Petri Nets

Barbara Re

Carl Adam Petri

- Introduced in 1962 (Petri's PhD thesis)
- 60's and 70's main focus on theory 80's focus on tools and applications
- Now applied in several fields
- Success due to simple and clean graphical and conceptual representation

(1926 – 2010)

Formal and abstract business process specification

- **Formal**: the semantics of process becomes well defined and not ambiguous
- **Abstract**: execution environment is disregarded (Remind about separation of concerns)

- Petri Net can be used to model dynamic system with a static structure:
  - The static structure is represented by a Petri Net (process level)
  - The dynamic behavior is capture by the token play of the Petri Net (instance level)

Basic Elements

Place

A place can stand for a state a medium a buffer a condition a repository of resources a type …

Transition

A transition can stand for an event an operation a transformation a transportation a task an activity …

Token

A token can stand for a physical object a piece of data a resource an activation mark a message a document a case …

Petri Net - Definition

A Petri Net is a tuple (P, T, F, M_o) where

- P is a finite set of places
- T is a finite set of transitions
- $F \subseteq (P \times T) \cup (T \times P)$ is a flow relation
- $M : P \rightarrow N$ is the initial marking

Marking

- A Marking M: P → N denotes the number of tokens in each place
- The marking of a Petri net represents its state
- *M (a) = o* denotes the absence of tokens in a place *a*

Firing a trasition

- 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 → M′ indicates that by firing t, the state of the Petri net changes from M to M′
  - M → M′ indicates that there is a transition t such that M →M′
  - $M_1 \rightarrow M_n$ means that there is a sequence of transition $t_1, t_2, \ldots, t_{n-1}$ *for* $1 <= i < n$
  - A state M′ is reachable from a state M if and only if M → M′

Pre-set and Post-set - Transitions

A place p is an input place for trantition t iff
$(p, t) \in F$
We let $\bullet t$ denote the set of input places of t.
(pre-set of t)

A place p is an output place for trantition t iff
$(t, p) \in F$
We let $t\bullet$ denote the set of output places of t.
(post-set of t)

Pre-set and Post-set - Place

Analogously, we let

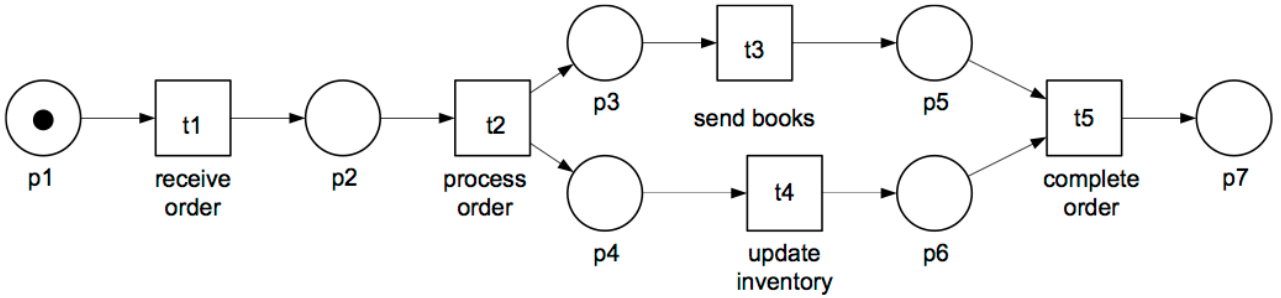
$\bullet p$ denote the set of transitions that share p as output place

$p\bullet$ denote the set of transitions that share p as input place

Formally:

$$\bullet x = \{\, y \mid (y, x) \in F \,\}$$
$$x\bullet = \{\, y \mid (x, y) \in F \,\}$$

Sample petri net – single instance



M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

GIVE THE FORMAL DEFINITION OF THE PETRI
in term of set T, P, F, pre-set e post-set!!!

Example

A Petri Net is a tuple (P, T, F, M_0) where

- P is a finite set of places = (p1, p2, p3 …. p7)
- T is a finite set of transitions = (t1, …. t5)
- F $\subseteq$ (P X T) U (T X U) is a flow relation F= ((p1,t1) (t1, p2) (p2, t2) …)
- M : P → N is the initial marking M = (1, 0, 0 …. 0)

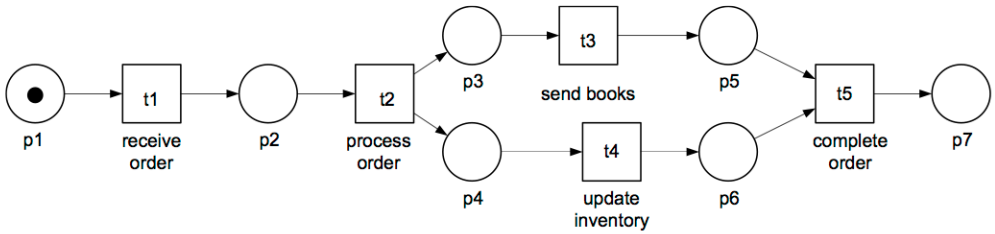
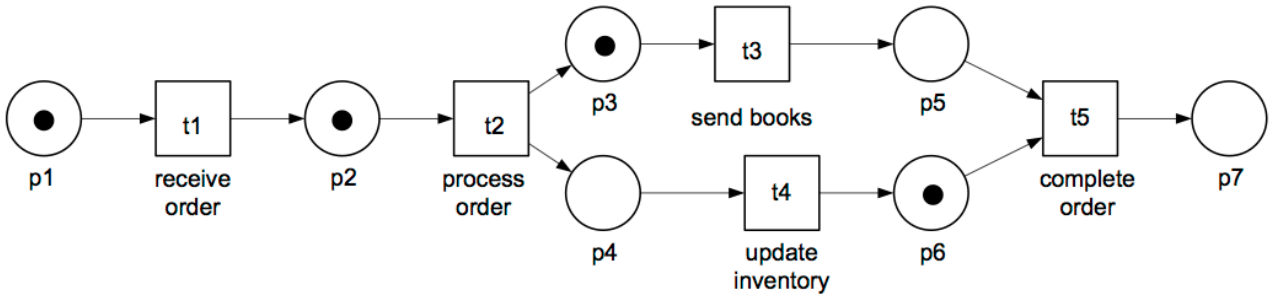


