Business Processes Modelling

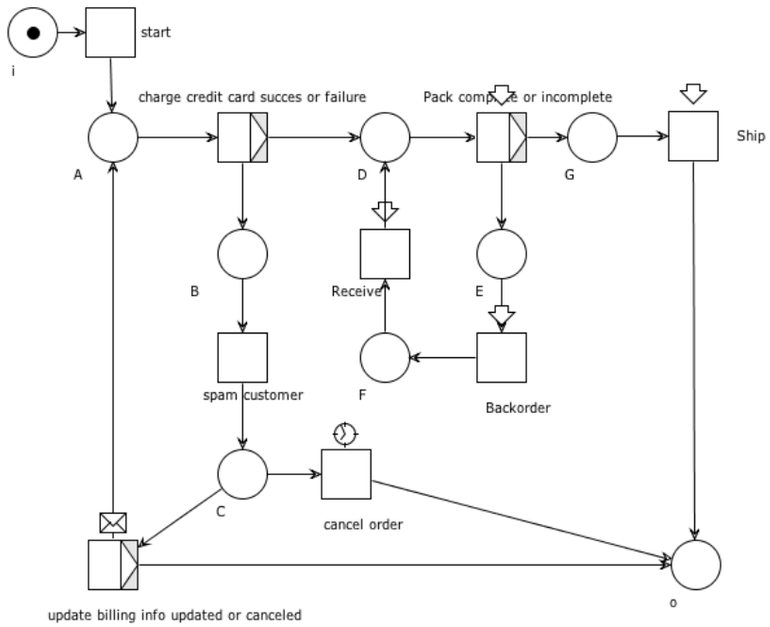
MPB (6 cfu, 295AA)

Roberto Bruni

http://www.di.unipi.it/~bruni

04 - Petri nets basics

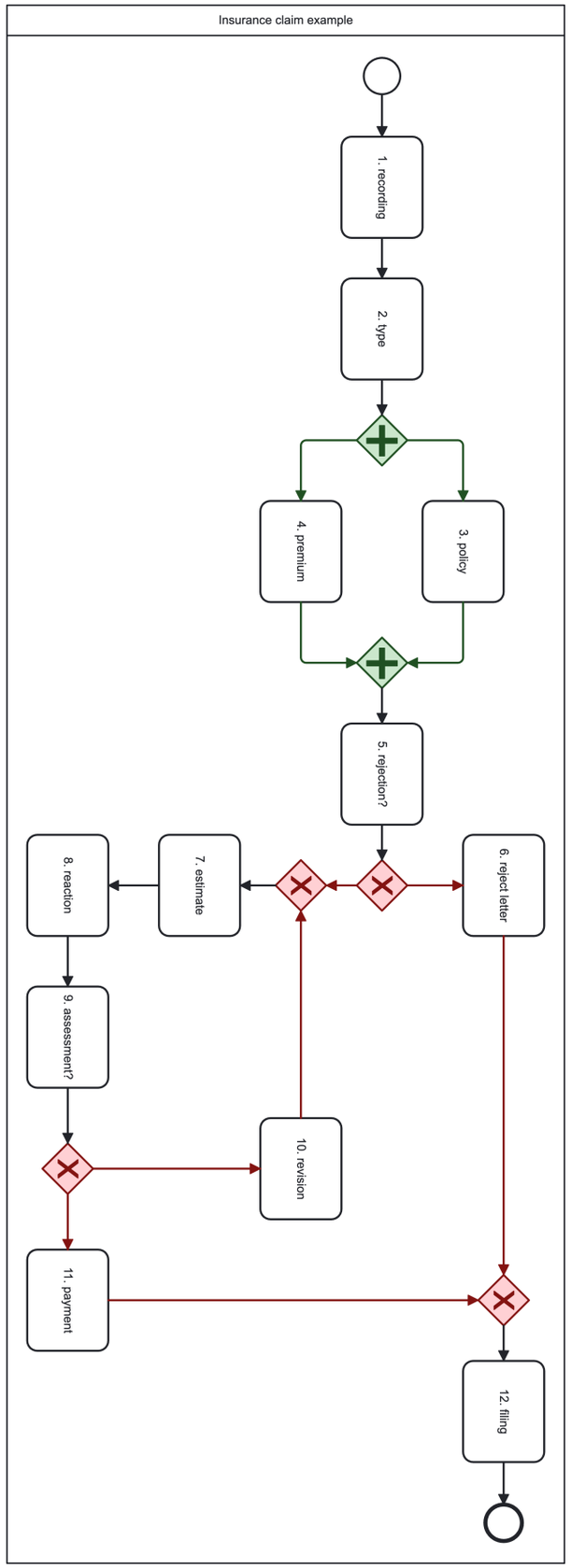
Object



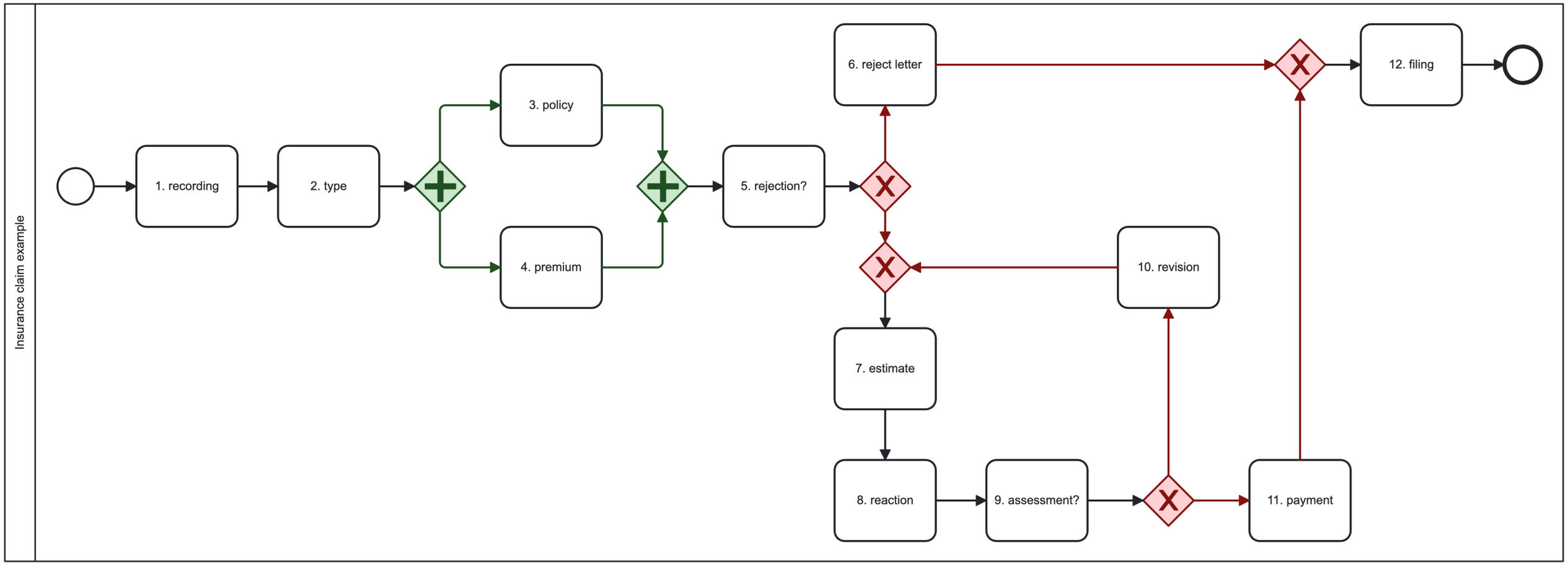
Informal introduction to Petri nets and workflow nets

Ch.4.4 of Business Process Management: Concepts, Languages, Architectures

An important difference



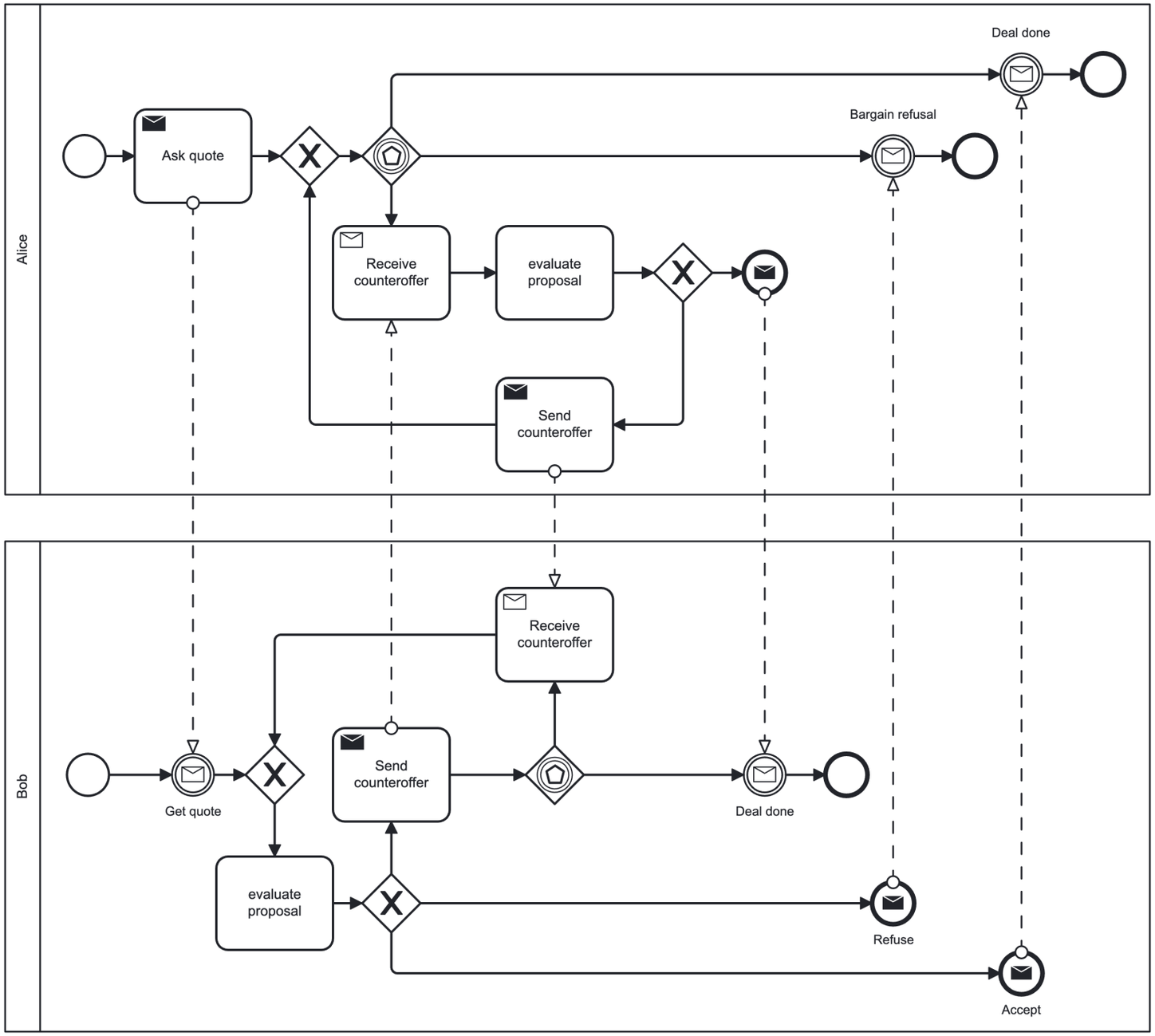
Current state of the claim?



https://camunda.com/download/modeler/

https://bpmn.io/

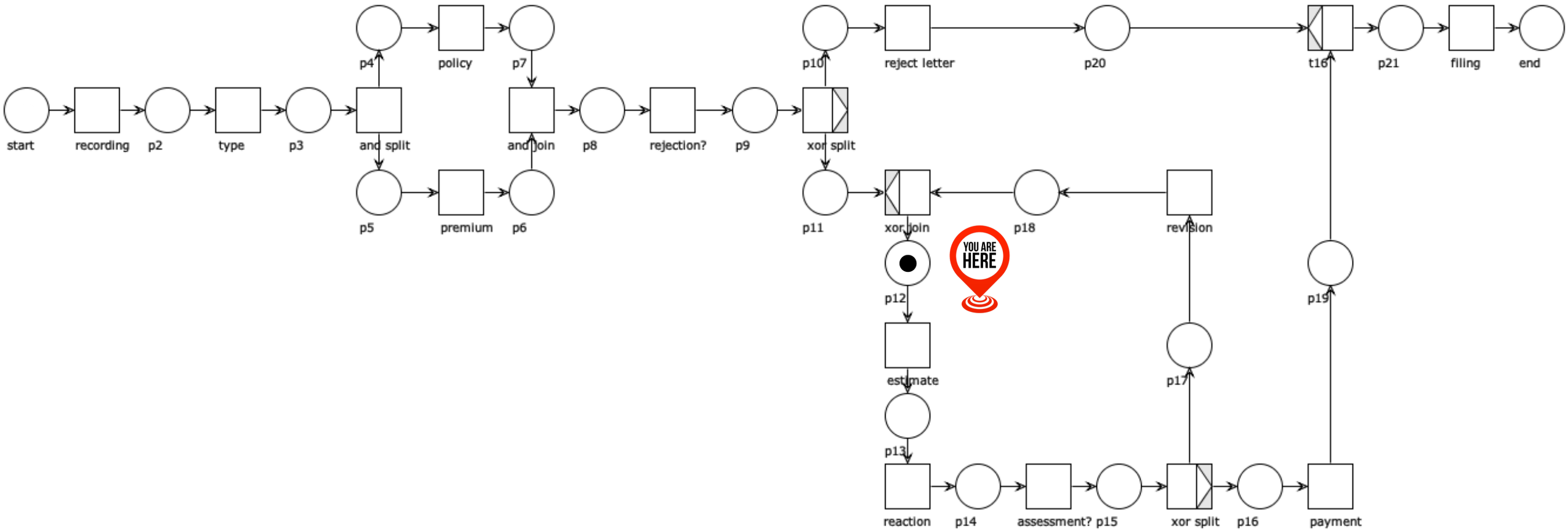
Current state of the bargain?



