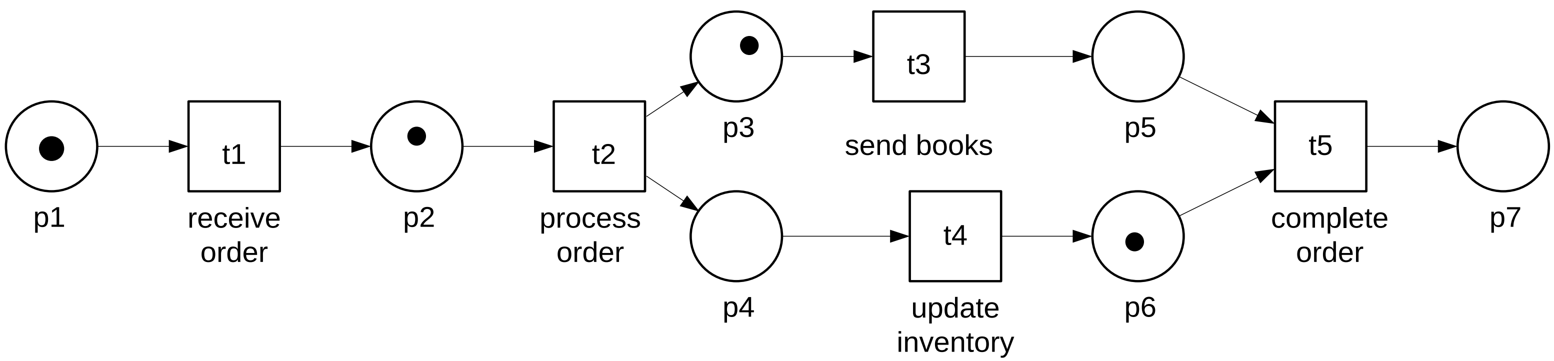
Reachability

- Example
  - (0,1,0,0,1) is reachable from (1,1,0,0,0) via o=(t1,t2,t3)
  - (0,1,1,0,0) is not reachable from (1,1,0,0,0) because there is no corresponding sequence of transition firing



Process Instances in Petri Nets

- Idea
  - Each process instance is represented by a set of tokens
  - Each token belongs to exactly one process instance
- Problems of classical Petri nets
  - Tokens are not distinguishable
  - Several process instances represented by a Petri net, but C / E nets do not allow independent process progress



Place/Transition Nets

- Idea
  - In each place, an arbitrary number of tokens can reside
  - The output places of an enabled transition may contain tokens
  - Thus, several process instances in a Petri net can be represented
    - What further condition must be met?
  - Edges can be weighted; the firing behavior of transitions depends on the edge weights
- Place/Transition nets
  - Allow many tokens in a place
  - True extension of C/E nets

Place/Transition Nets

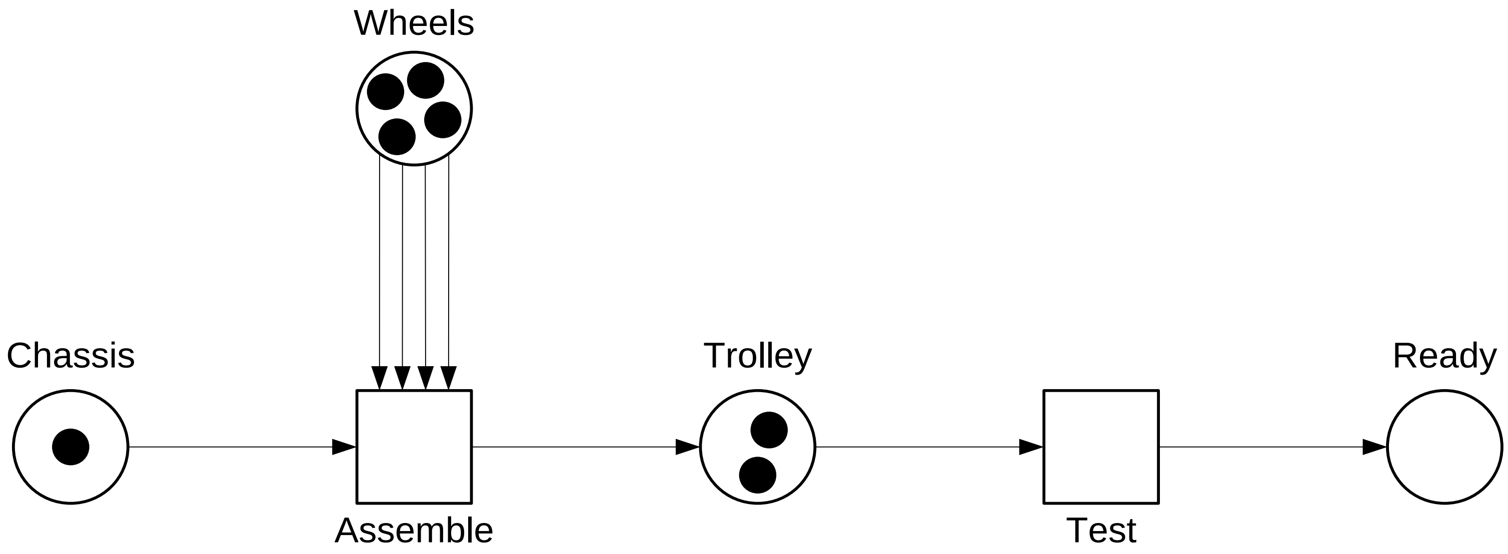
Definition 4.5 (P, T, F, ω) is a *place transition net* if (P, T, F) is a Petri net and $\omega : F \rightarrow \mathbb{N}$ is a weighting function that assigns a natural number to each arc, the *weight* of the arc.

The dynamic behaviour of place transition is defined as follows:

- A transition t of a place transition net is *enabled* if each input place p of t contains at least the number of tokens defined as the weight of the connecting arc, i.e., if $M(p) \geq \omega((p,t))$.
- When a transition t fires, the number of tokens withdrawn from its input places and the number of tokens added to its output places are determined by the weights of the respective arcs.
- From each input place p of t, $\omega((p,t))$ tokens are withdrawn and $\omega((t,q))$ tokens are added to each output place q.
- The firing of a transition t in a state M results in state M', where

$$(\forall p \in \bullet t) M'(p) = M(p) - \omega((p,t)) \wedge (\forall p \in t \bullet) M'(p) = M(p) + \omega((t,p))$$

Place/Transition Nets: Example

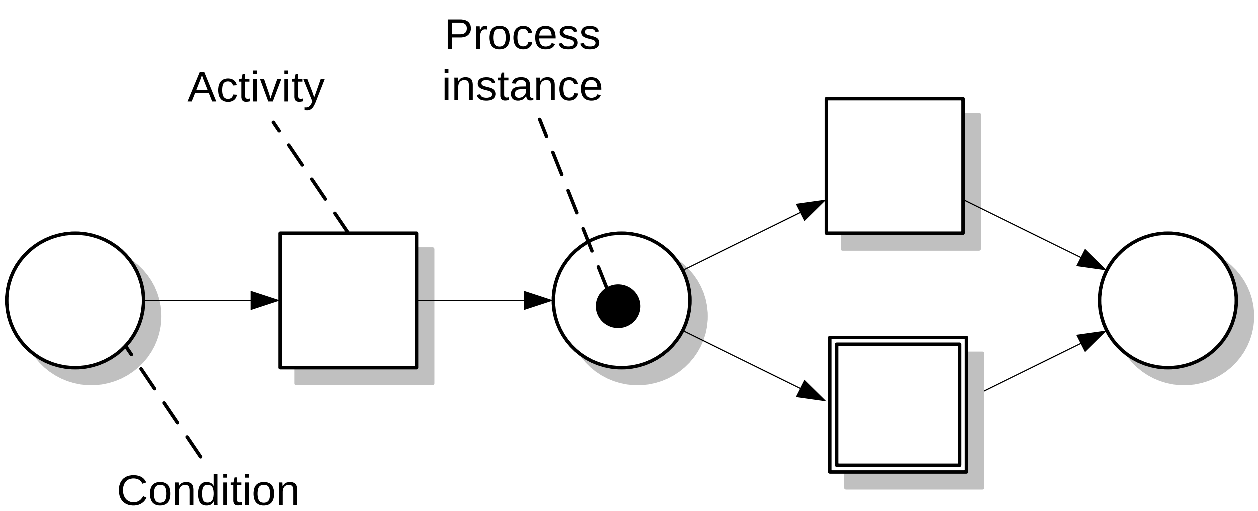


Workflow Nets

- Idea
  - Using Petri nets to model business processes
- Illustration of concepts
  - Transitions represent activities
  - Places represent states
  - Edges represent the control flow
  - Tokens can carry structured values
  - Process instances' behavior is represented by firing rules