Fig. 4.26. Sample Petri net representing single process instance

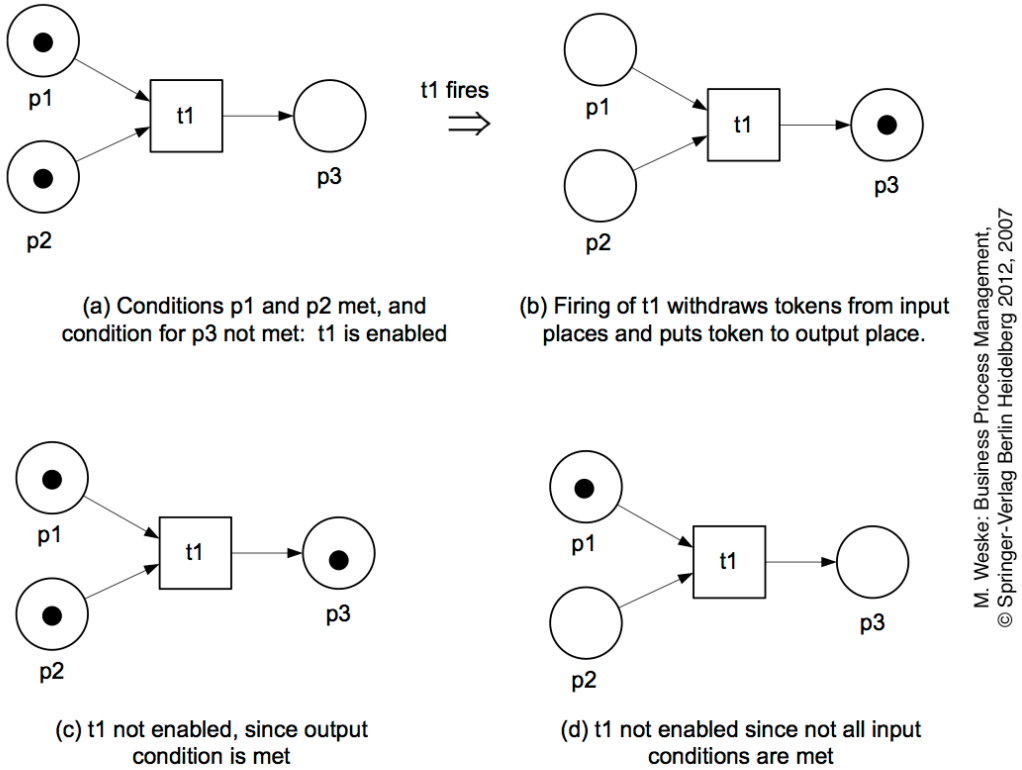
Sample petri net – multiple instance



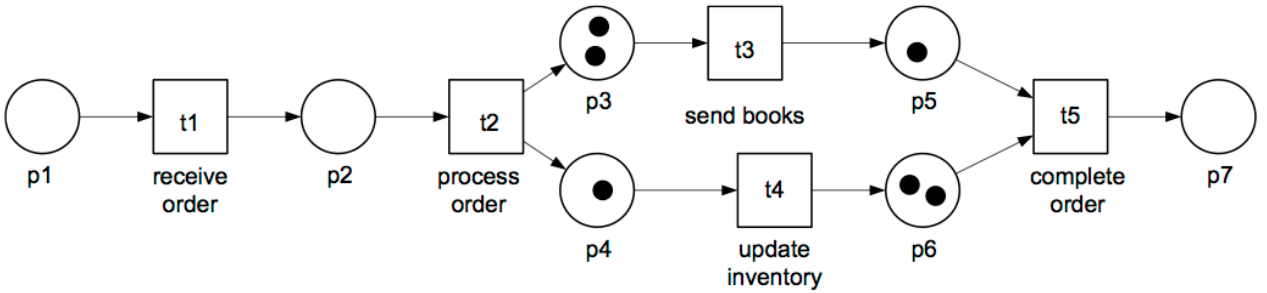
M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

What are the instance active?
Give the marking for earch active instance!

Safe-petri net or 1-bounded



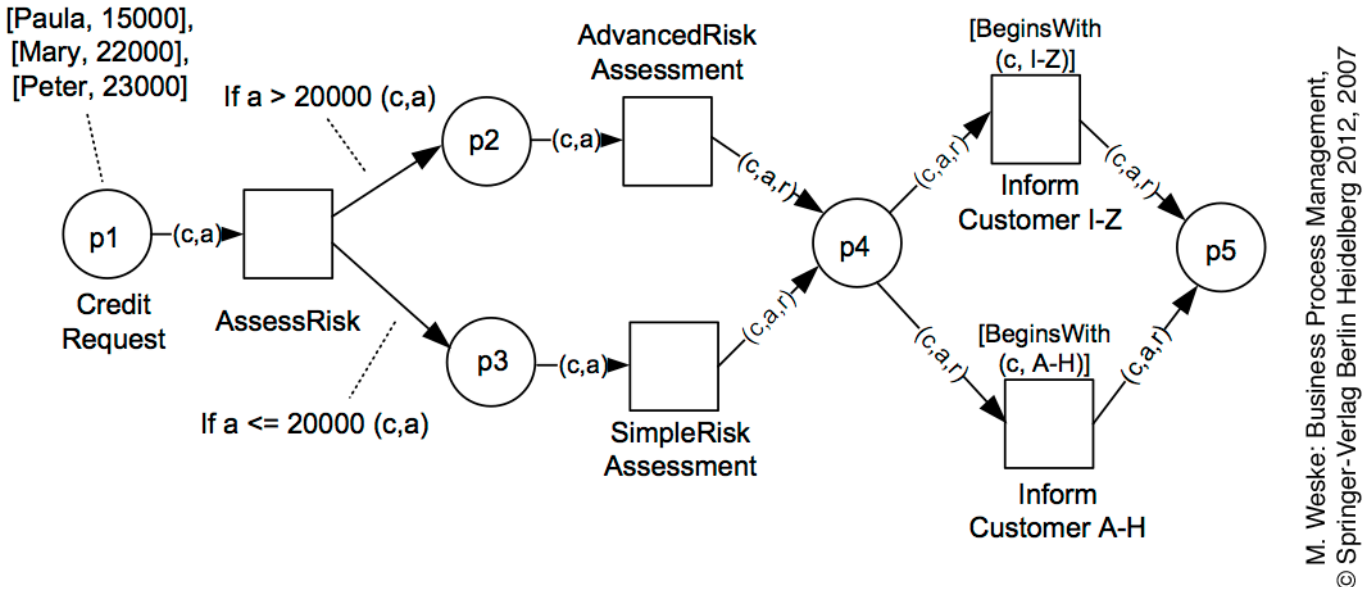
Sample Petri net – multiple instance



M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

GIVE THE MARKING FOR EACH INSTANCE!!!!