https://camunda.com/download/modeler/

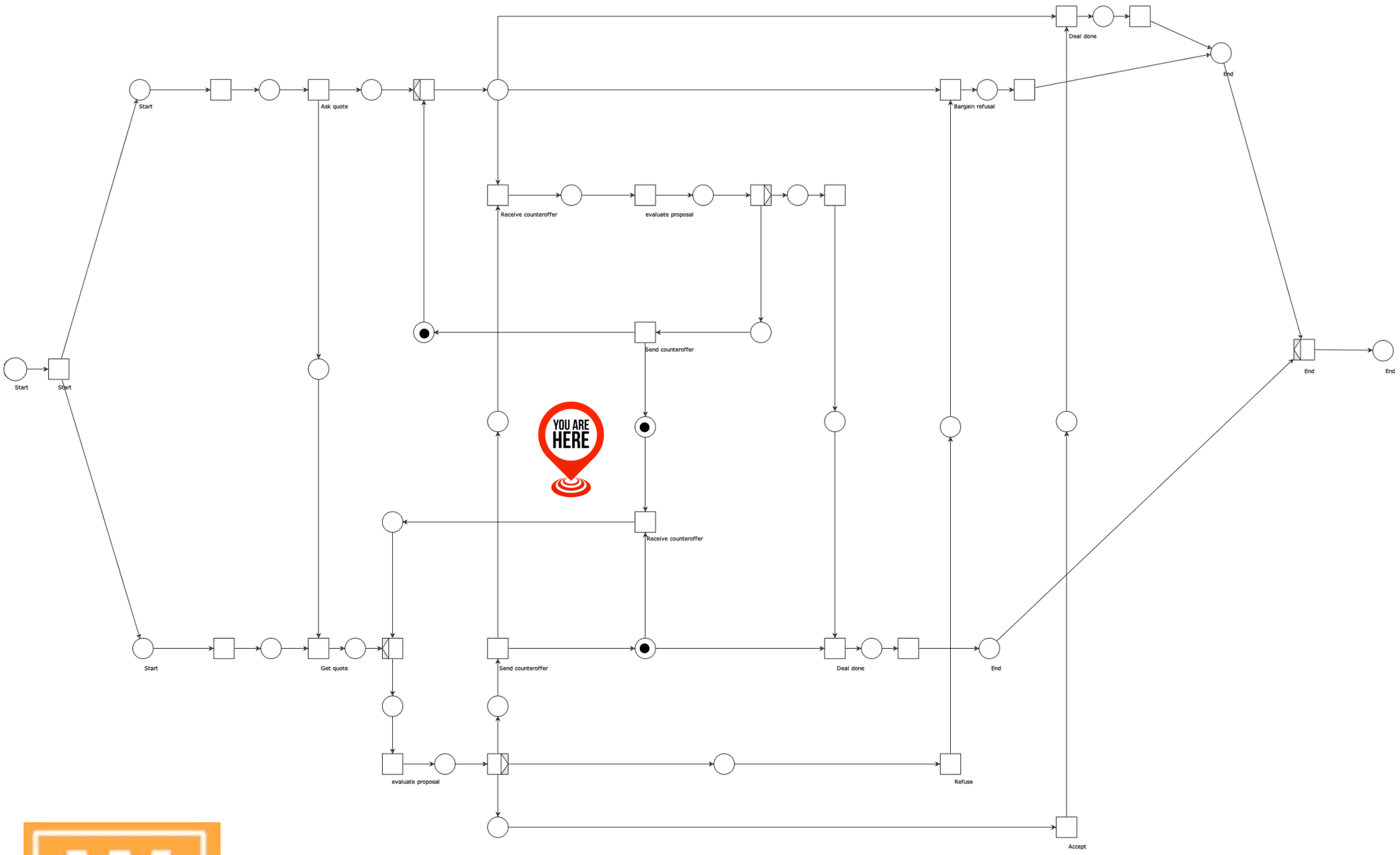
https://bpmn.io/

Disambiguation (WfN)



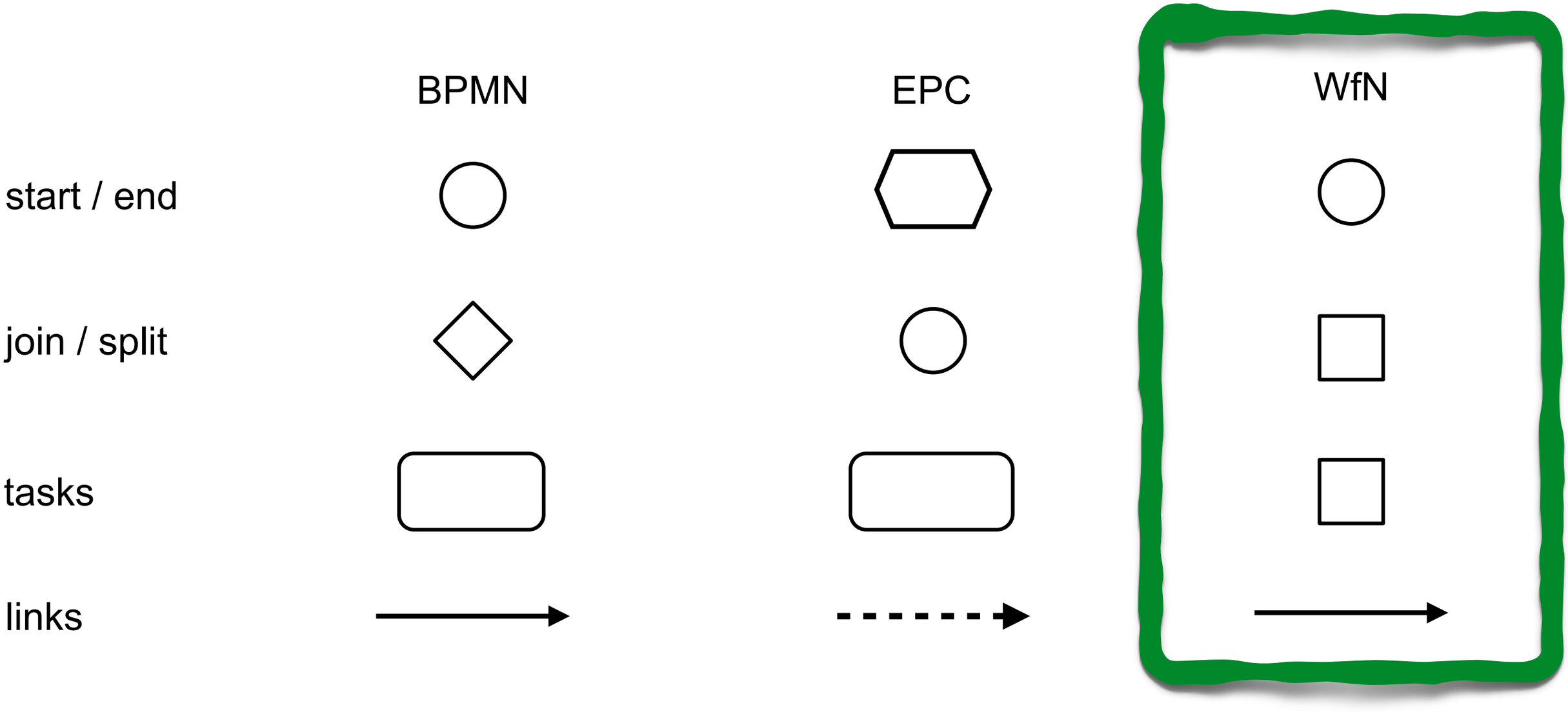
http://woped.dhbw-karlsruhe.de/

Disambiguation (WfN)



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

A sneak peek at standards BPMN, EPC, Workflow nets

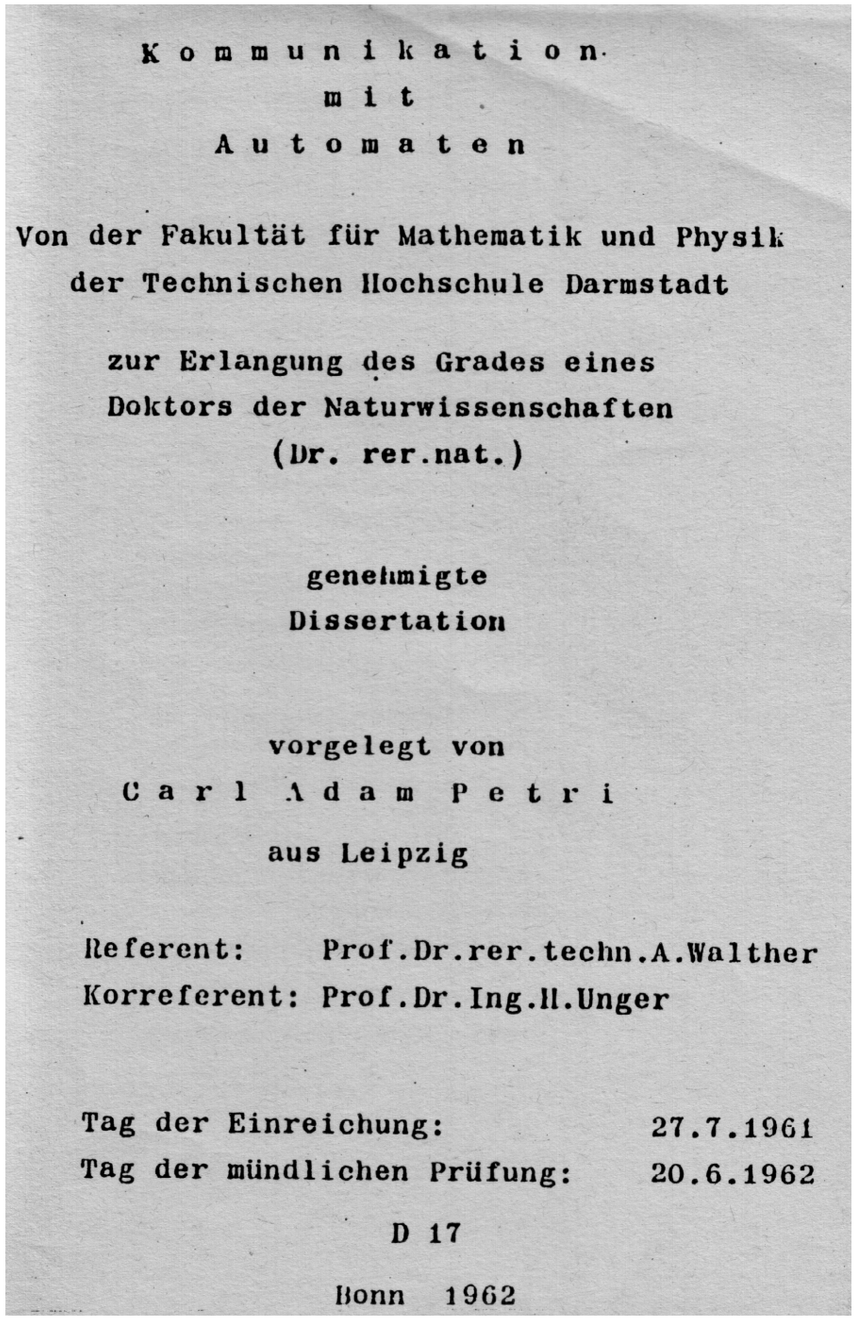


Petri nets:
basic definitions

Carl Adam Petri

July 12, 1926 - July 2, 2010

http://www.informatik.uni-hamburg.de/TGI/mitarbeiter/profs/petri_eng.html



Kommunikation
mit
Automaten

Von der Fakultät für Mathematik und Physik
der Technischen Hochschule Darmstadt

zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer.nat.)

genehmigte
Dissertation

vorgelegt von
Carl Adam Petri

aus Leipzig

Referent: Prof.Dr.rer.techn.A.Walther
Korreferent: Prof.Dr.Ing.H.Unger

Tag der Einreichung: 27.7.1961
Tag der mündlichen Prüfung: 20.6.1962

D 17

Bonn 1962

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

Petri nets for us

Formal and abstract business process specification

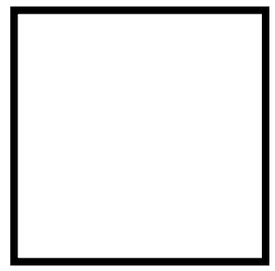
Formal: the semantics of process instances becomes well defined and not ambiguous

Abstract: execution environment is disregarded

(Remind about separation of concerns)

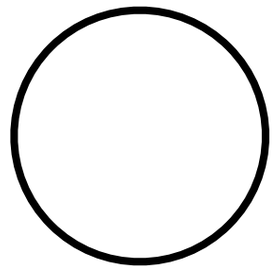
Transitions

A transition can stand for
an operation
a calculation
an evaluation
a transformation
a transportation
a task
an activity
a decision
...



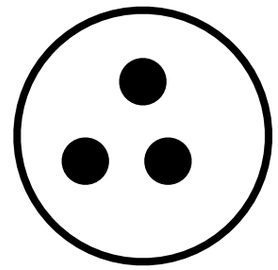
Places

A place can stand for
a state
a medium
a buffer
a condition
a repository of resources
a type
a memory location
...