Example Workflow Net

- Activities, conditions, process instances
- Nested activities represented by transitions with double border
- Beispiel
  - XOR-Split expressible by classical firing rule of transitions



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

Workflow net: Characterization

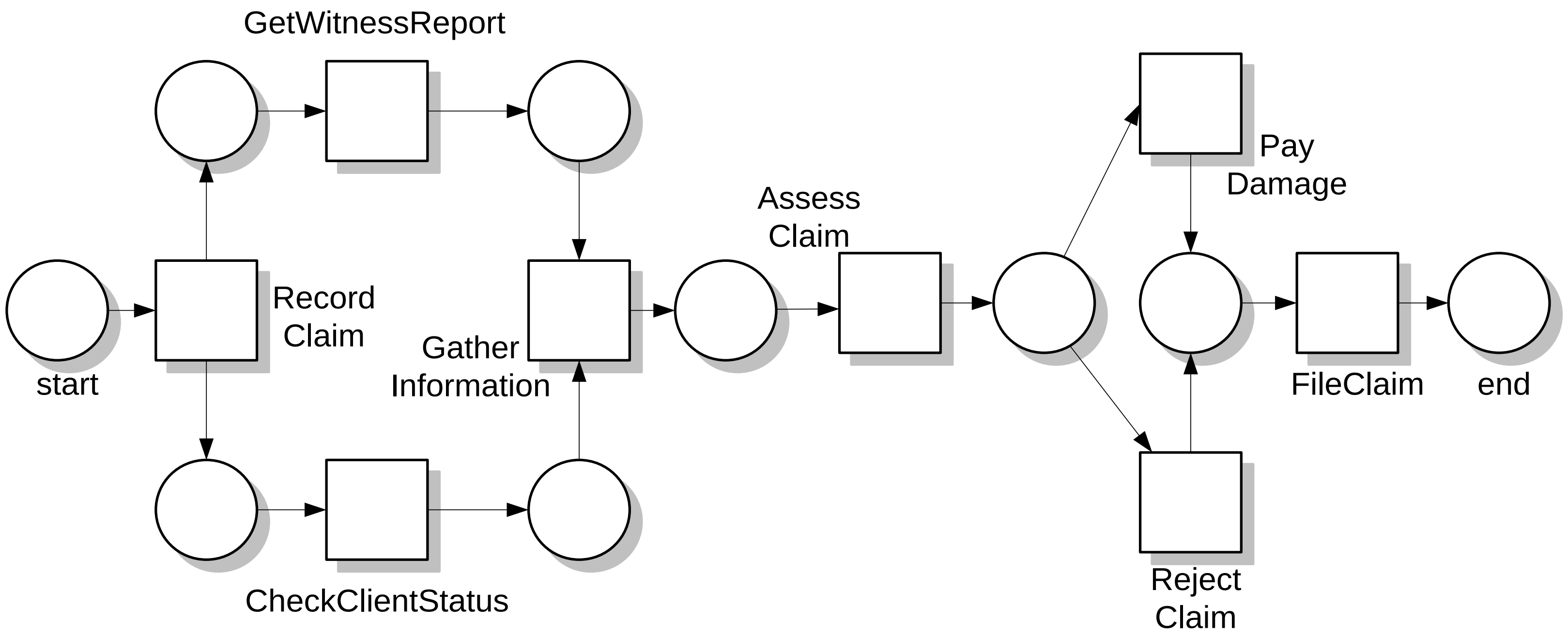
- A Workflow net is a Petri net with
  - A specific input place i (the initial place)
  - A specific output place o (the final place)
  - For i, no incoming edges as well as o has no outgoing edges
- Remarks
  - A token in i represents a not-yet started process instance
  - A token in o represents a terminated process instance
  - Each process instance is represented by a token flow from i to o

Workflow Nets: Definition

Definition 4.8 A Petri net $PN = (P, T, F)$ is called *workflow net*, if and only if the following conditions hold:

- There is a distinguished place $i \in P$ (called initial place) that has no incoming edge, i.e., $\bullet i = \emptyset$.
- There is a distinguished place $o \in P$ (called final place) that has no outgoing edge, i.e., $o\bullet = \emptyset$.
- Every place and every transition is located on a path from the initial place to the final place.

Example Workflow Net



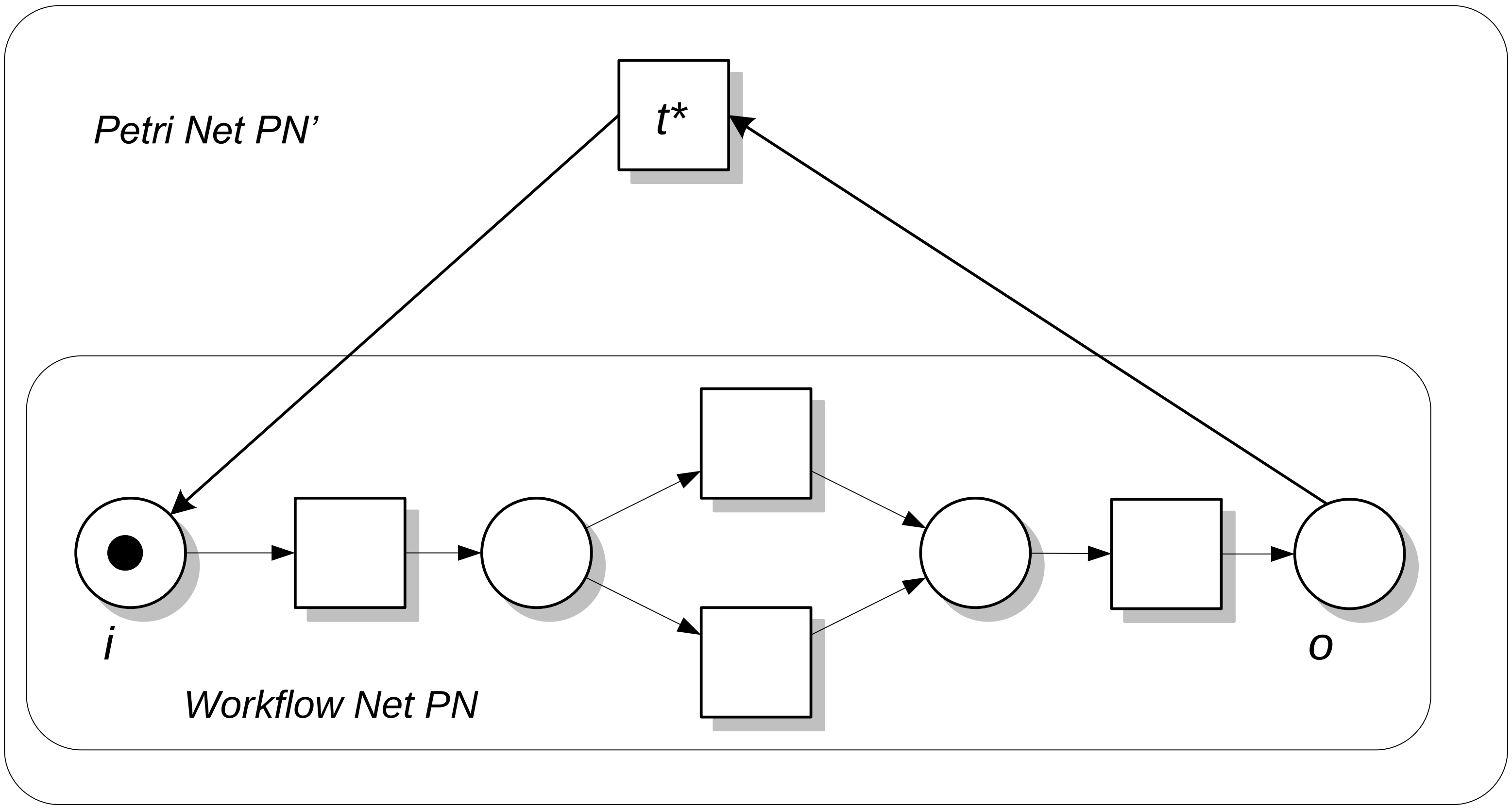
from M. Weske: Business Process Management, © Springer-Verlag Berlin Heidelberg 2007