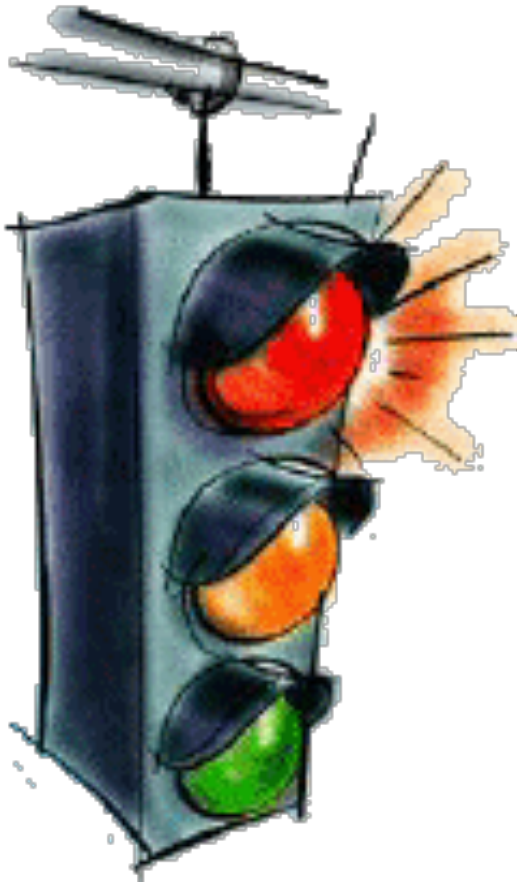
Colured Petri Net



- In the context of colured petri net tokens are typed so that difference instances are always addressed

Exercise

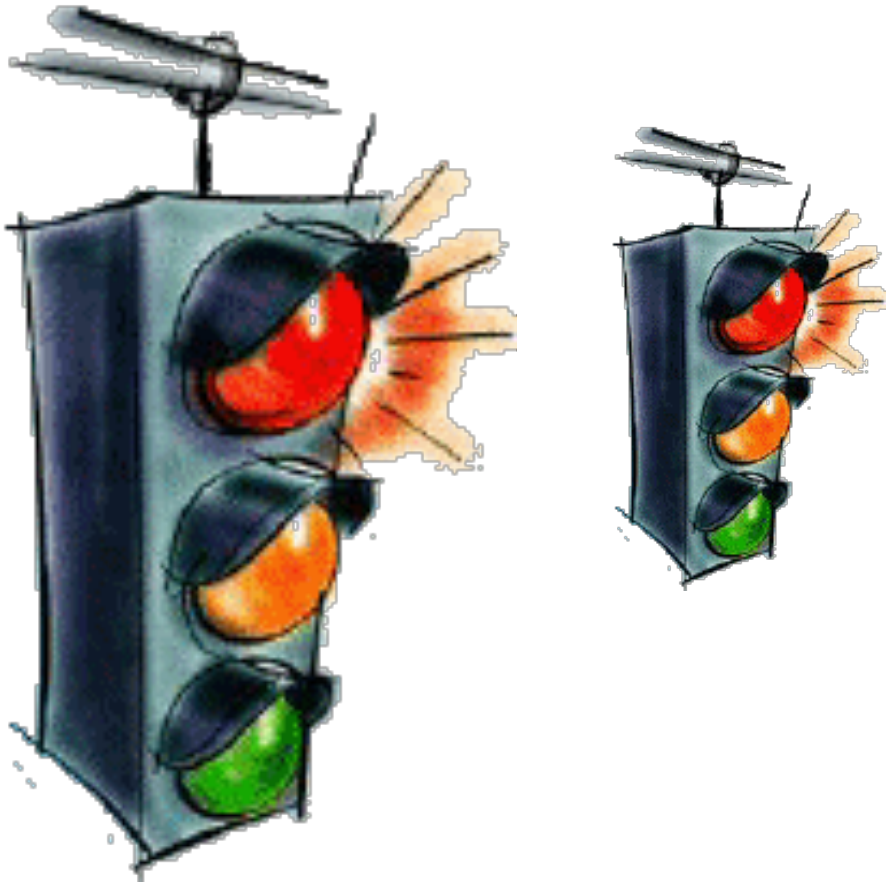
Modelling using petri net the traffic light scenario



Exercise

Modelling using petri net two traffic light

Complete the net in such a way that the two lights can never be green at the same time

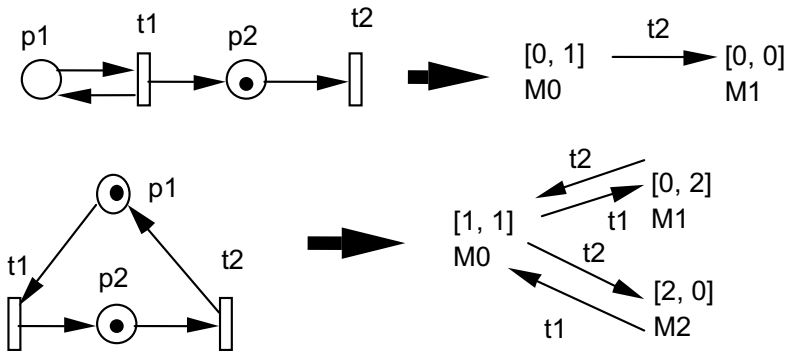


Reachability tree

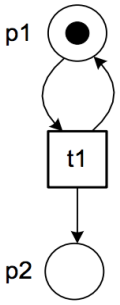
Definition: The reachability tree, also called marking graph, of a Petri net (N, Mo) is a graph in which

- nodes corresponds to reachable markings
- arcs correpond to feasible transitions

Remark: the reachability tree ca be infinite



Coverability Tree (from Murata)



If, on the path from the root to M, there exists a marking M'' such that M'(p) ≥ M''(p) for each p and M'≠M'' (i.e., M'' is coverable), then replace M'(p) by ω for each p such that M'(p) > M''(p)

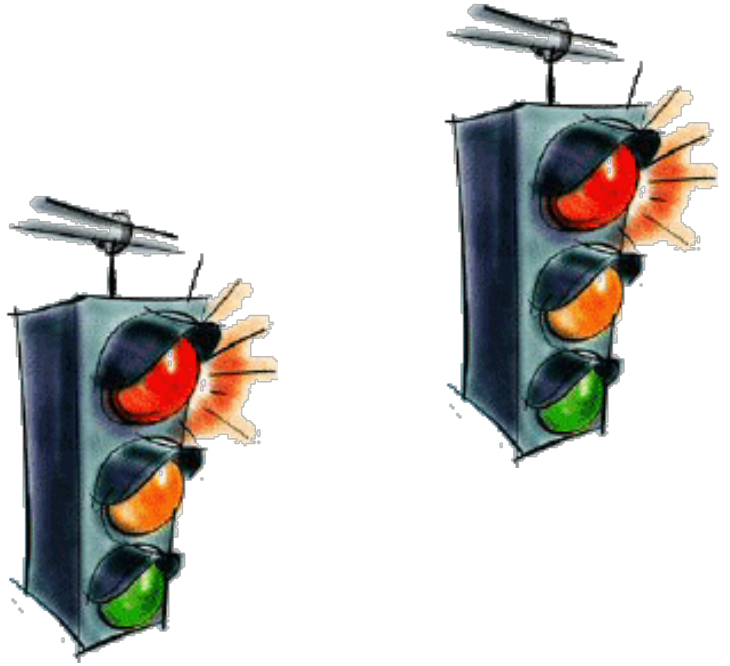
Mo [1,0] ->t1-> M1[1,1]