Tokens (within places)

A token can stand for
a physical object
a piece of data
a record



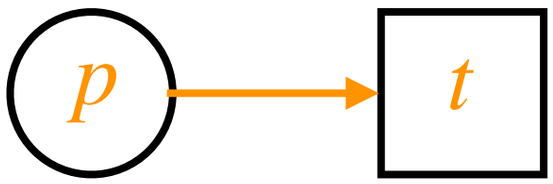
a resource
an activation mark
a message
a document
a case
a value
...

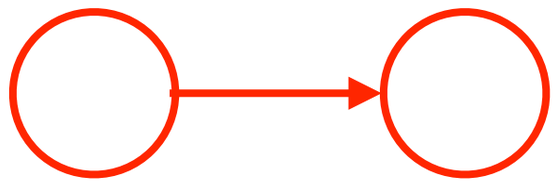
Arcs

An arc represents a dependency:

from a place p to a transition t

(a token from p is consumed to fire the transition t)

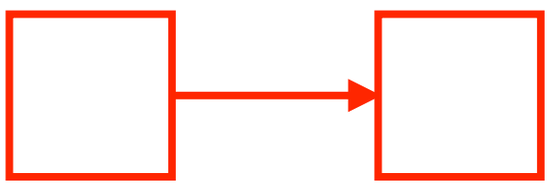




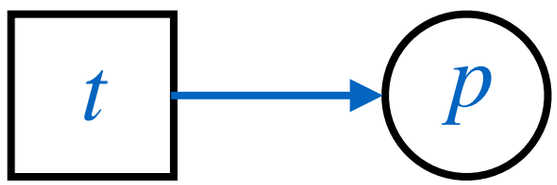
not admitted!

from a transition t to a place p

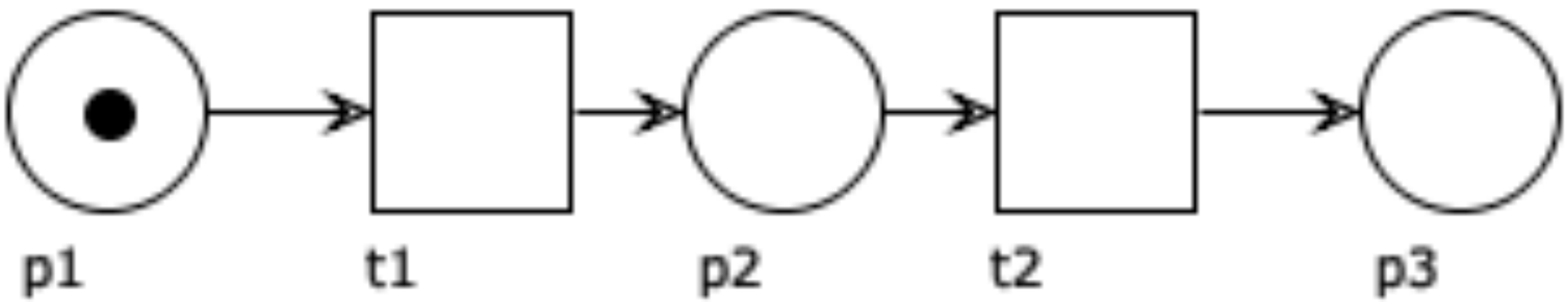
(the firing of the transition t produces a token in p)



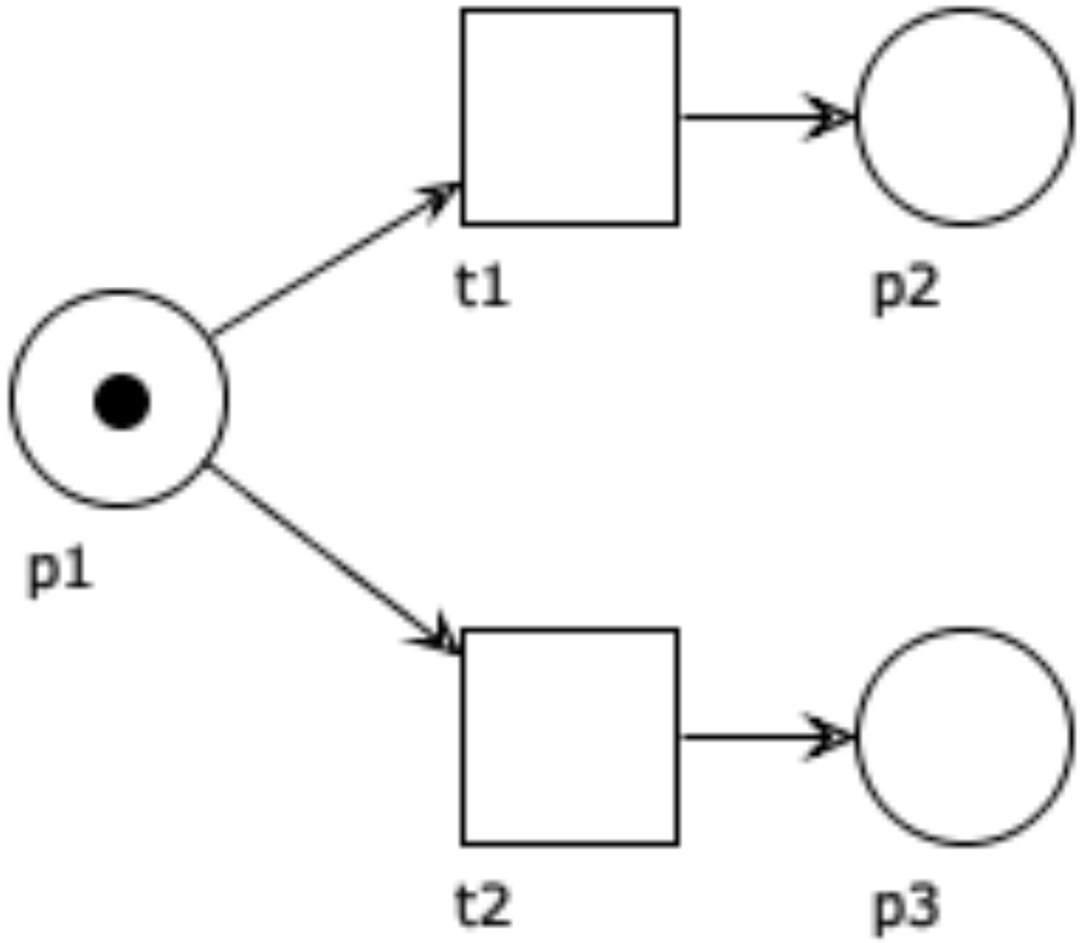
not admitted!