Properties of Workflow Nets

1. i is the only initial place: If PN is a Workflow net with initial place i, for all $p \in P$: $\bullet p \neq \emptyset$ or $p = i$
2. o is the only final place: If PN is a Workflow net with final place o, for all $p \in P$: $p\bullet \neq \emptyset$ or $p = o$
3. Let PN be a Workflow net. If we add a transition t*, which connects o to i (i.e. $\bullet t^* = \{o\}$ and $t^*\bullet = \{i\}$), the resulting Petri net is strongly connected

- Remark
  - A Petri net is strongly connected if for any pair of nodes x,y a path from x to y exists.

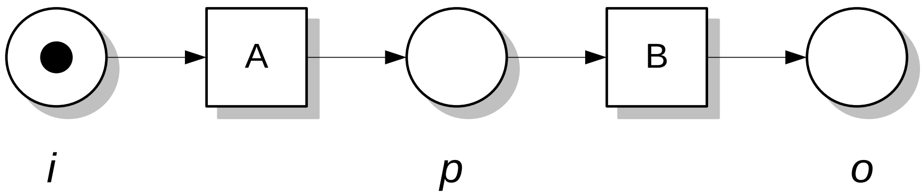
Workflow Nets: Property 3



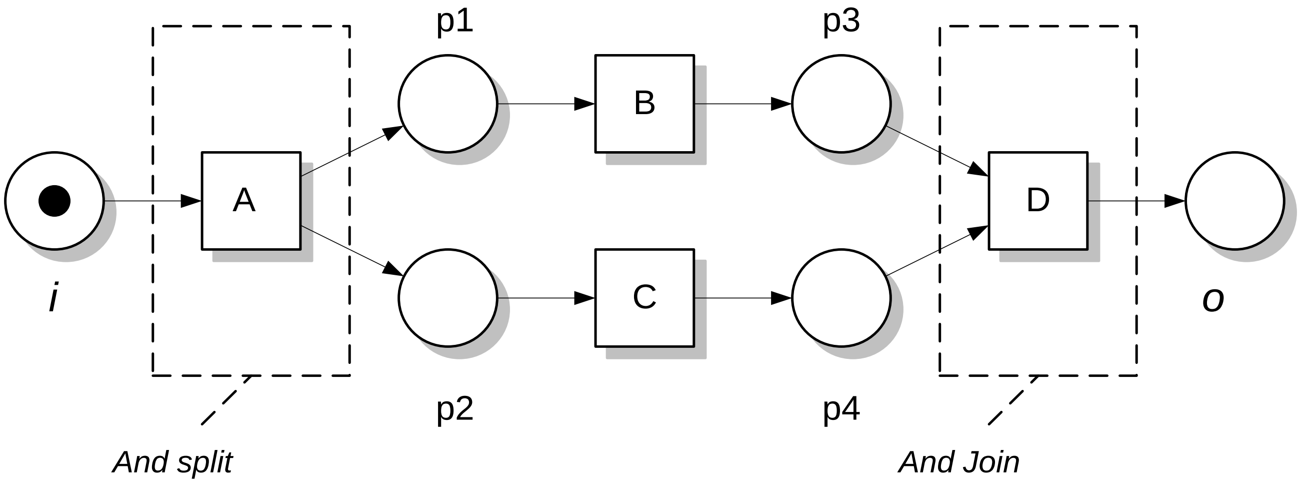
from M. Weske: Business Process Management, © Springer-Verlag Berlin Heidelberg 2007

Control Structures in Workflow Nets

- Sequence



- AND Split / AND Join



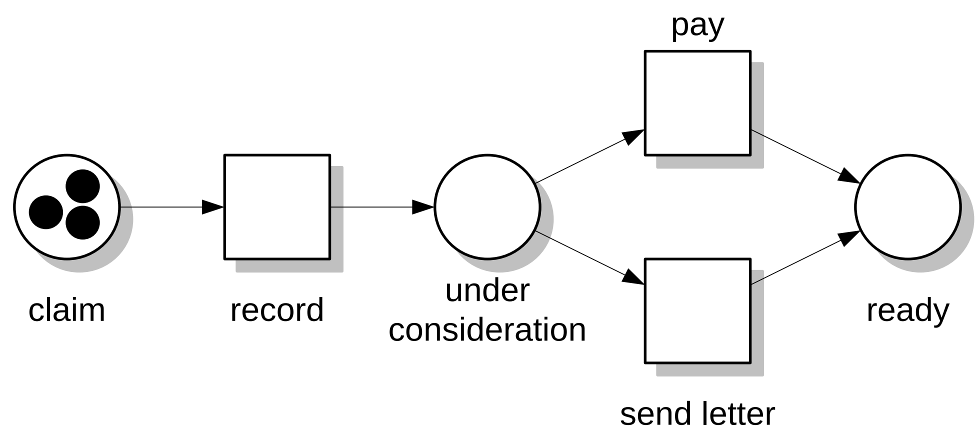
Analysis of Workflow Nets

- Idea
  - Generic structural correctness criteria for workflow nets
  - Undesirable behavior of process instances is thus excluded
- Basics
  - Reachability analysis: Which states can be reached?
  - What properties do these states possess?

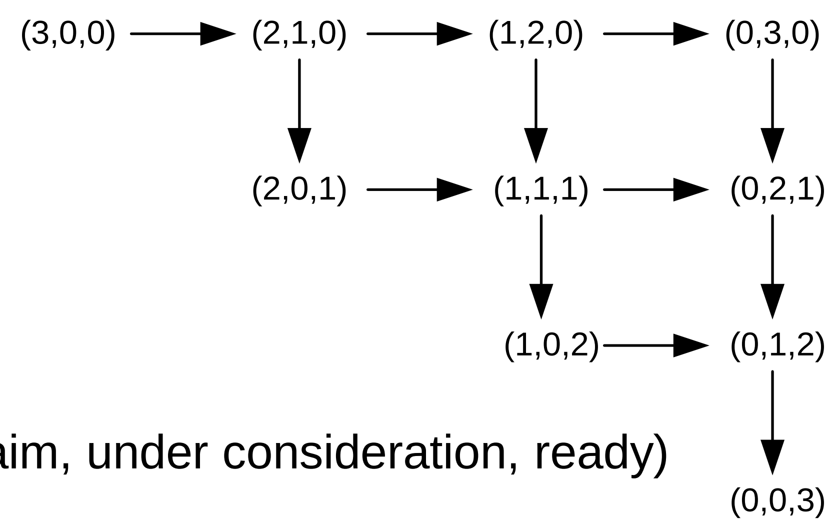
Reachability

- Idea
  - A Petri net system determines the reachable states
- Hint
  - Communication with the environment is not considered here
- Representation of the reachability graph
  - Nodes represent states
  - Edges represent state transitions, by firing transitions
  - Multiple outgoing edges: non-deterministic behavior
- Naïve technique
  - Manual creation of the reachability graph and analysis