Mo [1,0] ->t1-> M1[1, ω]

Mo [1,0] ->t1-> M1[1, ω] t1

Exercise

- Draw the reachability graph of the net



Mapping from BPMN to Petri Nets

Mapping a Need for Verification

- Business Processes are typically defined by business experts which ask for graphical and user-friendly notations
- Most notations such as BPMN 2.0 used typically lack precisely defined semantics limiting the possibility of analysis
- In order to formally analysis Business Process we need to derive a formal model of the Business Process being verified
  - From BPMN 2.0 to Petri Net
  - From BPMN 2.0 to Process Algebra
  - From … To …

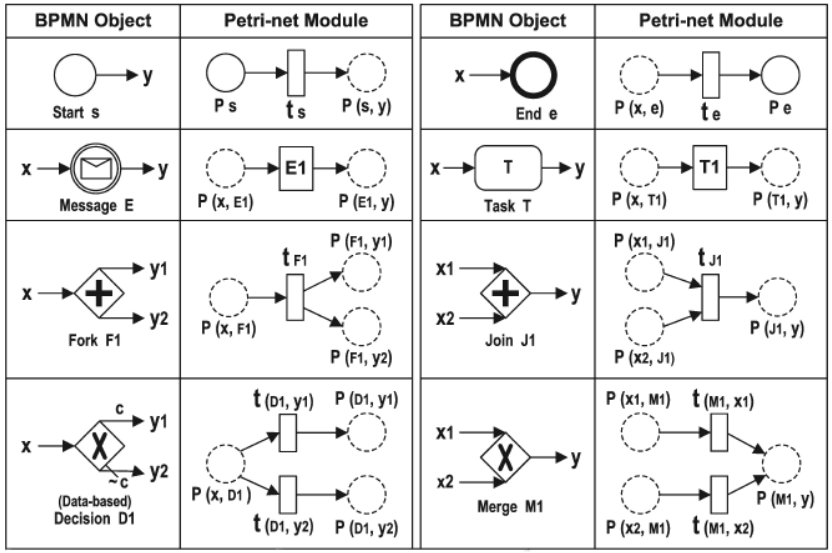
From BPMN 2.0 to Petri Net (I)

- Raedts et. al, 2007 - *Raedts, I., Petkovic, M., Usenko, Y. S., van der Werf, J. M., Groote, J. F., and Somers, L. J. (2007). Transformation of BPMN models for Behaviour Analysi, In MSVVEIS, pages 126-137.*
  - It converts BPMN 2.0 to Petri Nets extended with inhibitor and reset arcs
  - The proposed mapping is not comprehensive enough in order to clearly address all the BPMN 2.0 elements (i.e. messages, sub-process and loop activity are not included)

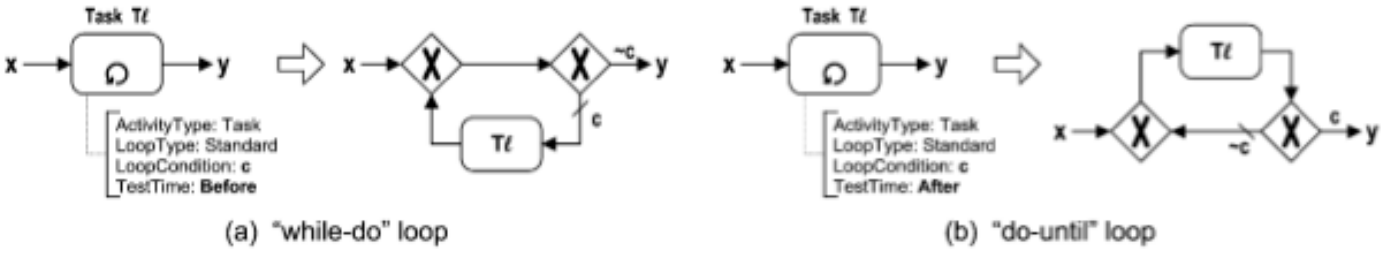
- Awad, 2010 - *Awad, A., Decker, G., and Lohmann, N. (2010). Diagnosing and repairing data anomalies in process models. BPM Workshops, vol. 43 of Lecture Notes in Business Information Processing, pp. 5-16. Springer.*
  - It formalizes the basic data object together with the more traditional control flow elements
  - It is missed a more general and complex mapping on the BPMN 2.0 elements such as sub-process, messages, etc.

From BPMN 2.0 to Petri Net (II)

- Dijkman, R. M., Dumas, M., and Ouyang, C. (2008). Semantics and analysis of business process models in BPMN. Inf. Softw. Technol., 50(12):1281-1294.
  - Up to our review it the most complete methodology to transform BPMN 2.0 models into Petri
  - It starts from the basic elements and introduces some more details regarding multiple instances, sub-process and exception handling
  - It can be easy extended to the whole specification



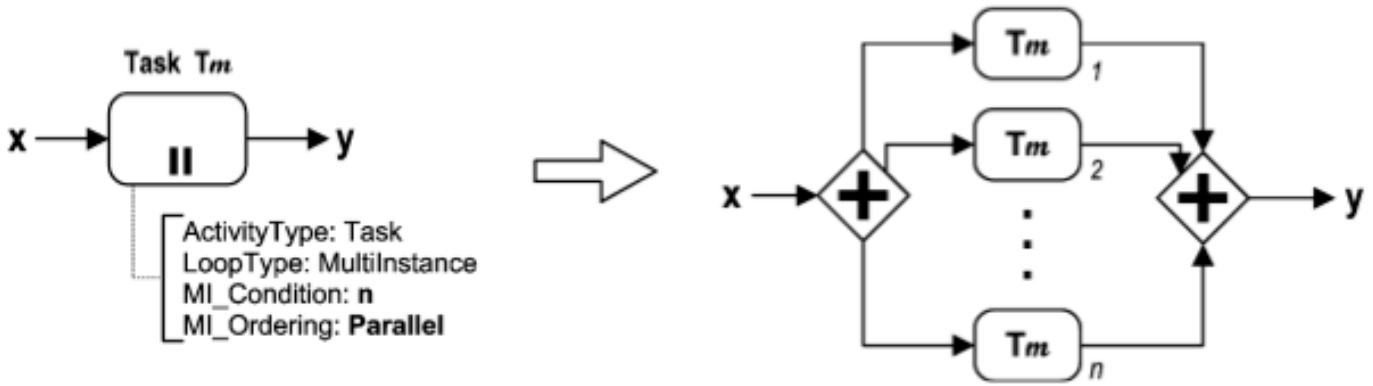
Macro expansions for repeated activities