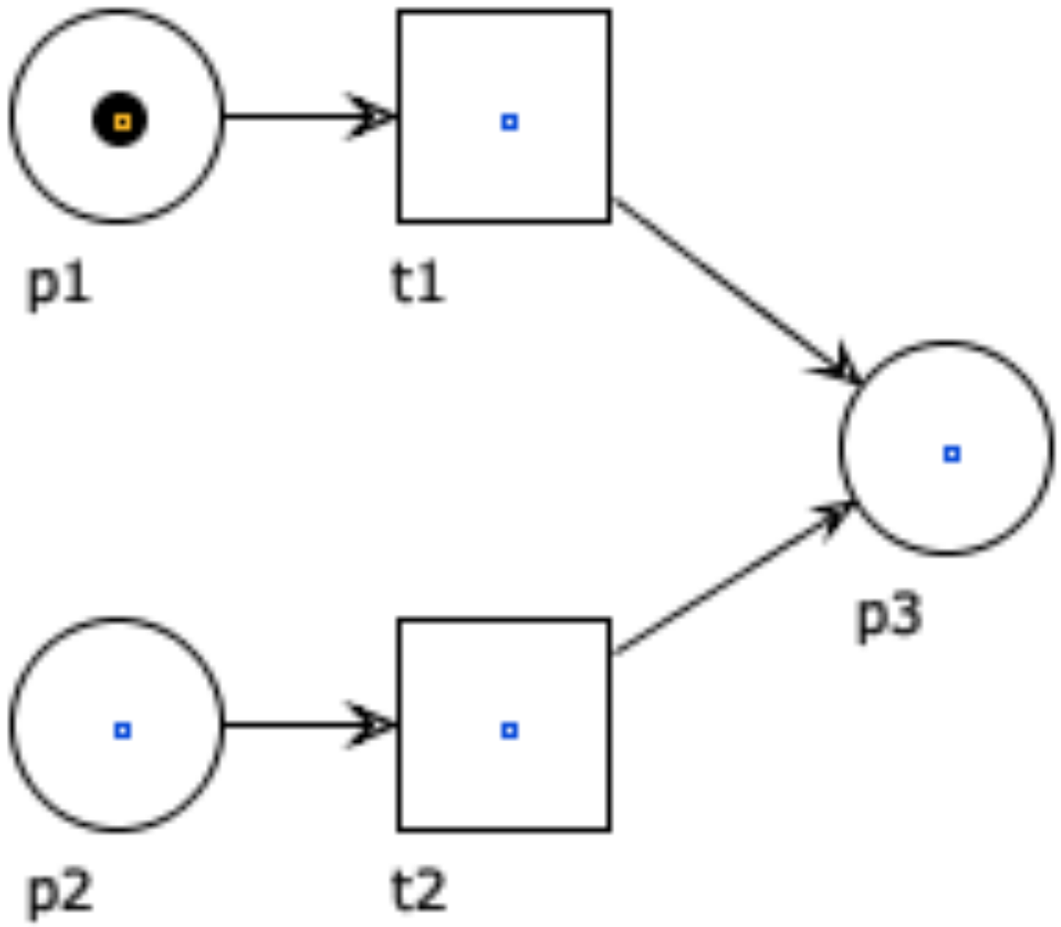
Example: sequence



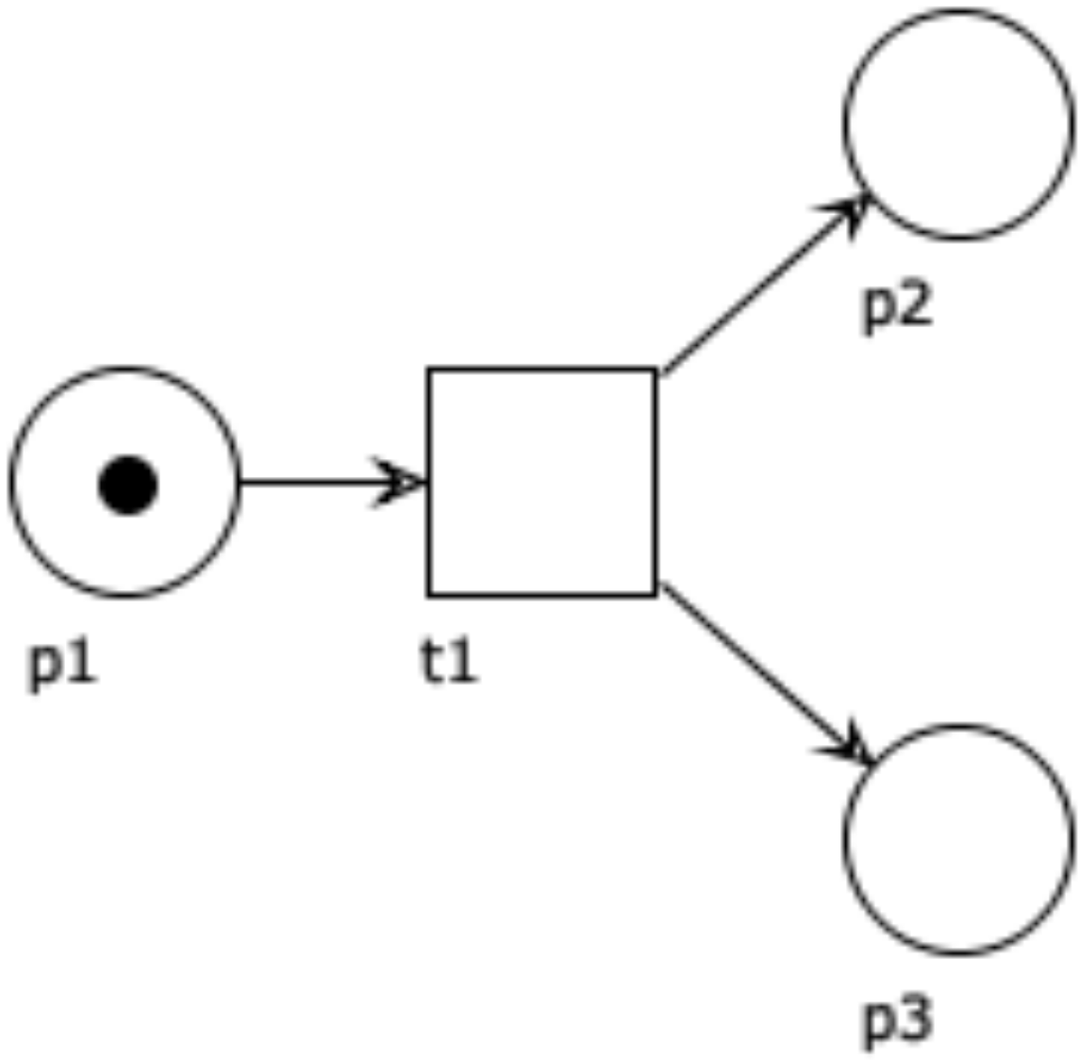
Example: XOR split



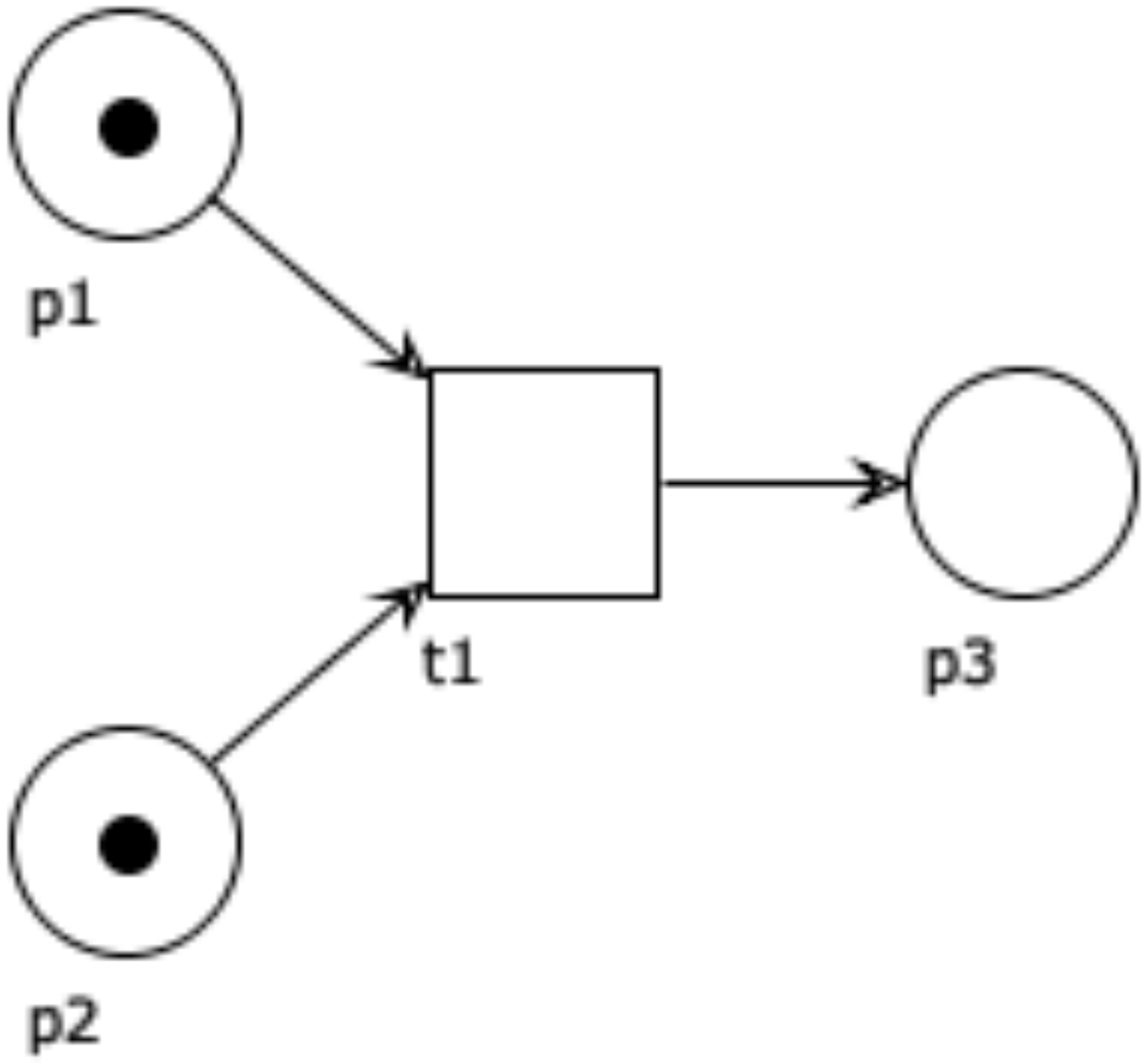
Example: XOR join



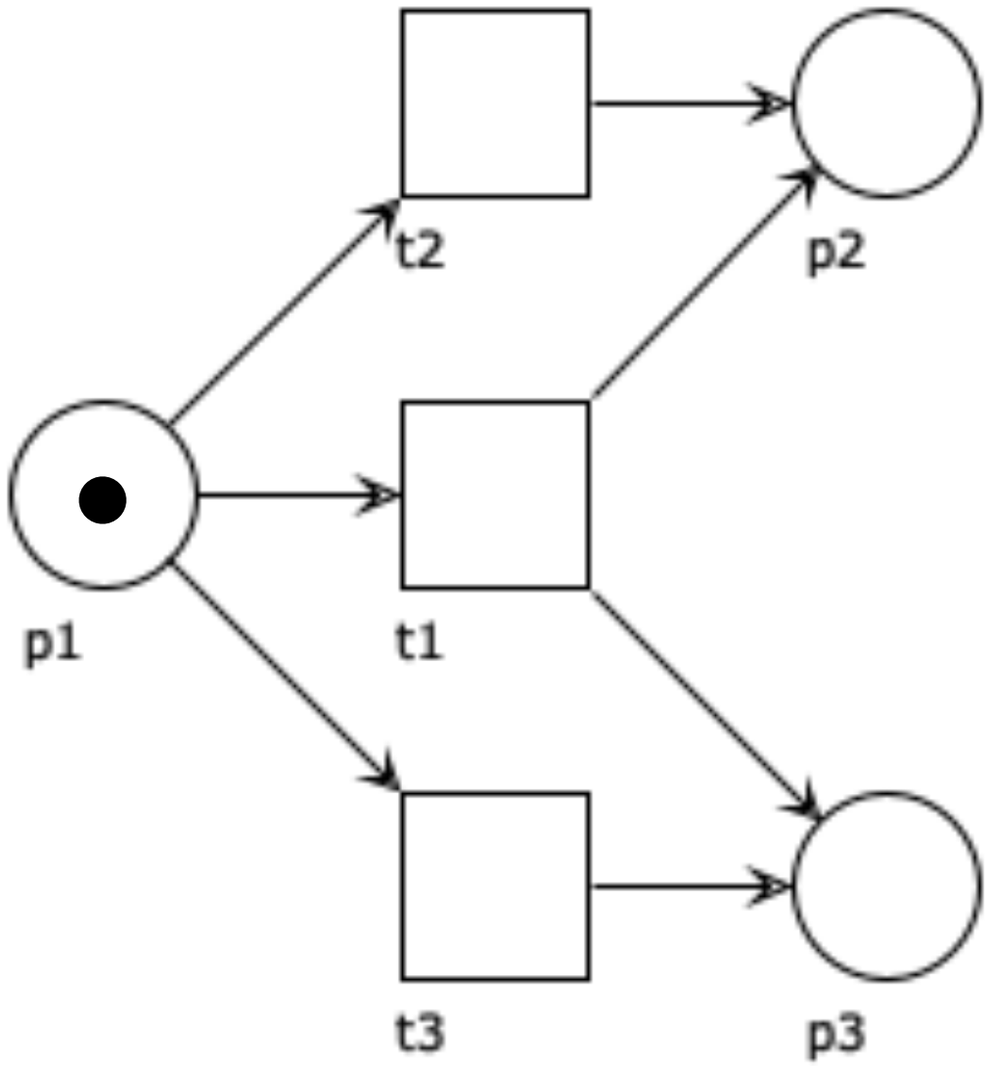
Example: AND split



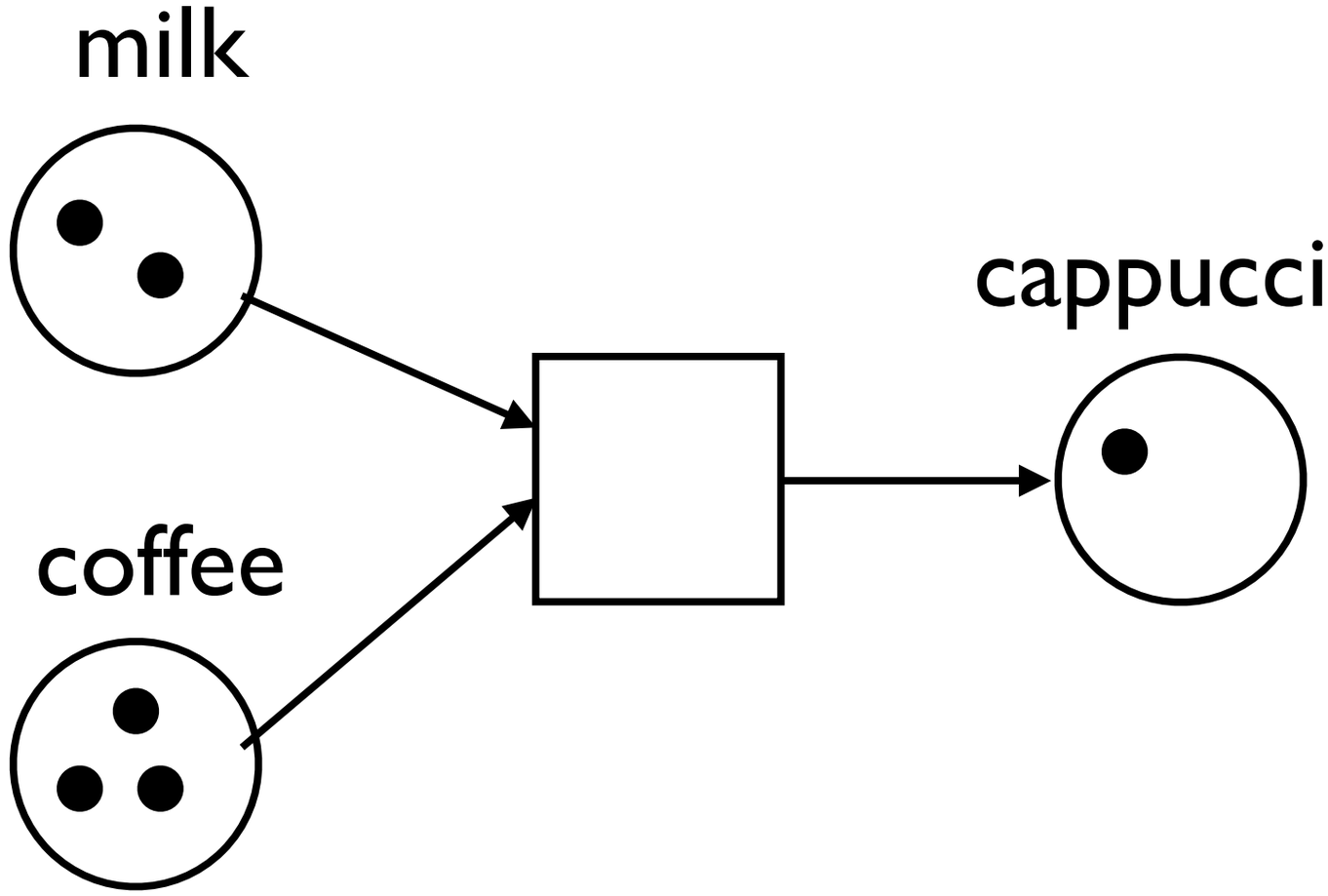
Example: AND join



Example: OR split



Example

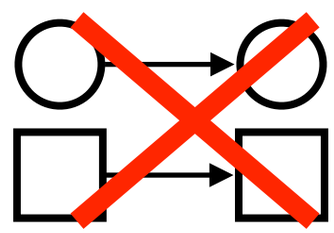


Petri nets

A **Petri net** is a tuple (P, T, F, M_0) 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**;

$$P \cap T = \emptyset$$



Pre-set

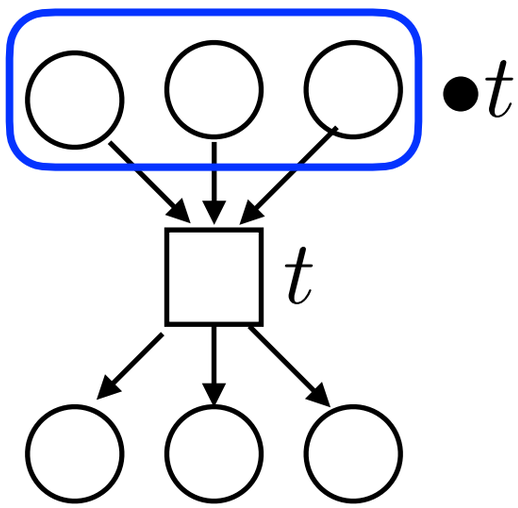
A place p is an input place for transition t iff

$$(p, t) \in F$$

We let $\bullet t$ denote the set of input places of t.

(pre-set of t)

tokens needed from



Pre-set and post-set

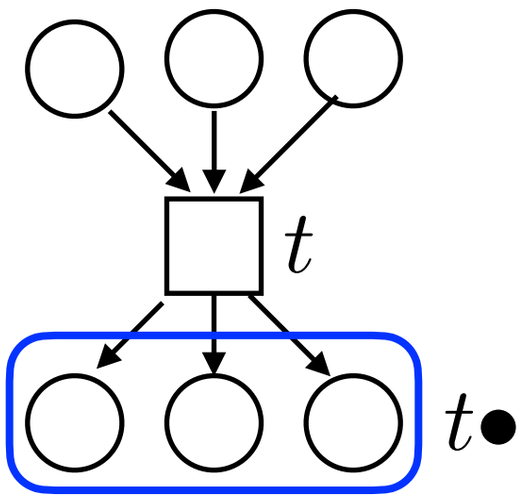
A place p is an output place for transition t iff

$$(t, p) \in F$$

We let $t\bullet$ denote the set of output places of t.