Example(1)

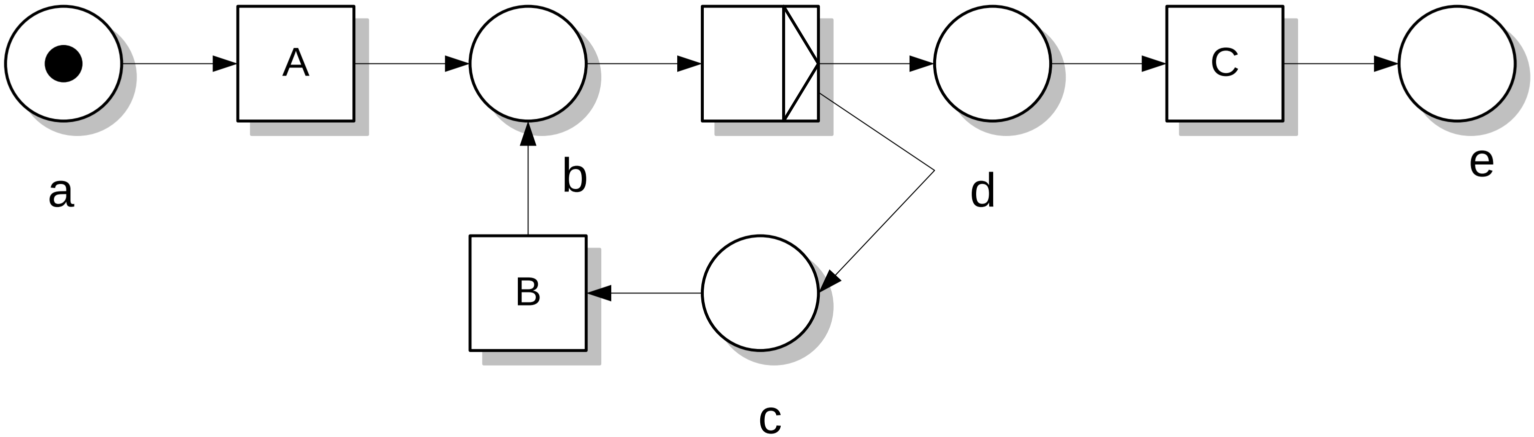


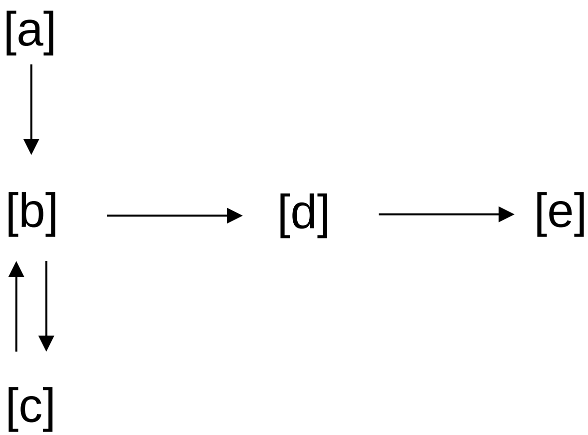
- Initial state (3,0,0) is always transferred to the final state (0,0,3)



State representation: (claim, under consideration, ready)

Example (2)