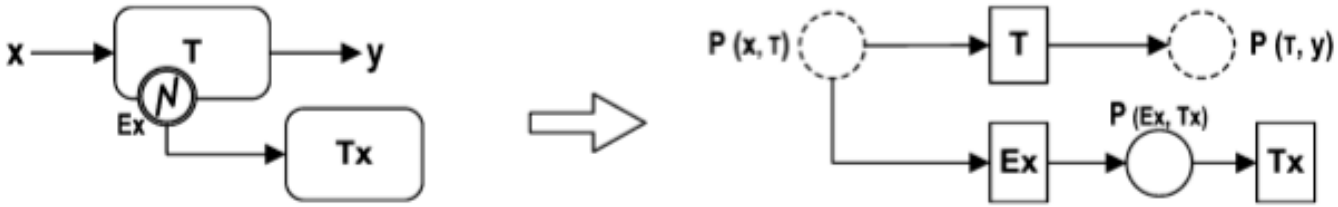
(a) "while-do" loop

(b) "do-until" loop

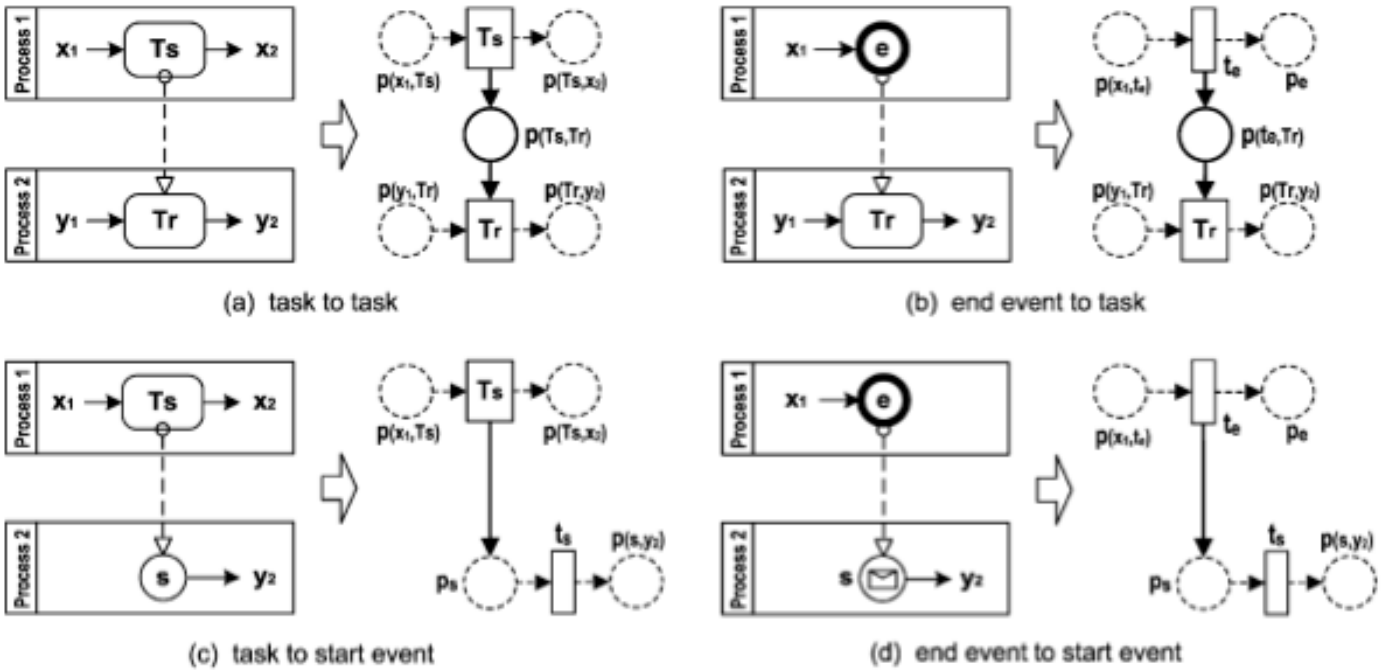
Macro expansion for a multi-instance activity where n is known at design time