(post-set of t)

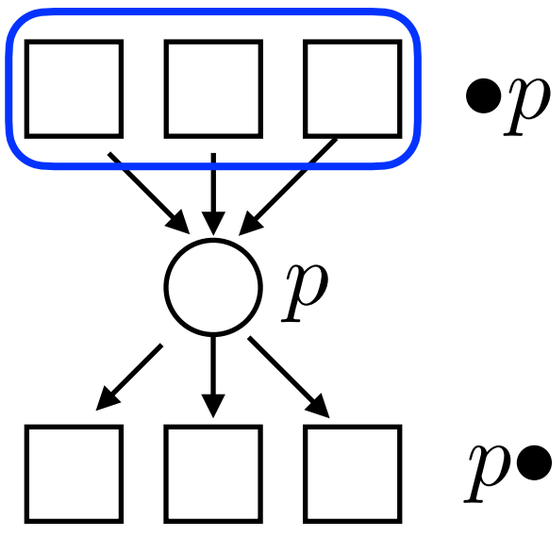
tokens produced in



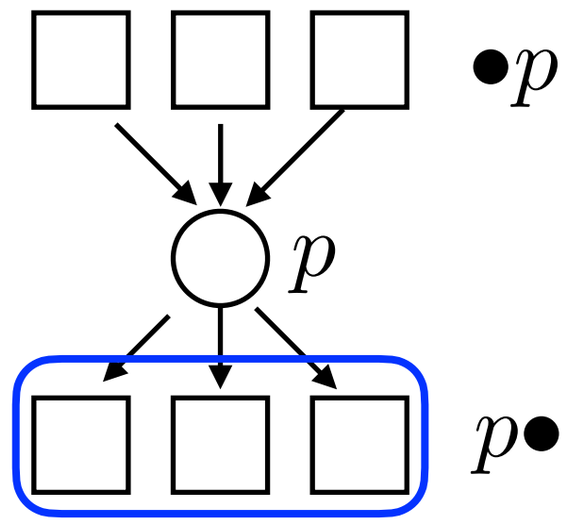
Pre-set and post-set

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



where tokens can come from



where tokens can go to

Pre-set and post-set

Formally:

$$\bullet x = \{\ y \mid (y, x) \in F\ \} \quad \text{pre-set}$$
$$x \bullet = \{\ y \mid (x, y) \in F\ \} \quad \text{post-set}$$

Token game

Enabling and firing

A transition t is enabled if each of its input places contains at least one token

When an enabled transition fires
it consumes a token from each input place
it produces a token into each output place

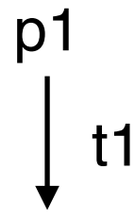
Some remarks

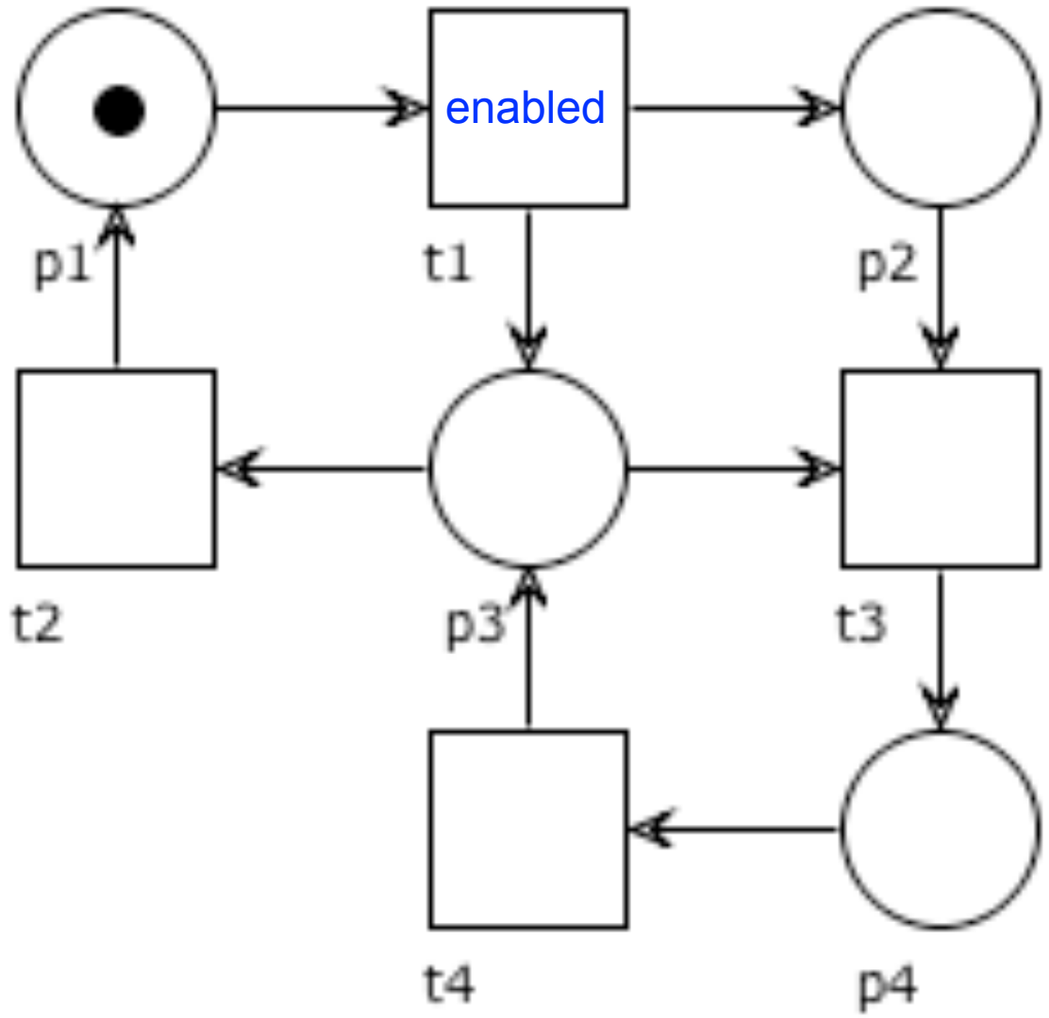
Firing is an atomic action

Our semantics is interleaving:
multiple transitions may be enabled,
but only one fires at a time

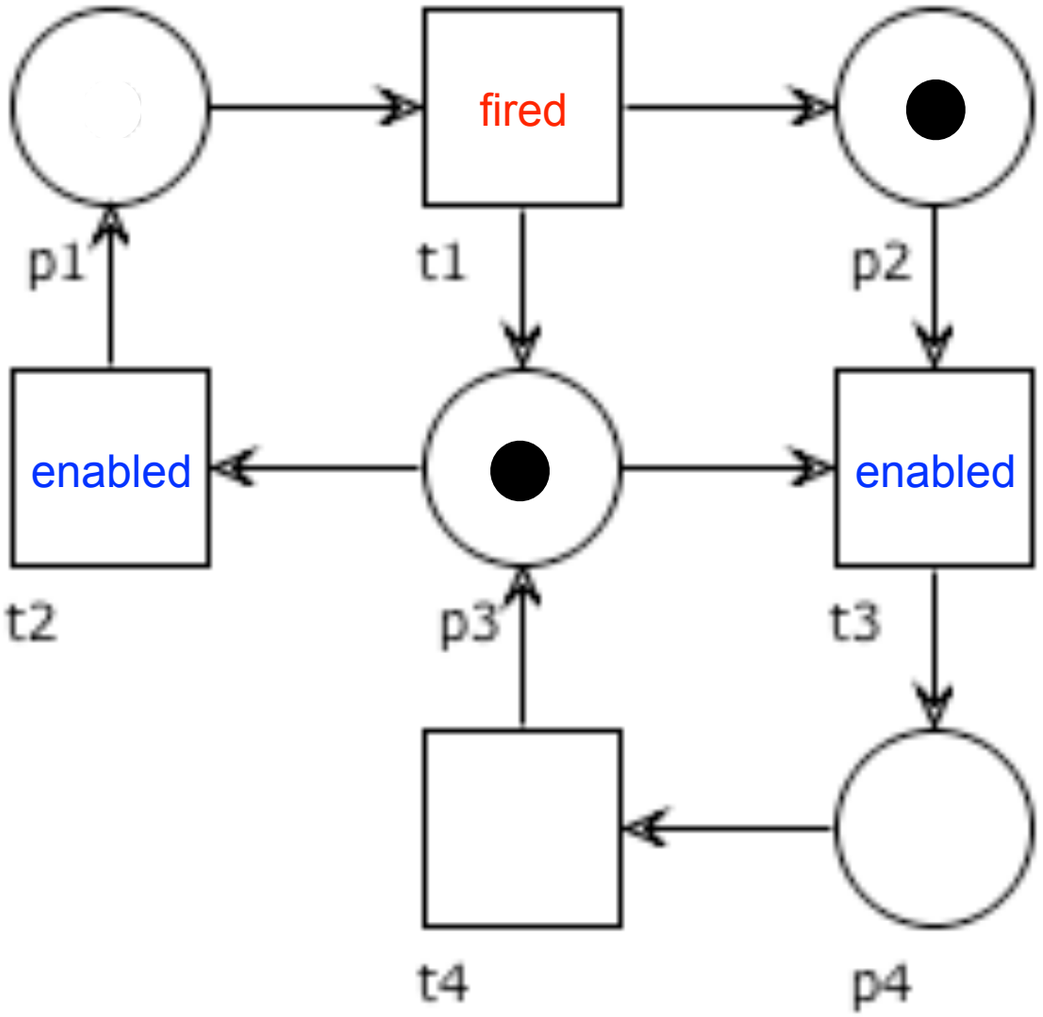
The network is static, but
the overall number of tokens may vary over time
(if transitions are fired for which the number of input places is not equal to the number of output places)