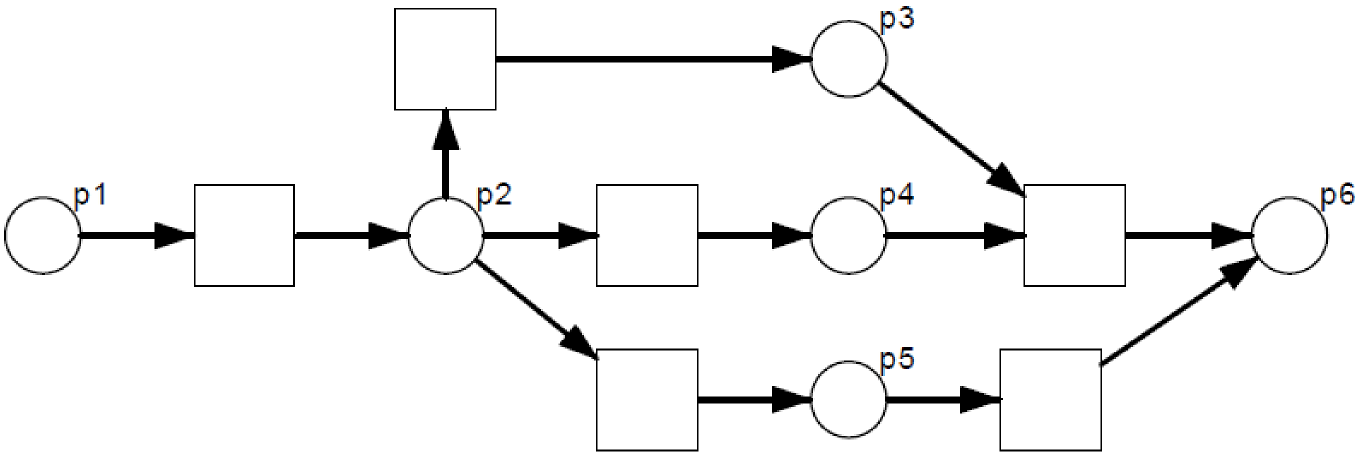




Hint

Compact representation of the state by places with tokens

Enablement and Reachability



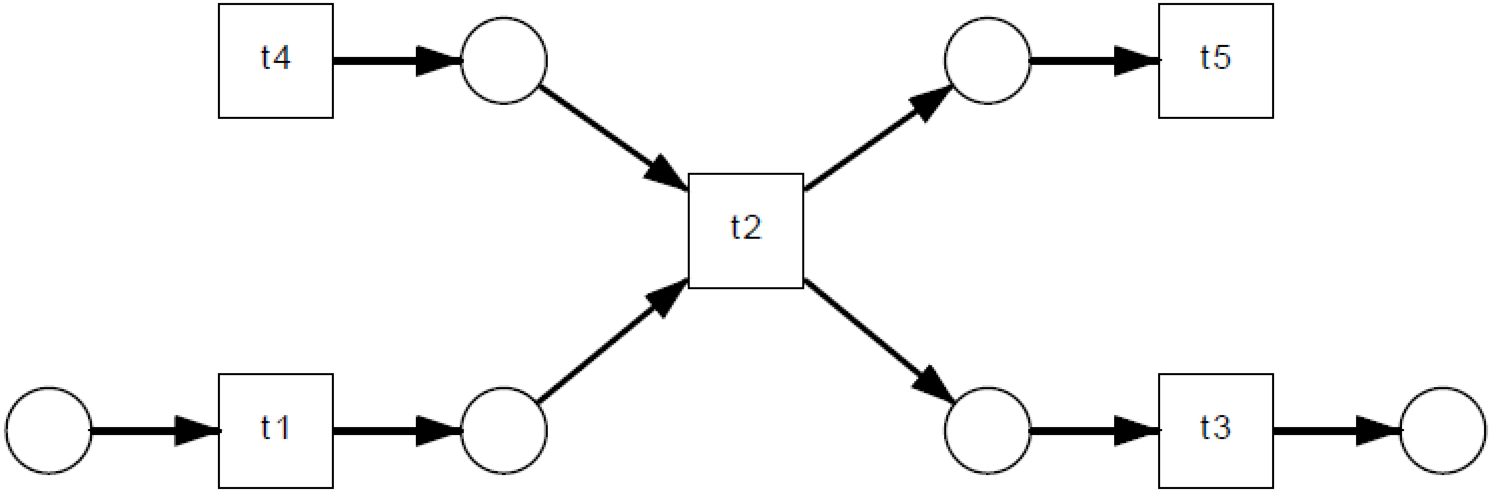
Structural Analysis

- Idea
  - Structural analysis of Workflow nets to find errors

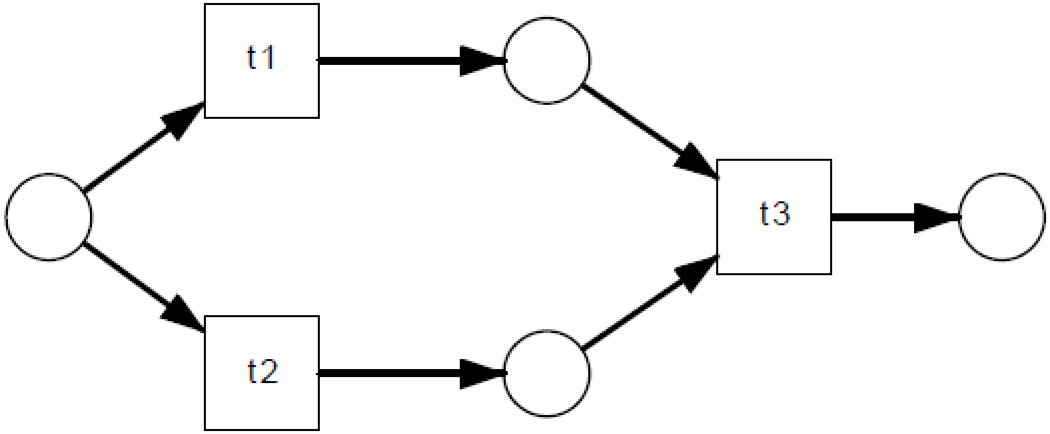
- Error possibilities
  - Transitions without in-output places
  - Transitions that can never fire (*dead transitions*)
  - Deadlocks, which prevent the process progress
  - Endless loops (*livelock*)
  - Activities that are performed after the end of the process

Structural Errors

- Dangling tasks

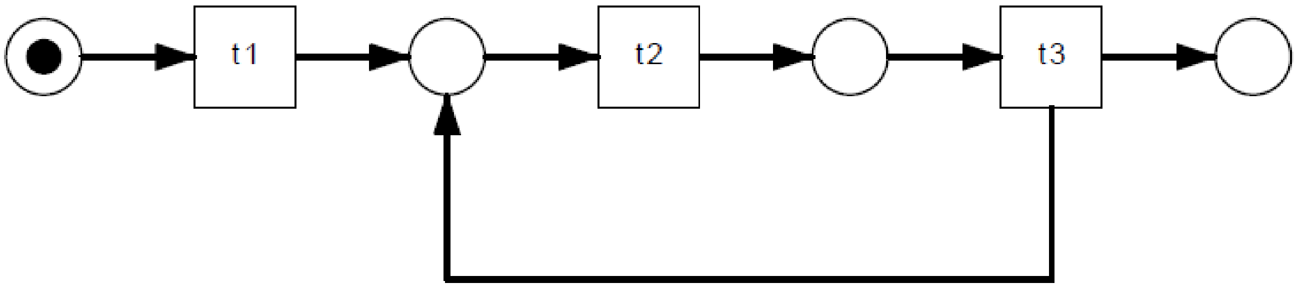


- Deadlock

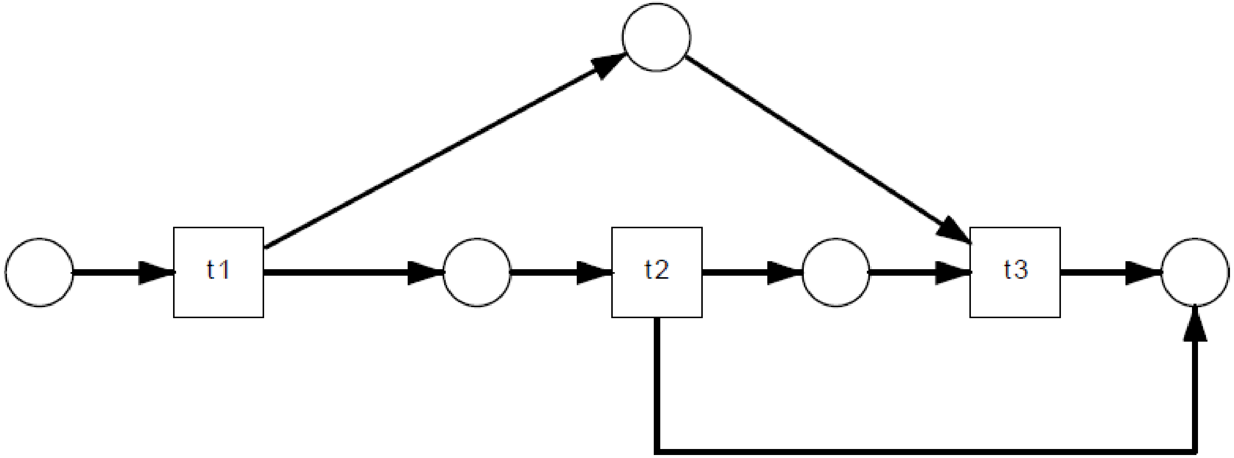


Structural Errors

- Livelock



- Remaining Tokens



Soundness-Property

- Soundness
  - (1) For each token on the initial place exactly one token appears eventually on the final place
  - (2) If a token appears on the final place, all other places are empty
  - (3) Each transition can be enabled
- Soundness based on fairness assumption
  - Each possible decision is finally met
  - Consequence
    - No transition starvation
    - Based on this assumption, the behavior of CPN is simulated

Definition

Definition 6.2 (States o,i and Relations $\geq, >$) Let $PN = (P, T, F)$ be a workflow net, $i \in P$ its start place and $o \in P$ its end place and M, M' markings.

- o is the state in which there is exactly one token in place $o \in P$ and no tokens in any other place of the workflow net
- i is the state in which there is exactly one token in place $i \in P$ and no token in any other place of the workflow net
- $M \geq M'$, if and only if $M(p) \geq M'(p), \forall p \in P$
- $M > M'$, if and only if $M \geq M' \wedge \exists p \in P : M(p) > M'(p), p \in P$

Soundness: Definition

Definition 6.3 A workflow system (PN, I) with a workflow net $PN = (P, T, F)$ is *sound*, if and only if

- For every state M reachable from state i there exists a firing sequence leading from M to o, i.e.,

$$\forall M (i \xrightarrow{*} M) \implies (M \xrightarrow{*} o)$$

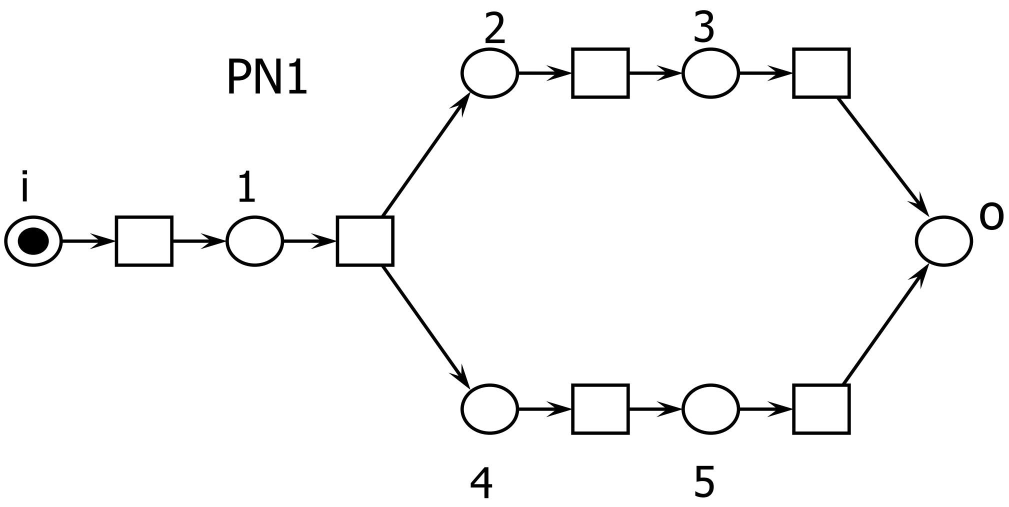
- State o is the only state reachable from state i with at least one token in place o, i.e.,