Mapping of a task with an exception flow



Mapping of message flows between BPMN processes



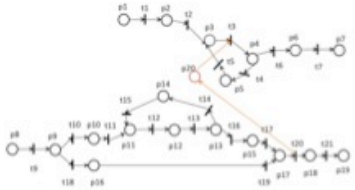
(a) task to task

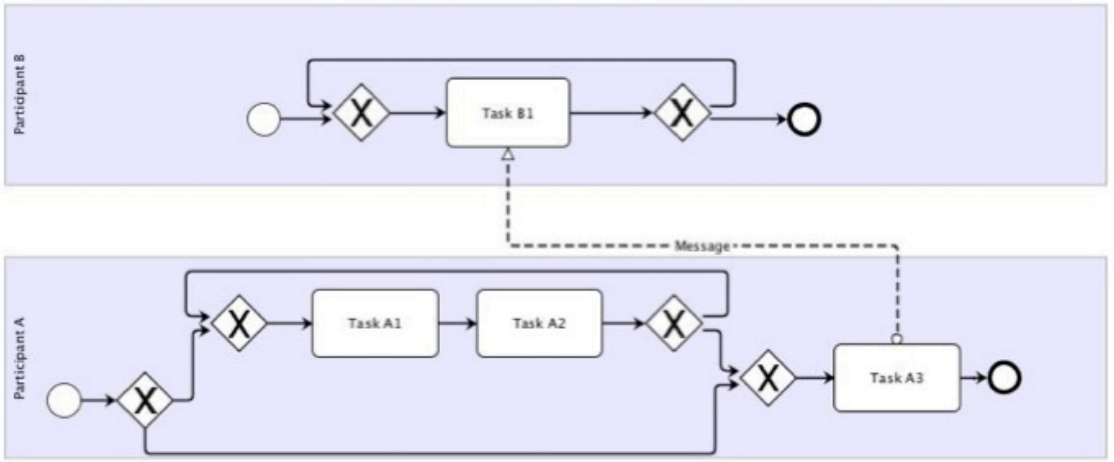
(b) end event to task

(c) task to start event

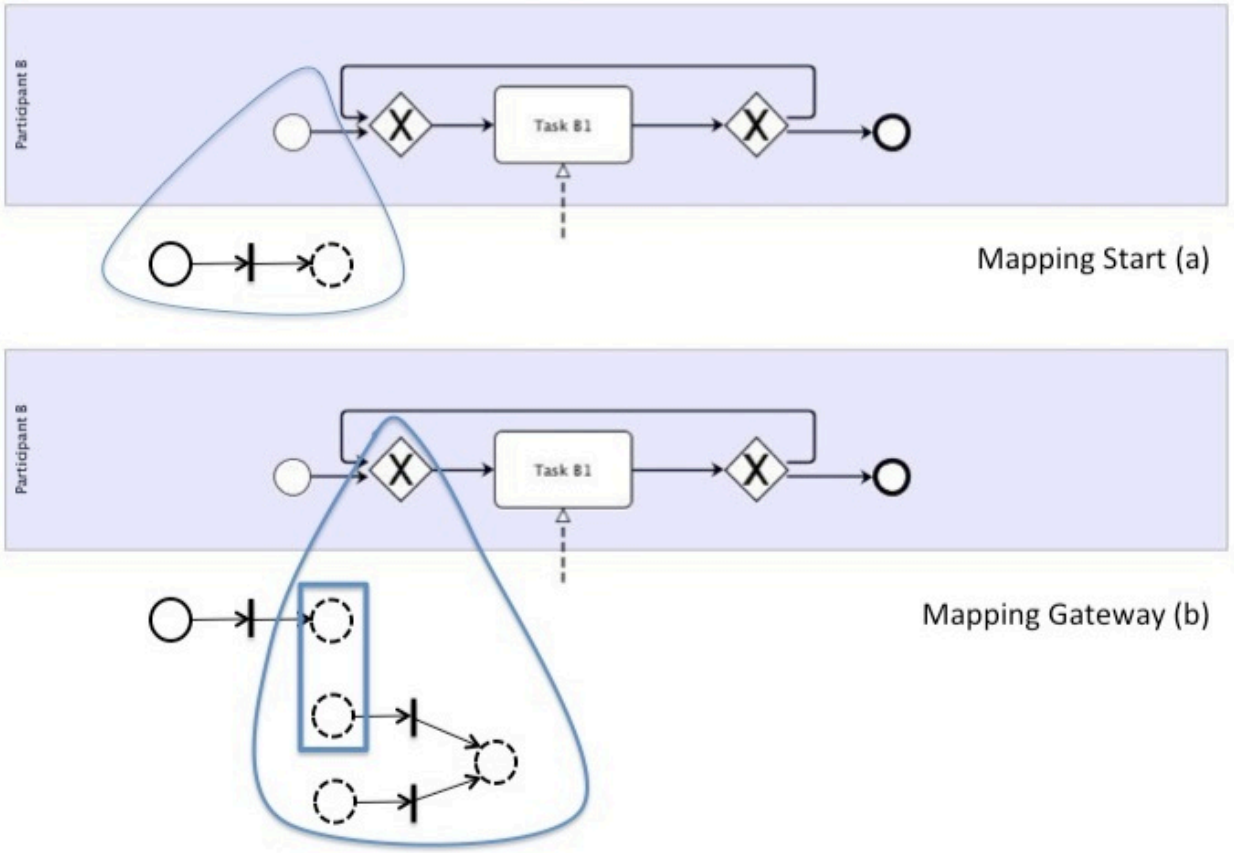
(d) end event to start event

Basic Example of Mapping BP into Petri Net