Example





Example



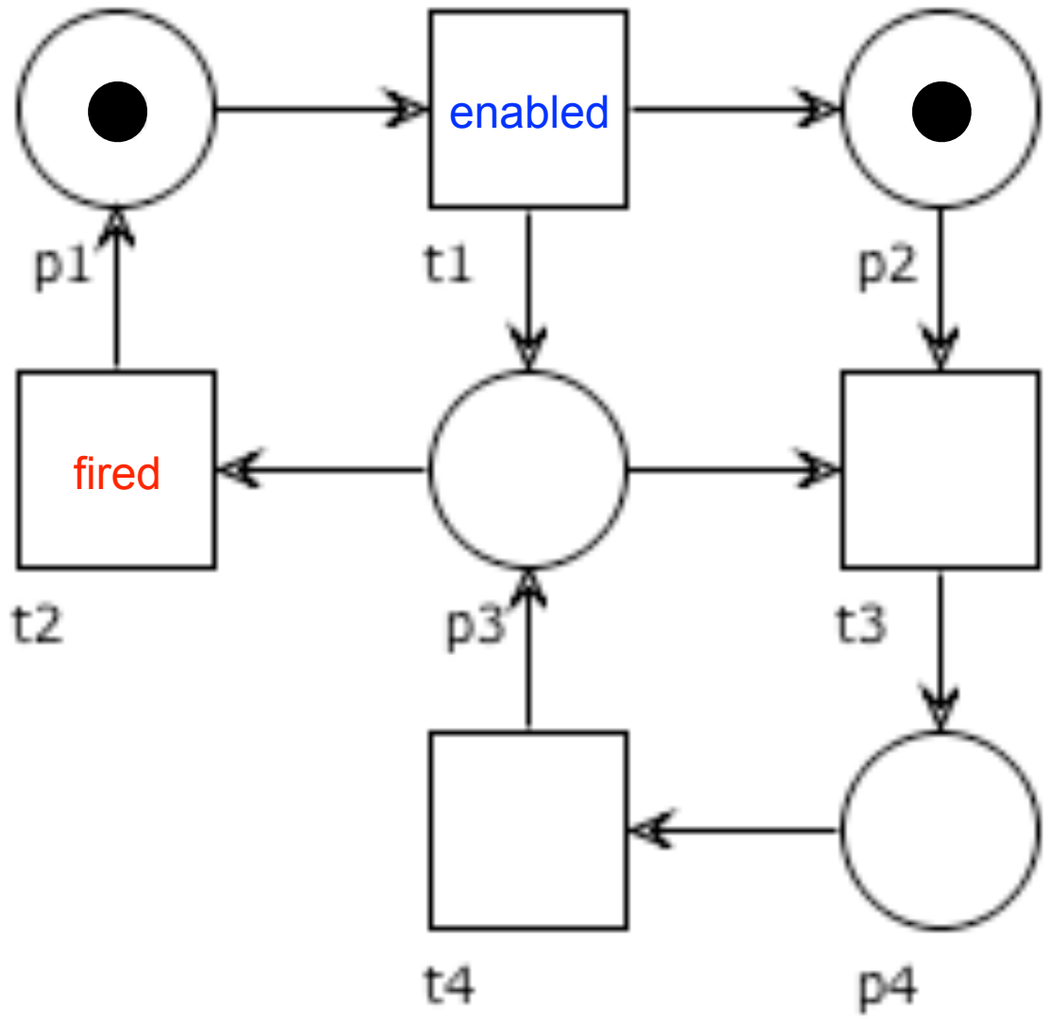
p1

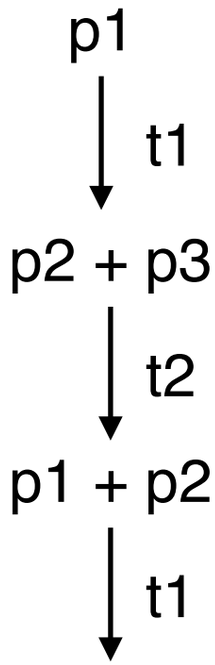
t1

p2 + p3

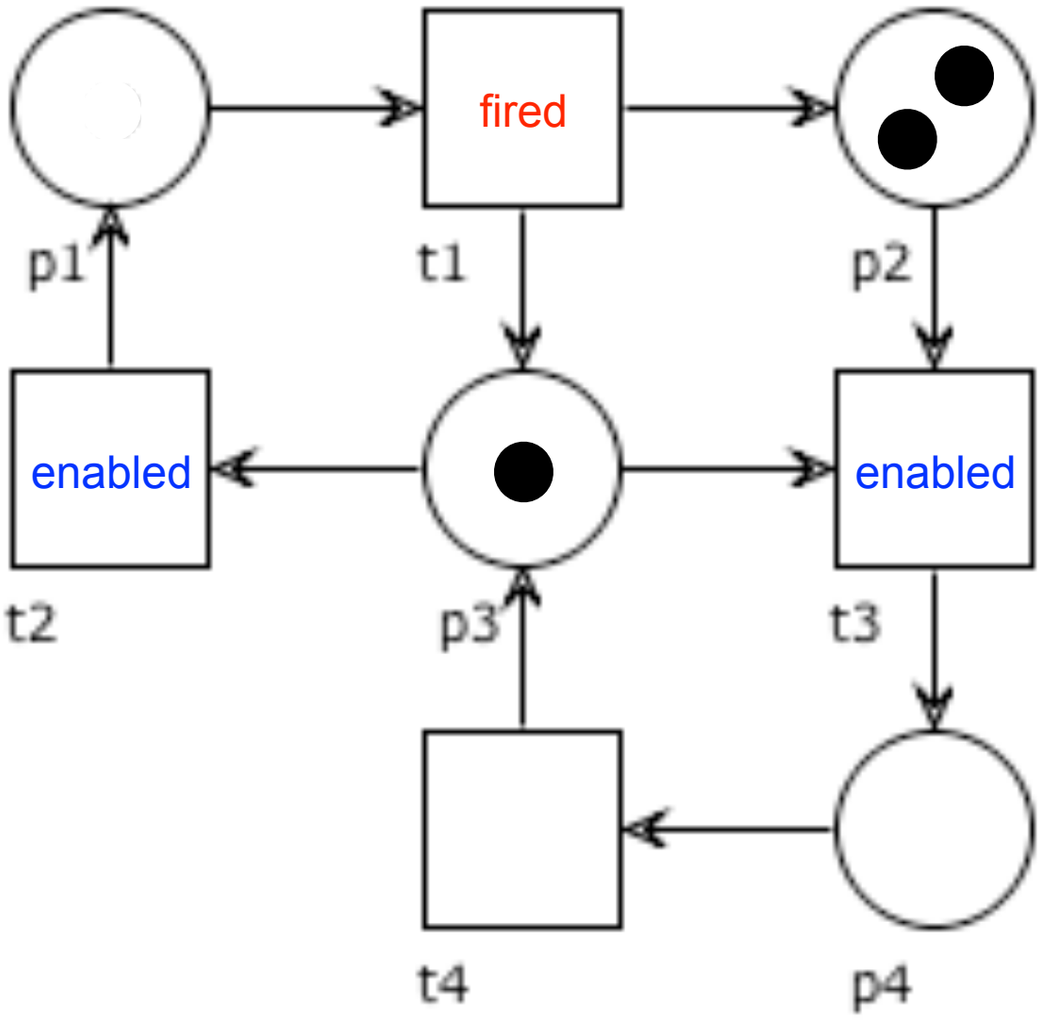
t2 / t3

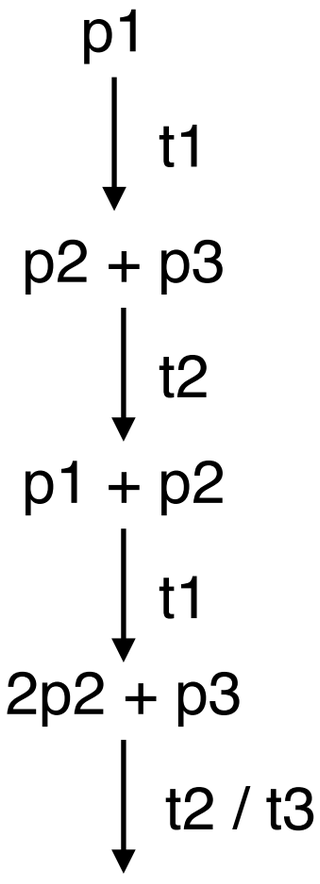
Example



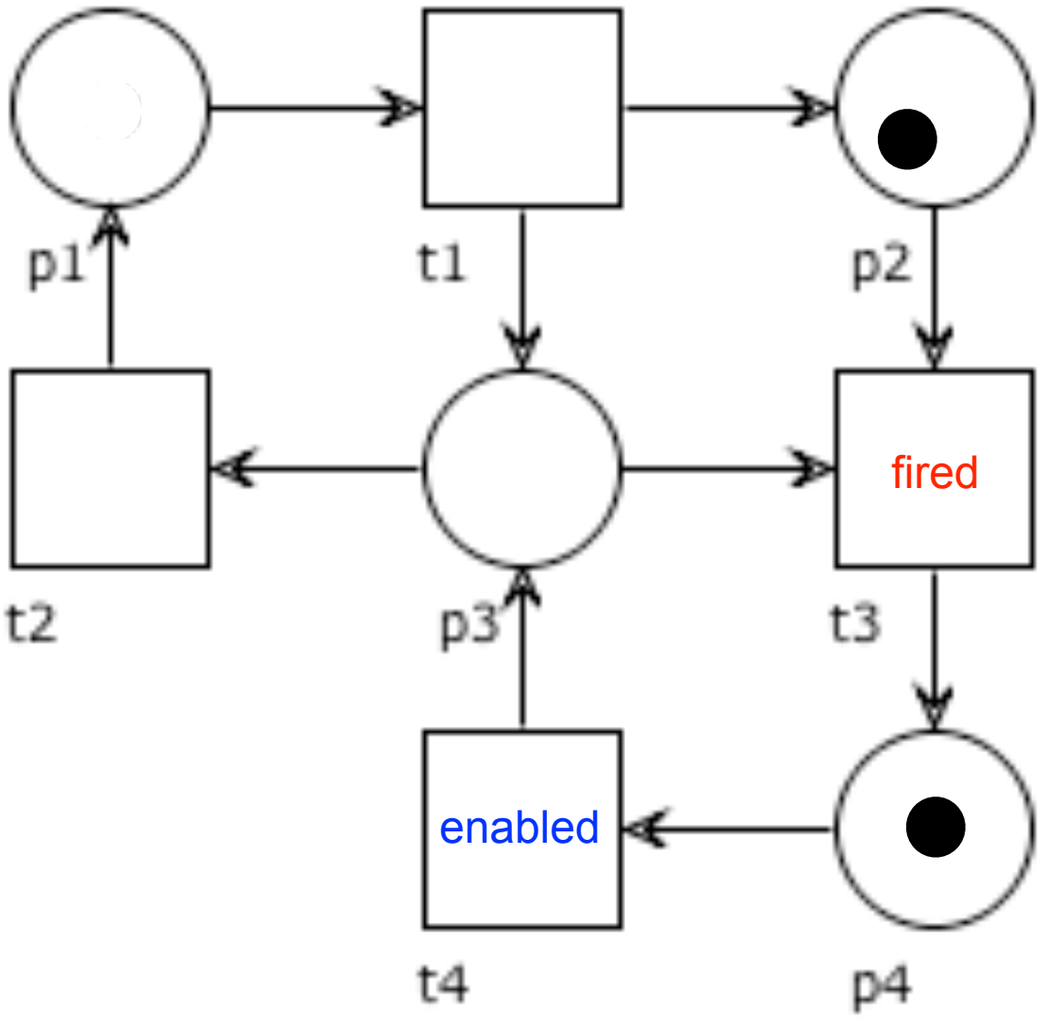


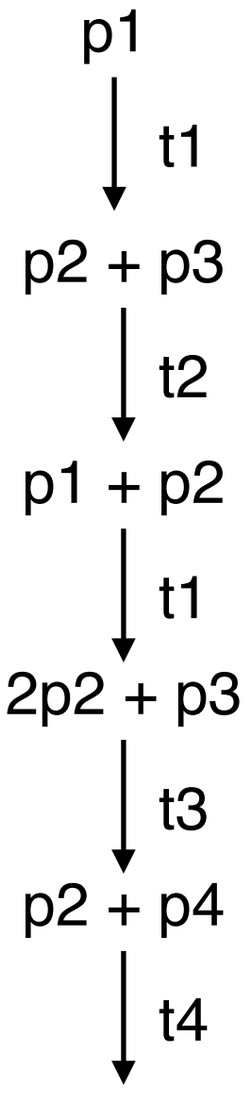
Example


fired
p1
t1
p2
enabled
enabled
t2
p3
t3
t4
p4

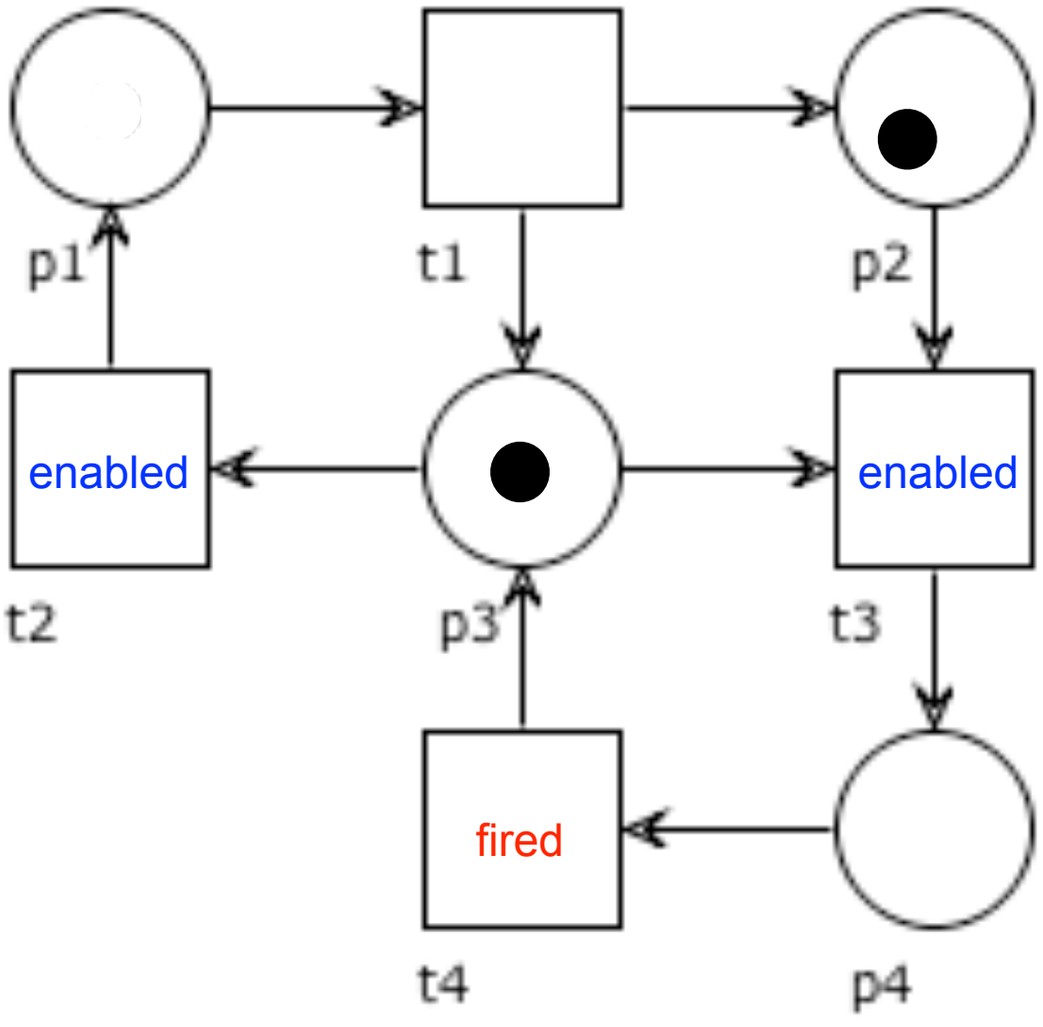

p1
t1
p2 + p3
t2
p1 + p2
t1
2p2 + p3
t2 / t3

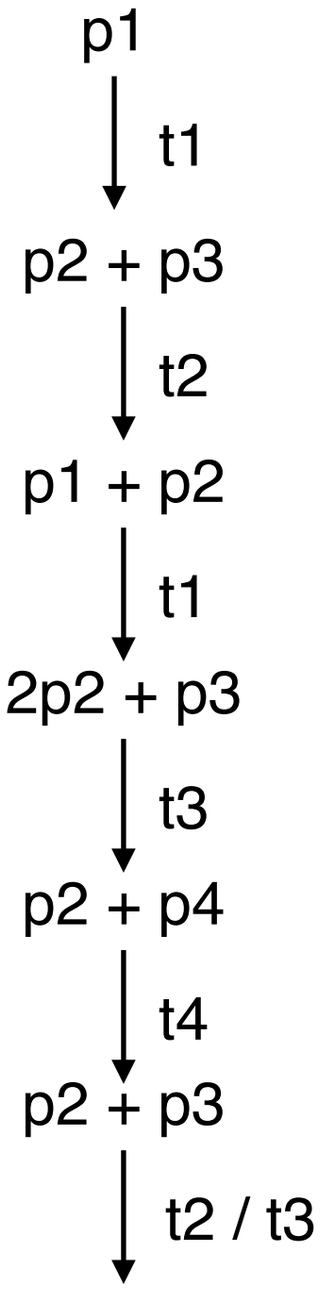