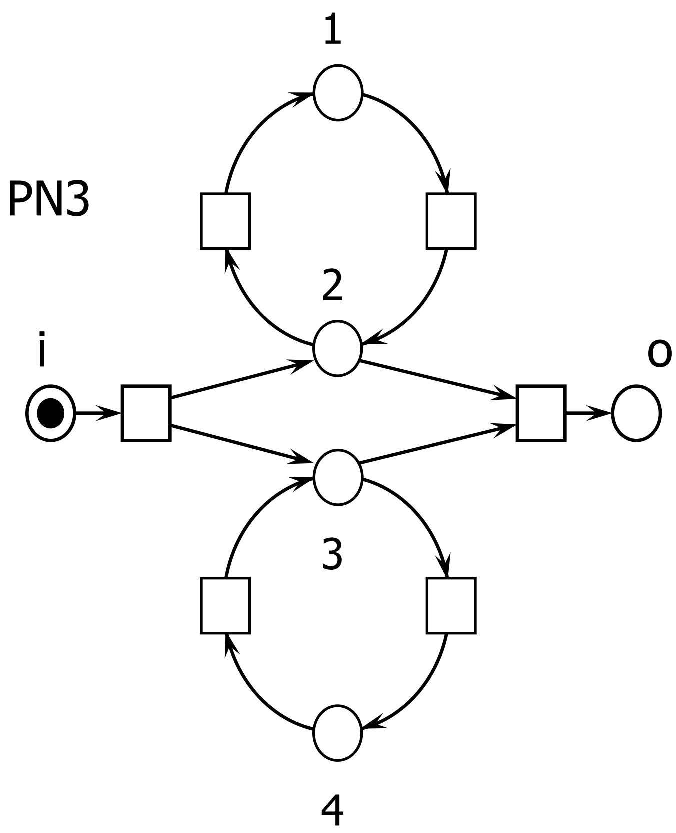
$$\forall M (i \xrightarrow{*} M \wedge M \geq o) \implies M = o$$

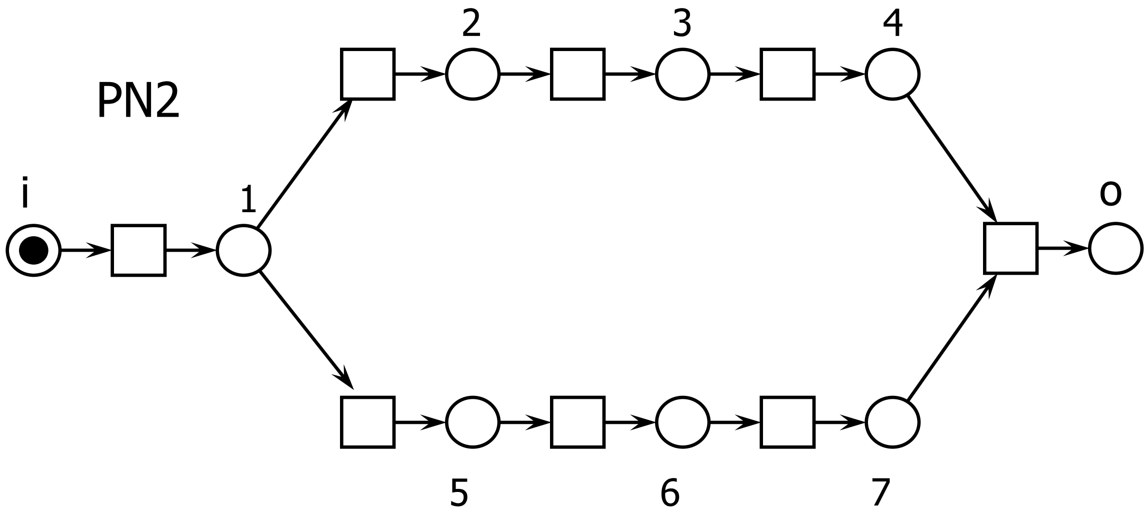
- There are no dead transitions in the workflow net in state i, i.e.,

$$(\forall t \in T)\ \exists M, M' : i \xrightarrow{*} M \xrightarrow{t} M'$$

Examples





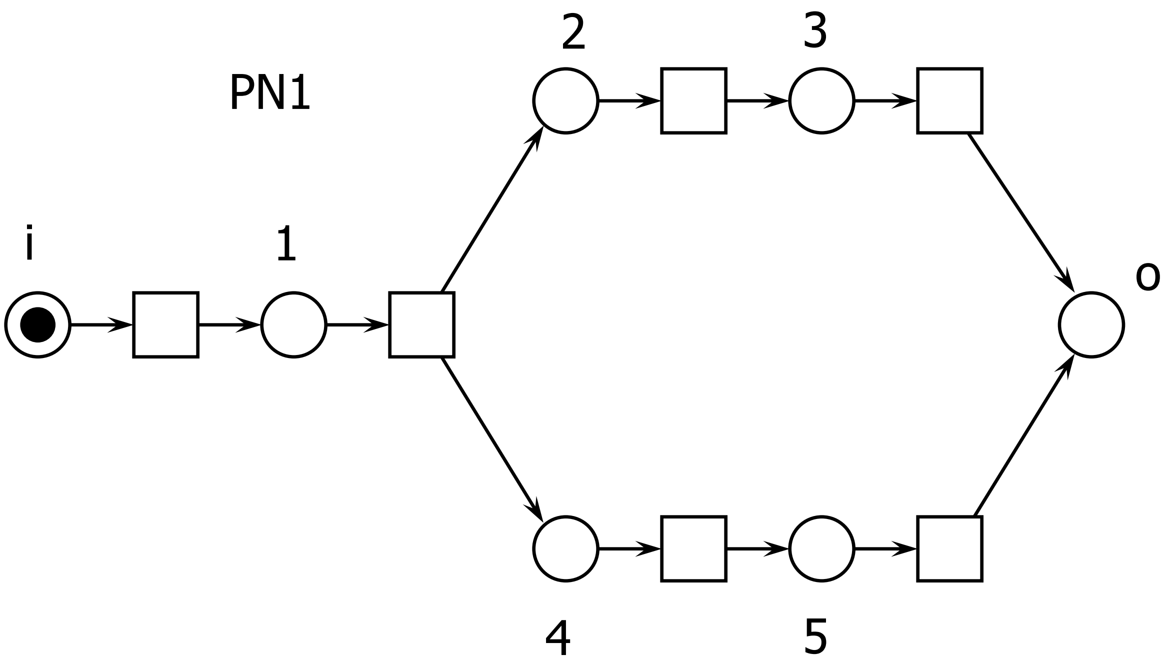


Soundness

- Check with Reachability analysis
  - Construction of reachability graph with initial state i
- Check Procedure
  - (1) Check if there is a path from any node to o
  - (2) Check if only in a state o at least one token is only in place o
  - (3) Check if every transition occurs in the reachability graph