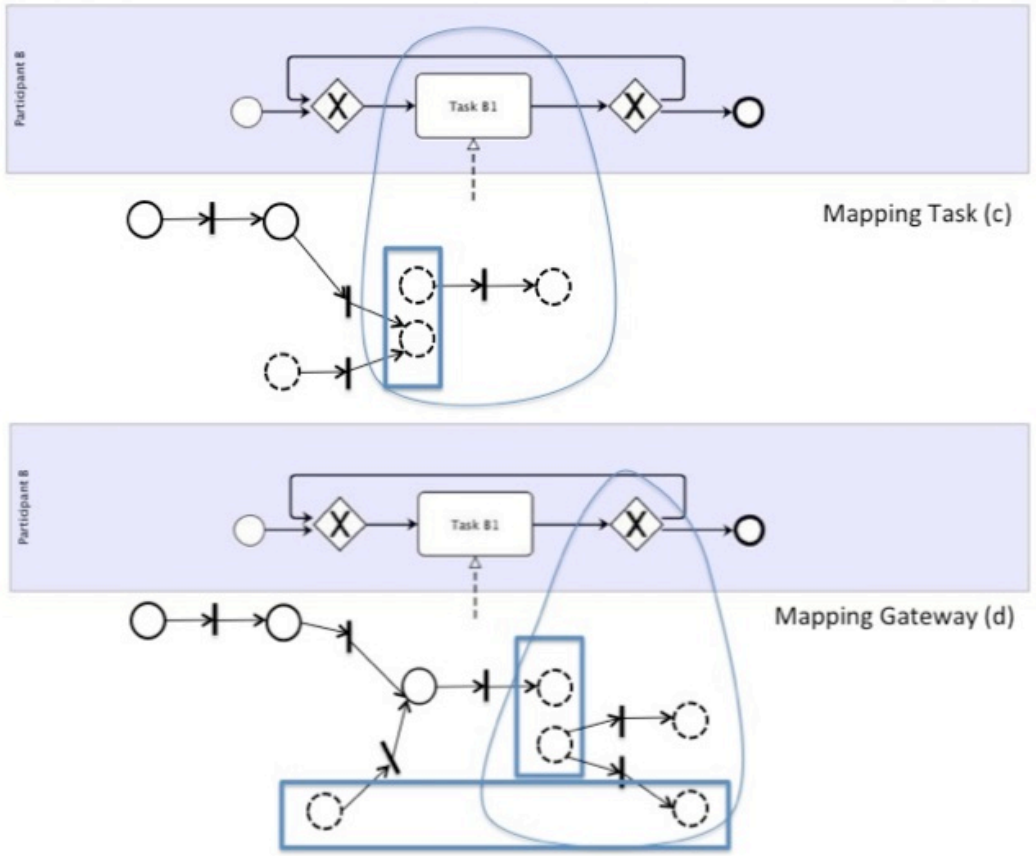




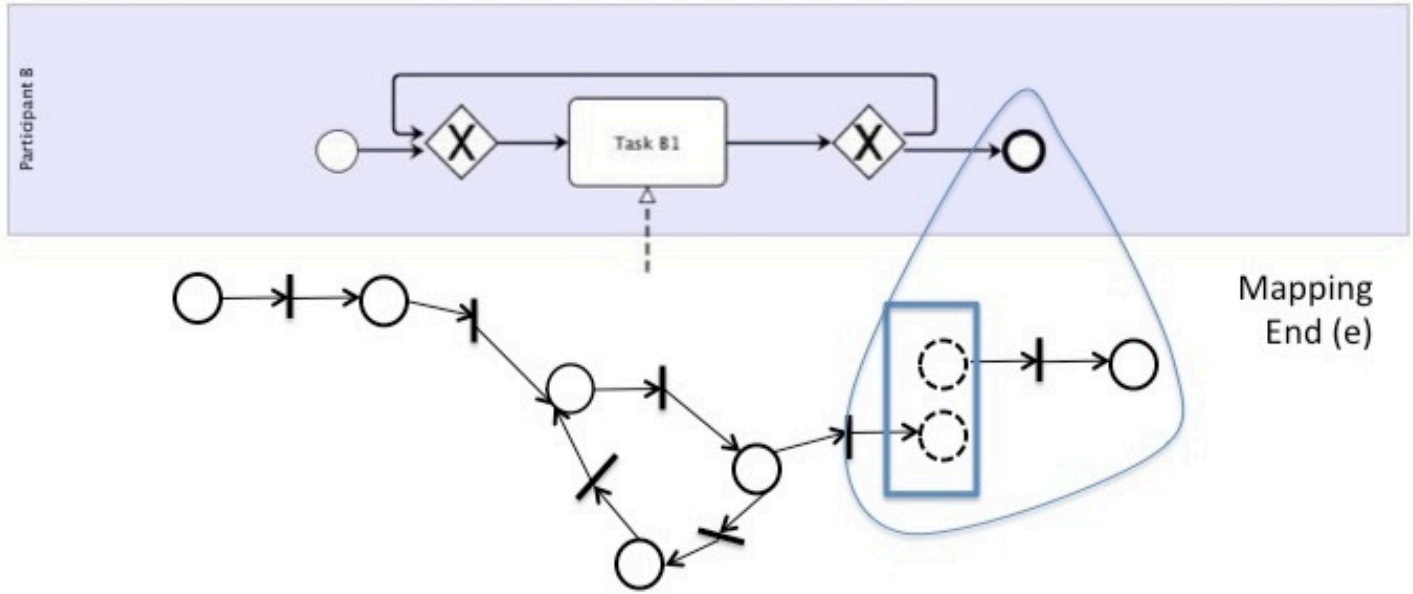
Basic Example of Mapping BP into Petri Net (step 1 participant B)



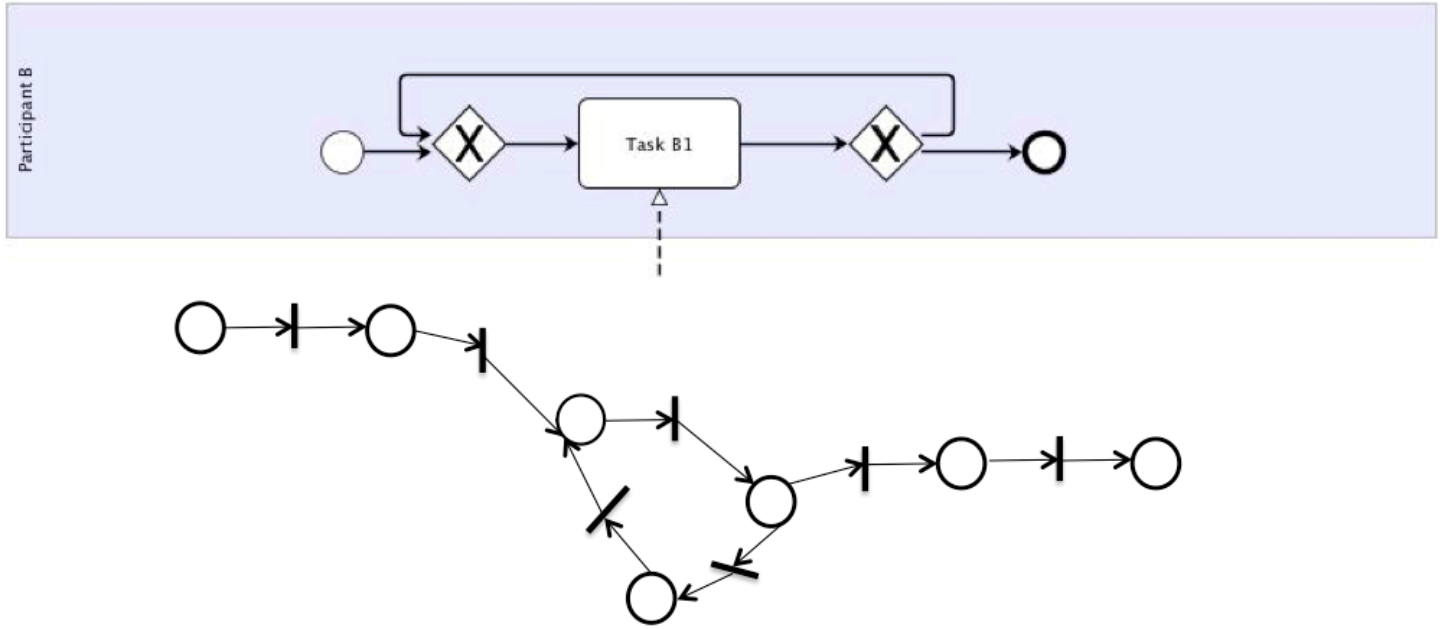
Basic Example of Mapping BP into Petri Net (step 2 participant B)



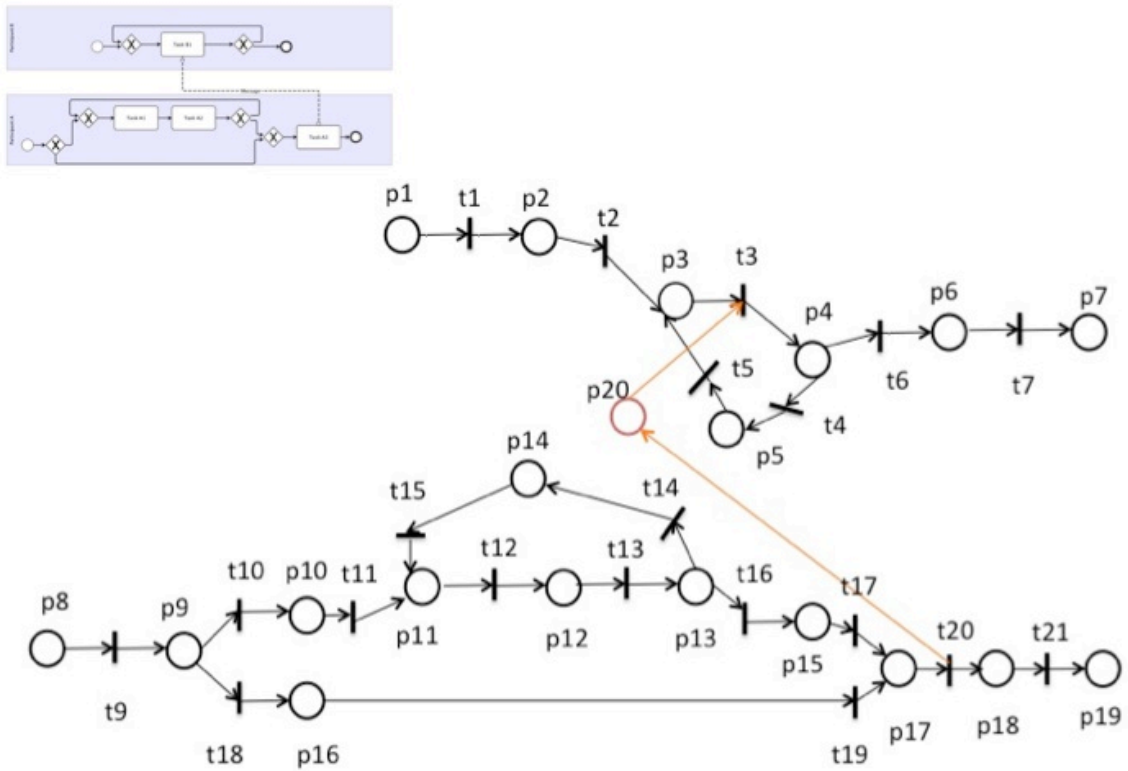
Basic Example of Mapping BP into Petri Net (step 3 participant B)