Example


p1
t1
p2
t2
p3
fired
t3
enabled
t4
p4

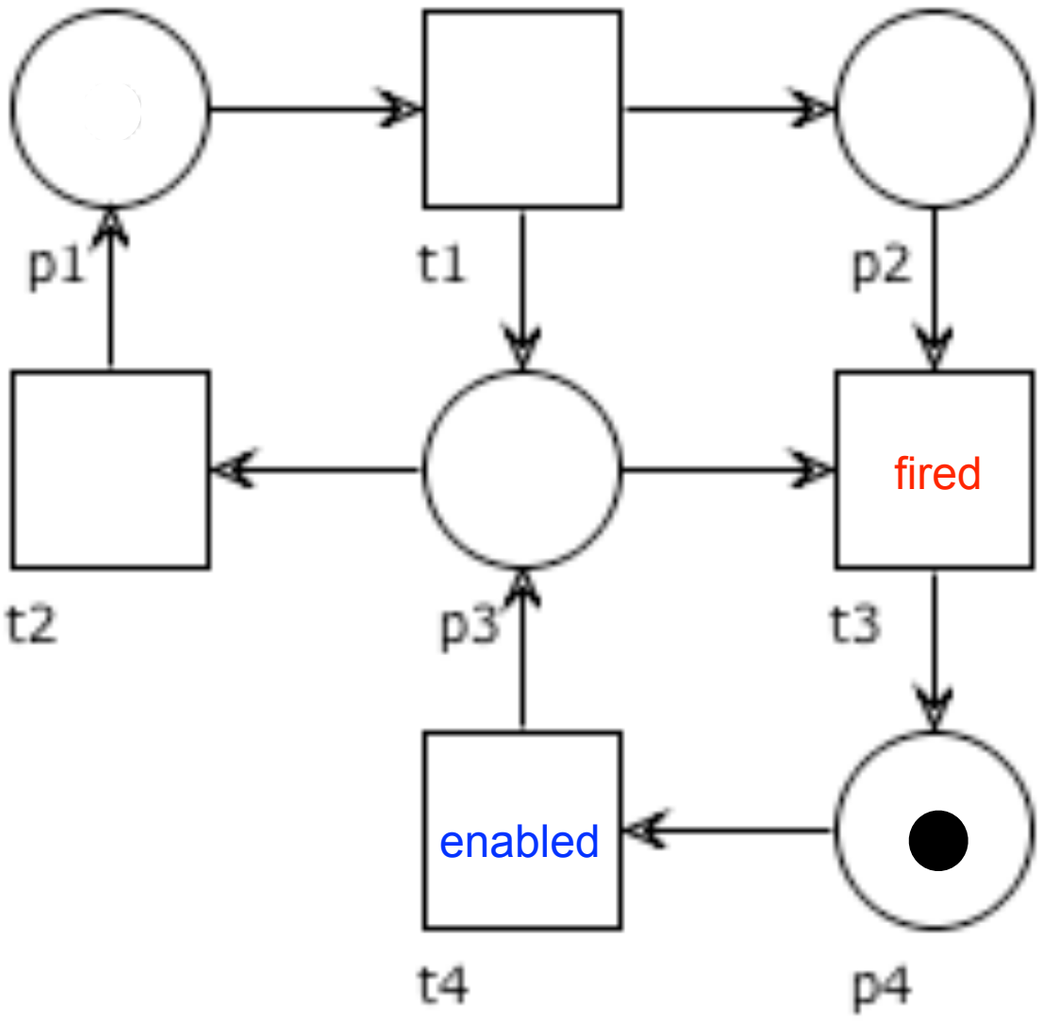

p1
t1
p2 + p3
t2
p1 + p2
t1
2p2 + p3
t3
p2 + p4
t4

Example


p1
t1
p2
enabled
enabled
t2
p3
t3
fired
t4
p4

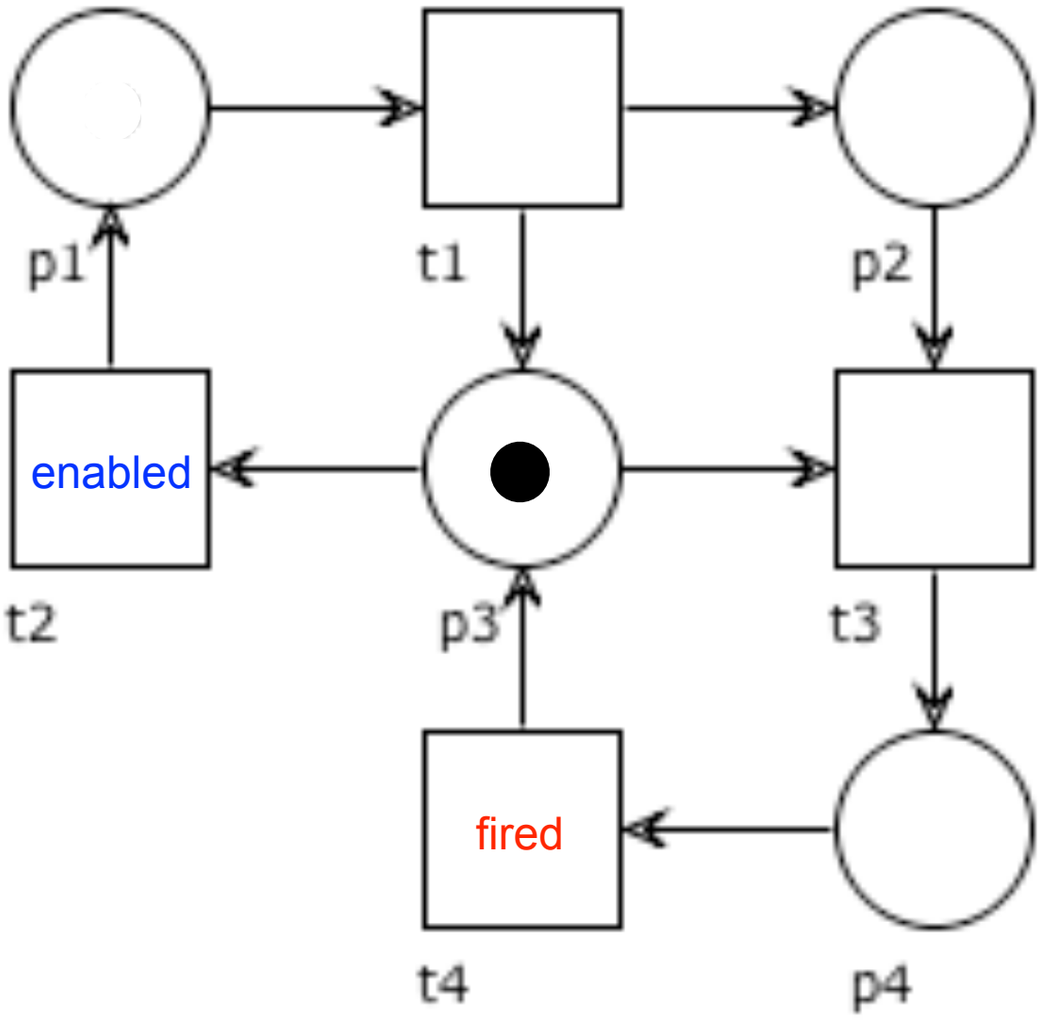

p1
t1
p2 + p3
t2
p1 + p2
t1
2p2 + p3
t3
p2 + p4
t4
p2 + p3
t2 / t3

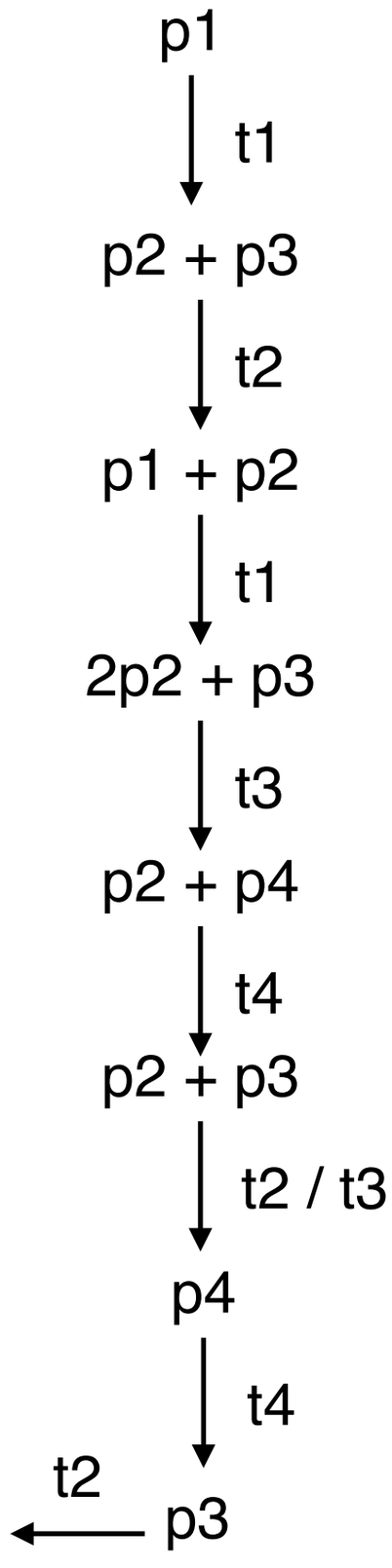
Example



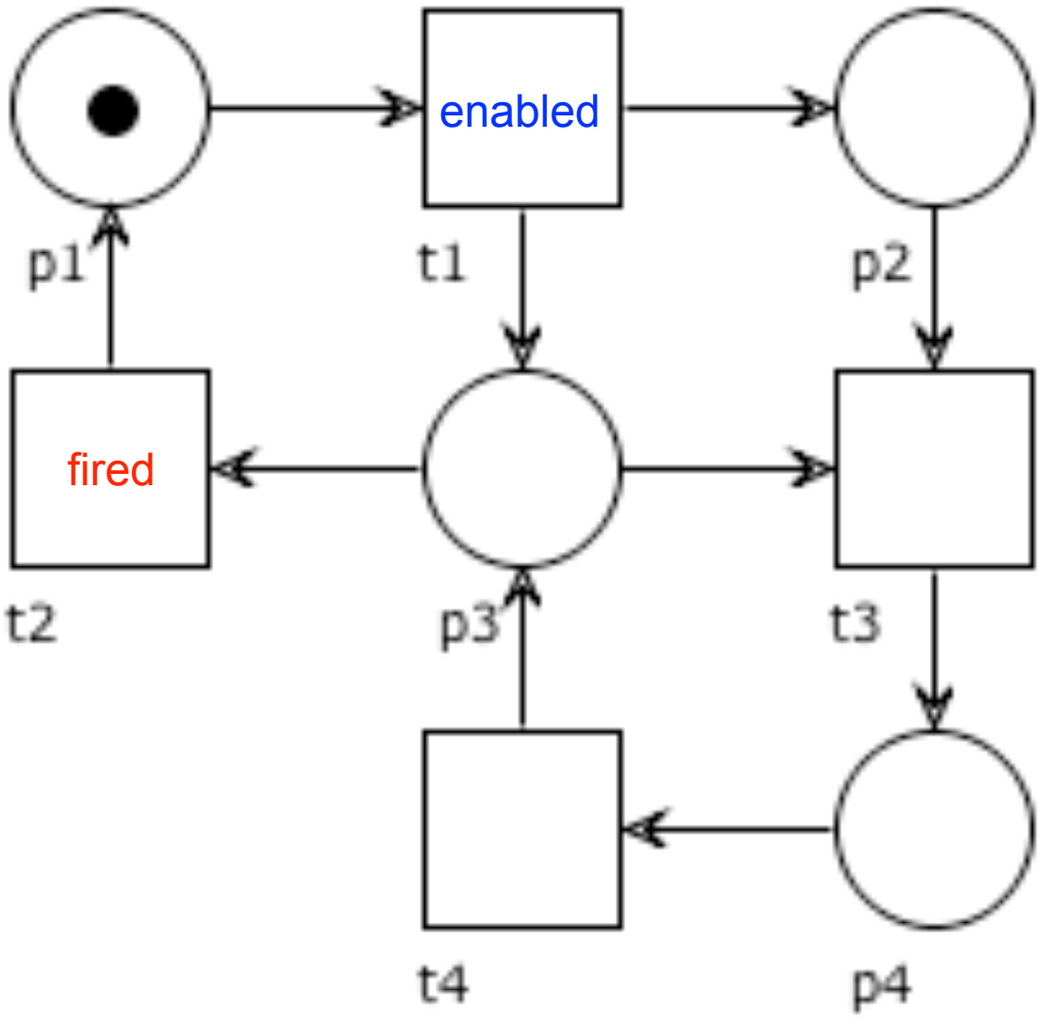
p1
t1
p2 + p3
t2
p1 + p2
t1
2p2 + p3
t3
p2 + p4
t4
p2 + p3
t2 / t3
p4
t4