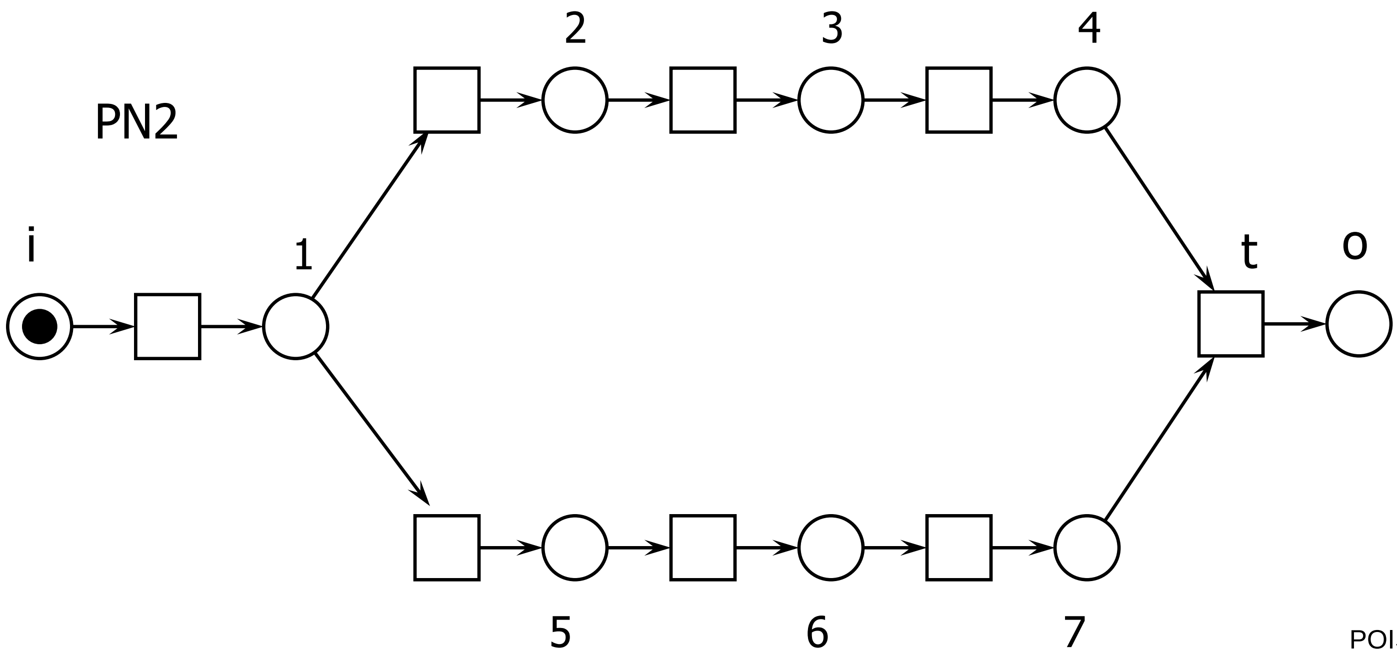
Examples

- PN1 ist not sound
  - After reaching the final place o, there are remaining tokens in the net
  - The reachable state (0,0,0,0,1,0,1) [Format: (i,1,2,3,4,5,o)] vioaltes condition (2)



Examples

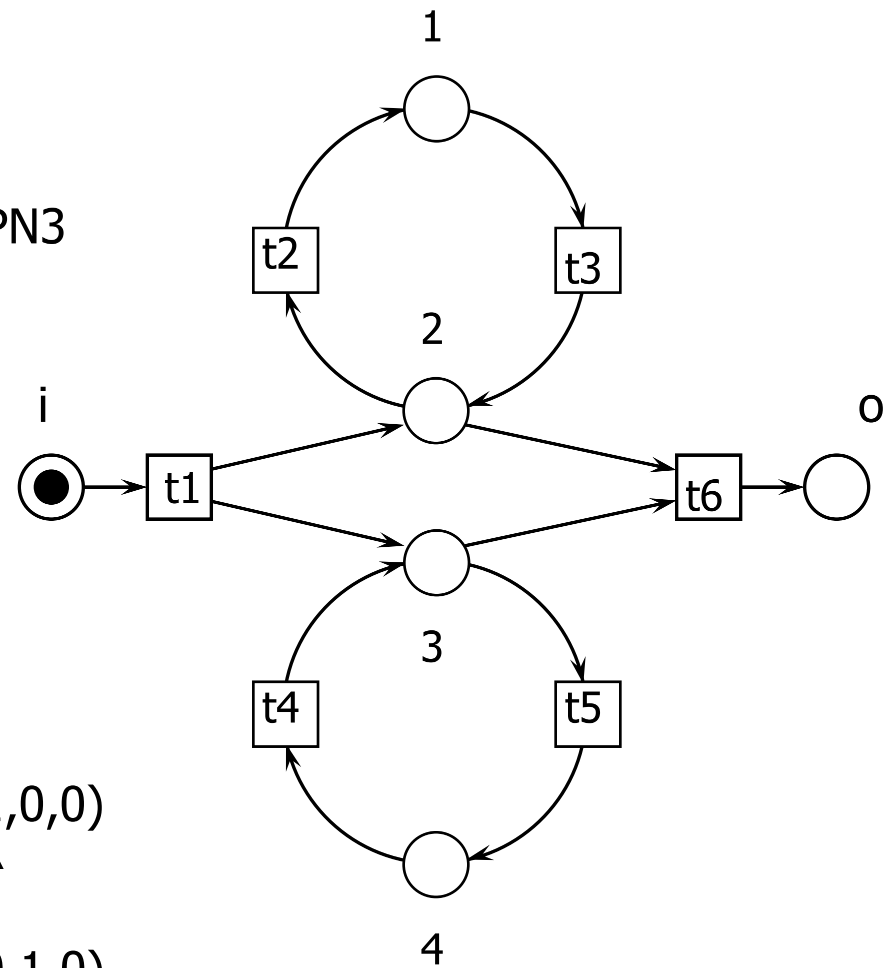
- PN2 is not sound
  - No termination
  - State (0,0,0,0,0,0,0,1,0) is Deadlock-state, violating condition (1)
  - Transition t can never fire, violating condition (3)