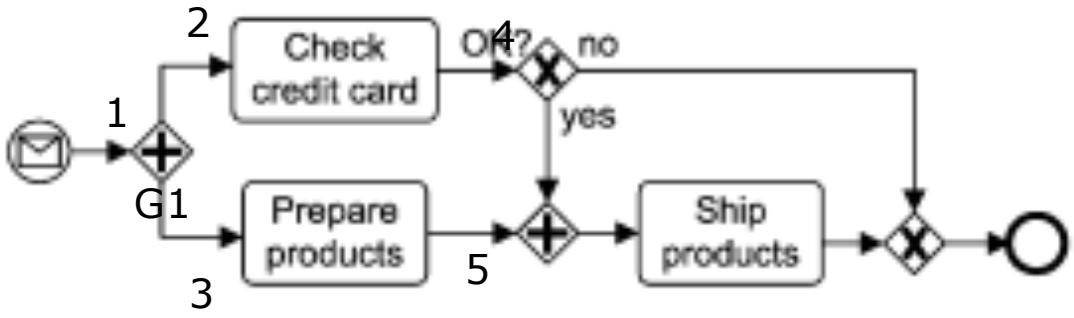
Basic Example of Mapping BP into Petri Net (participant B)

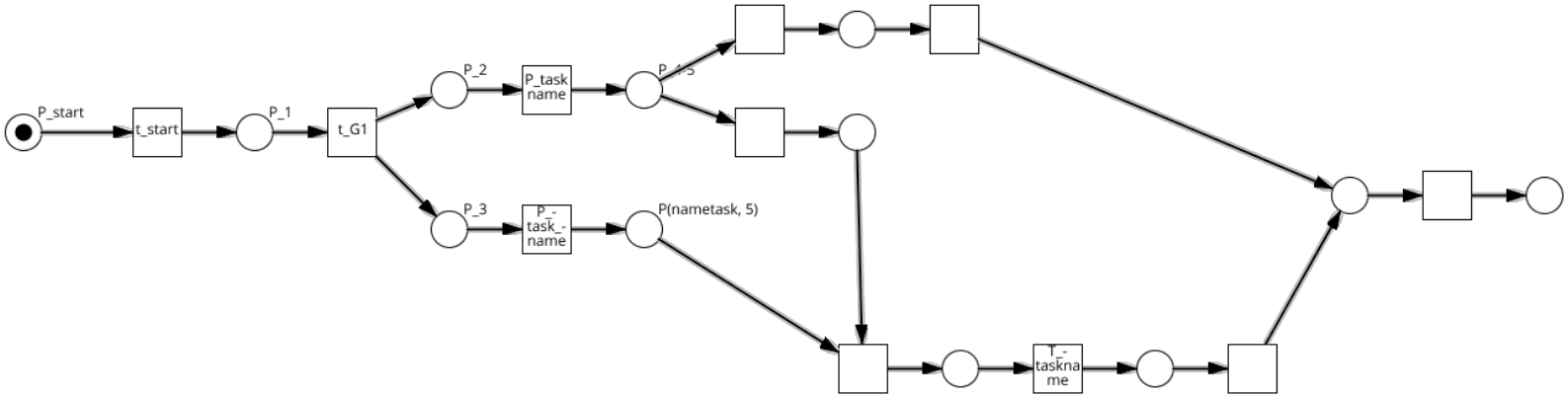


Basic Example of Mapping BP into Petri Net

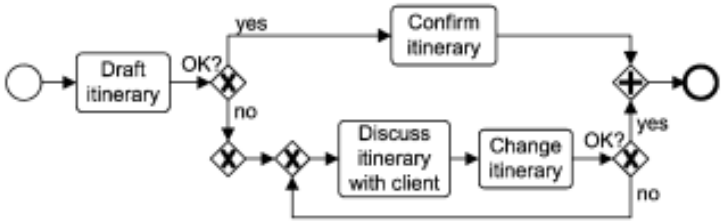


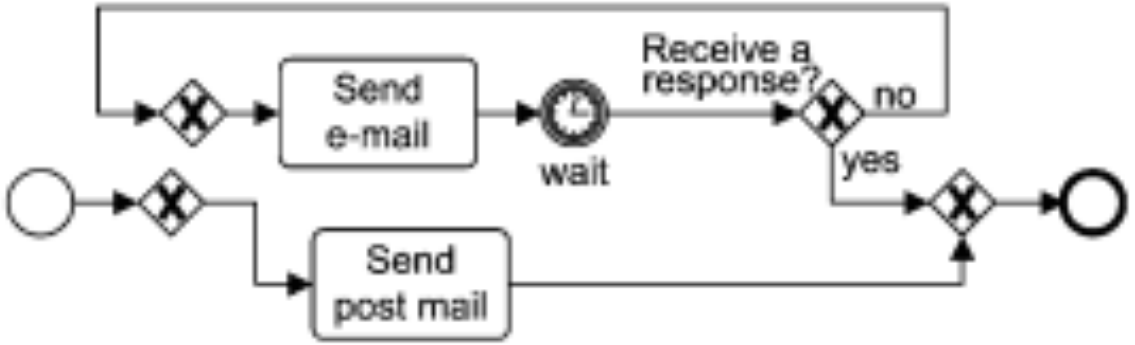
Mapping into Practice





Mapping into Practice





Woped Based Verification

http://woped.dhbw-karlsruhe.de/woped/