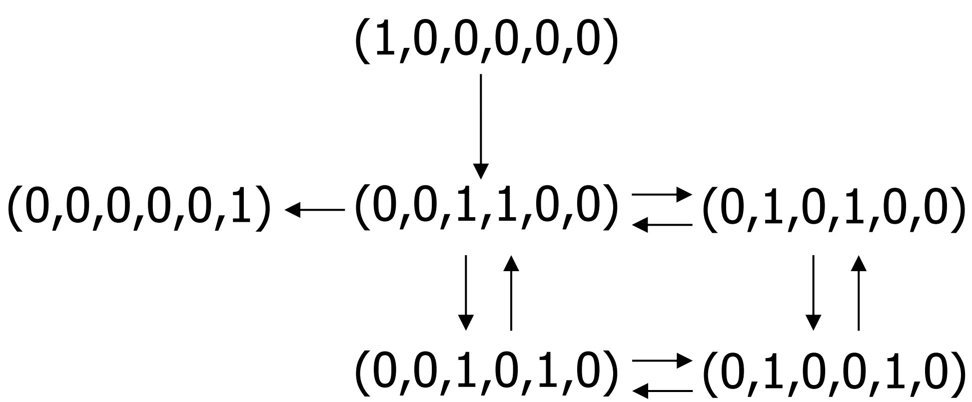


Examples

- PN3 sound
  - (1), (2) and (3) are fulfilled
  - Proof by reachability graph

PN3

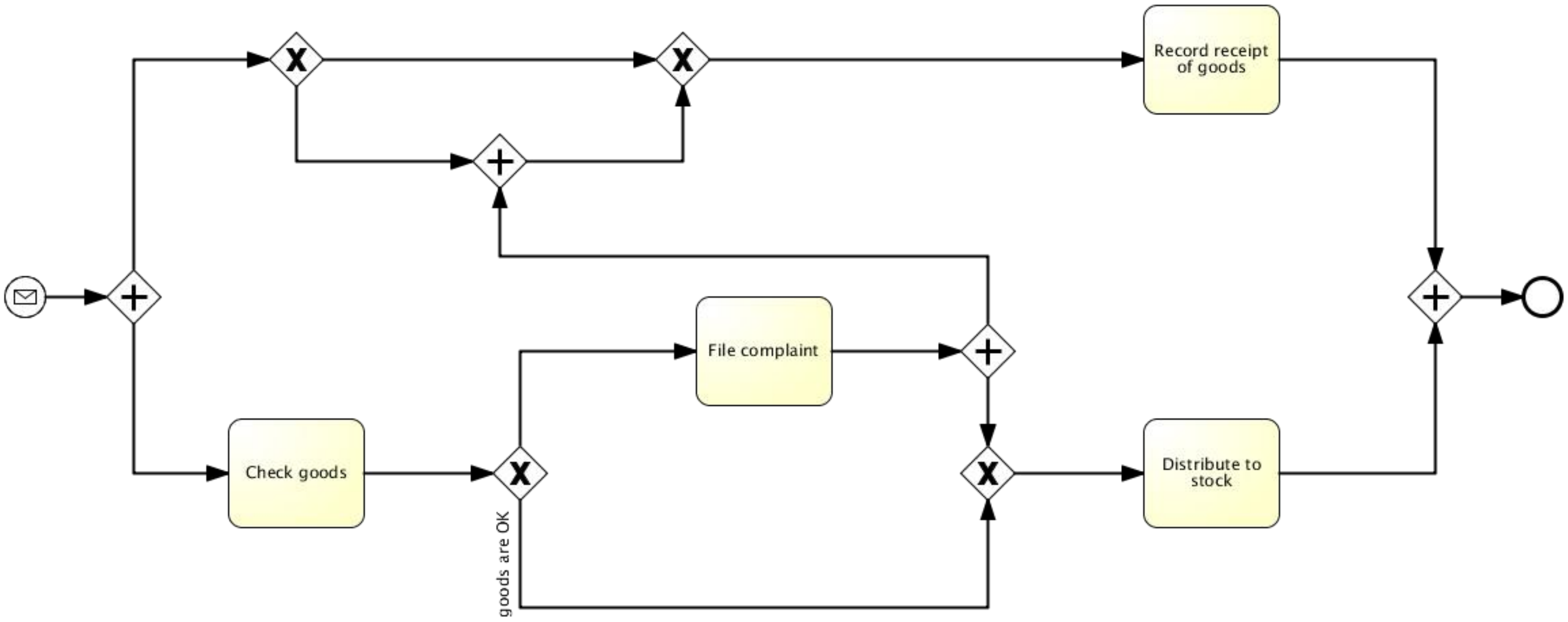




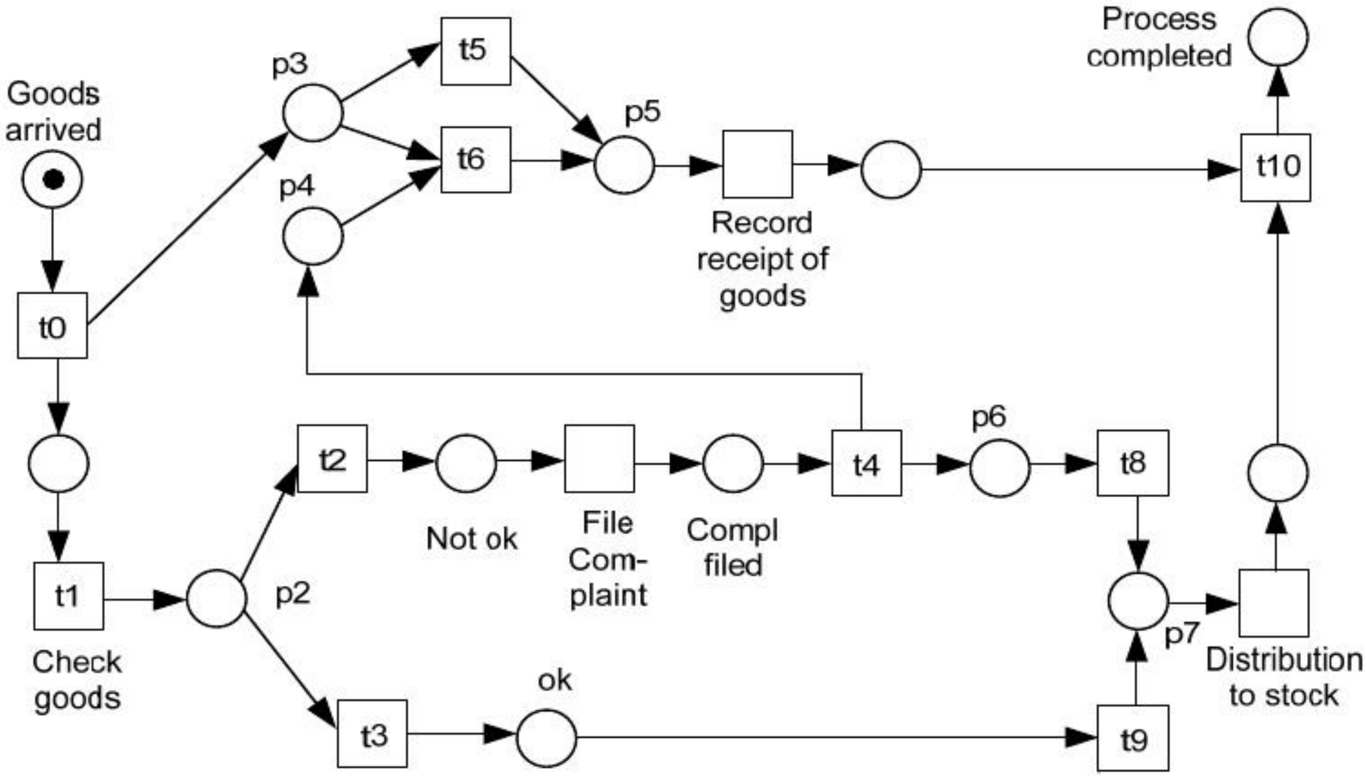
Relaxed Soundness

- Observation
  - *Soundness is very strong criterion that is not appropriate in every case*
  - Some times a high degree of freedom in the modeling process is needed
    - This can lead to processes that are not sound
    - Nevertheless, it is meaningful to verify
- Idea
  - Weakening soundness so that not all process instances must be sound, but every transition is involved in at least one process instance which is sound.

Motivation



- Process instances can run into deadlocks
- Yet, each task participates in a sound execution


Goods arrived
t0
p3
p4
t5
t6
p5
Record receipt of goods
Process completed
t10
t1
Check goods
p2
t2
Not ok
File Com-plaint
Compl filed
t4
p6
t8
p7
Distribution to stock
t3
ok
t9

Definition *Relaxed Soundness*

Definition 6.6 Let $S = (PN, i)$ be a workflow system. Let σ, σ' be firing sequences and let M, M' be states. σ is a *sound firing sequence* if it leads to a state from which a continuation to the final state o is possible: $i \xrightarrow{\sigma} M$ and $\exists \sigma'$ such that $M \xrightarrow{\sigma'} o$. ◇

Definition 6.7 A workflow system $S = (PN, i)$ is *relaxed sound* if and only if each transition of PN is an element of some sound firing sequence:

$$\forall t \in T \; \exists M, M' \; : \; (i \xrightarrow{*} M \xrightarrow{t} M' \xrightarrow{*} o)$$

◇