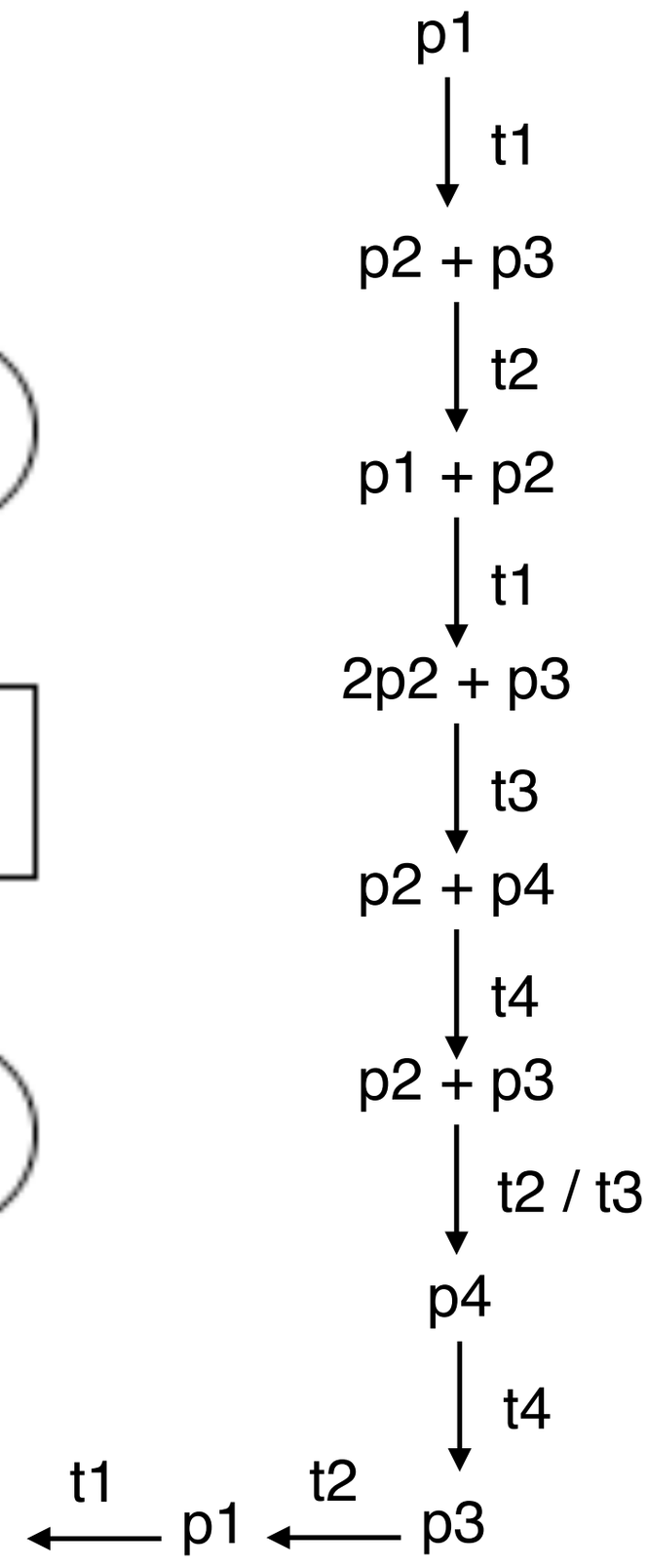
Example


p1
t1
p2
enabled
t2
p3
t3
fired
t4
p4


p1
t1
p2 + p3
t2
p1 + p2
t1
2p2 + p3
t3
p2 + p4
t4
p2 + p3
t2 / t3
p4
t4
t2
p3

Example


enabled
p1
t1
p2
fired
t2
p3
t3
t4
p4


p1
t1
p2 + p3
t2
p1 + p2
t1
2p2 + p3
t3
p2 + p4
t4
p2 + p3
t2 / t3
p4
t4
t1
t2
p1
p3

Workflow nets

Workflow nets features

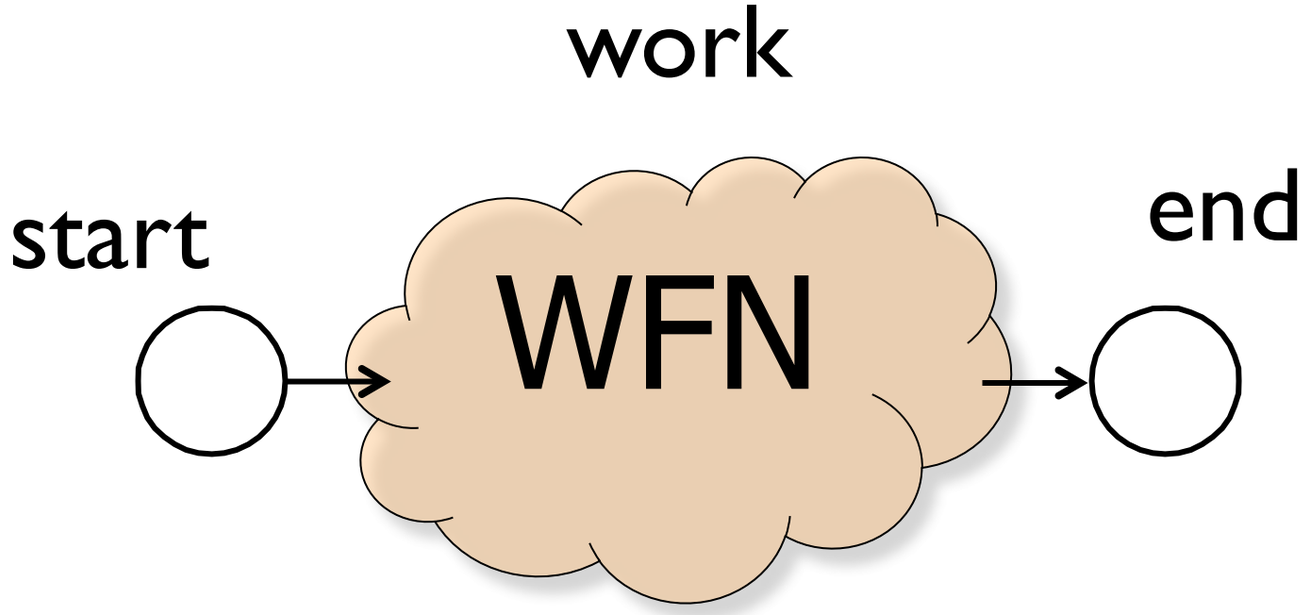
Tailored to the representation of business processes

Formal (unambiguous) semantics

Structural restrictions

Decorated graphical representation

Workflow net: idea



Workflow net

Definition:
A Petri net (P, T, F) is called **workflow net** if:

1. there is a distinguished *initial place* $i \in P$ with $\bullet i = \emptyset$
2. there is a distinguished *final place* $o \in P$ with $o\bullet = \emptyset$
3. every other place and transition belongs to a path from i to o

Basic properties

Lemma: In a workflow net there is a **unique** node with no incoming arc

Proof:

Let i be the initial place and o the final one

Suppose there is another node v with no incoming arc

node v must appear in a path from i to o

since $\bullet v = \emptyset$, v must be the first node of the path

thus $v = i$

Basic properties

Lemma: In a workflow net there is a **unique** node with no outgoing arc

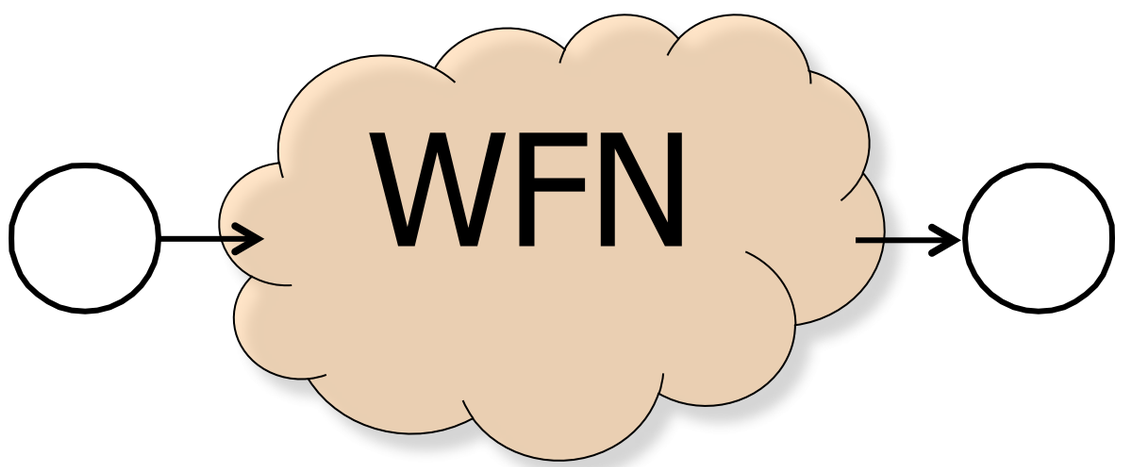
(the proof is analogous to the previous one)

Workflow net: Rationale

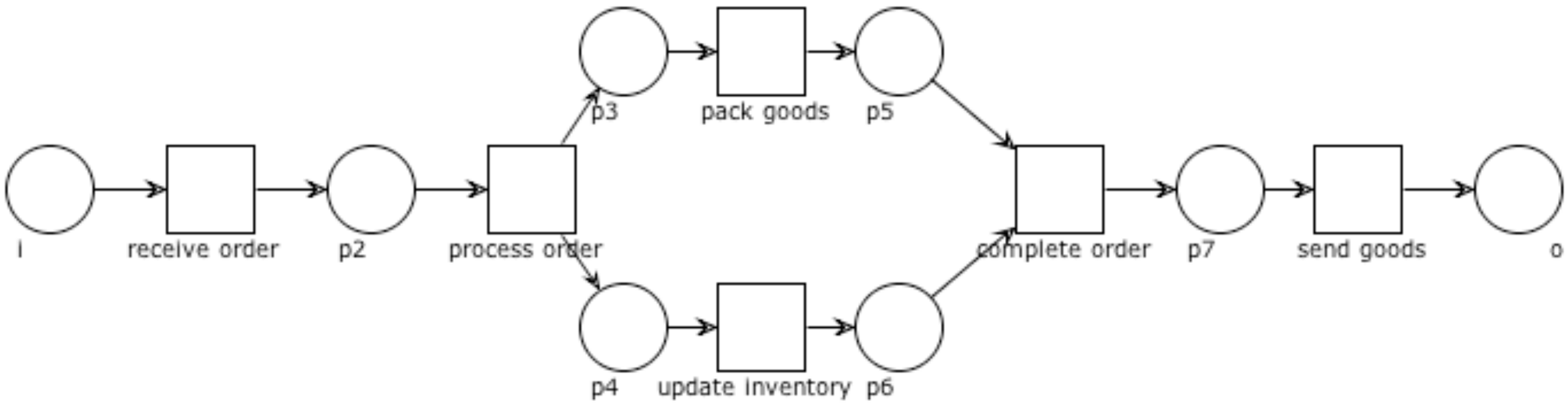
1. a token in i represents a process instance not yet started
2. a token in o represents a finished case
3. each place and each transition can participate in a case

Definition:
A Petri net (P, T, F) is called **workflow net** if:

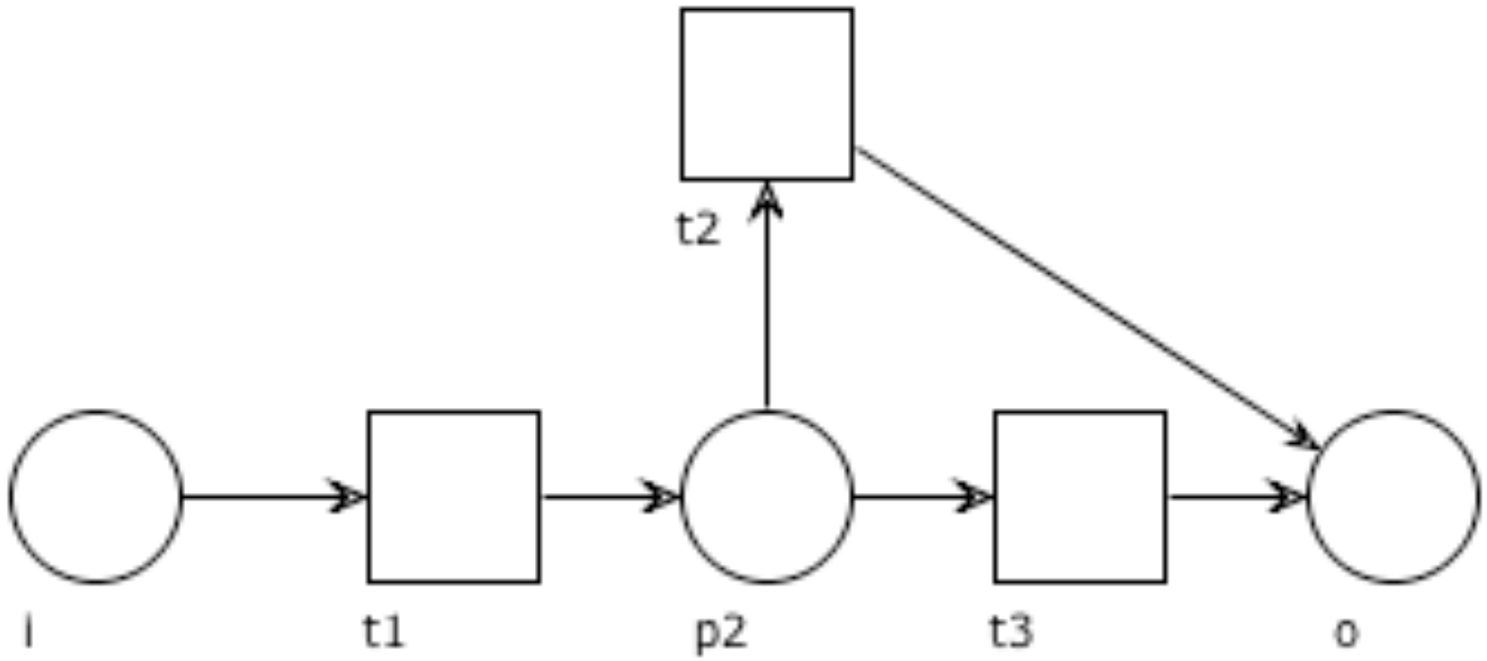
1. there is a distinguished *initial place* $i \in P$ with $\bullet i = \emptyset$
2. there is a distinguished *final place* $o \in P$ with $o\bullet = \emptyset$
3. every other place and transition belongs to a path from i to o



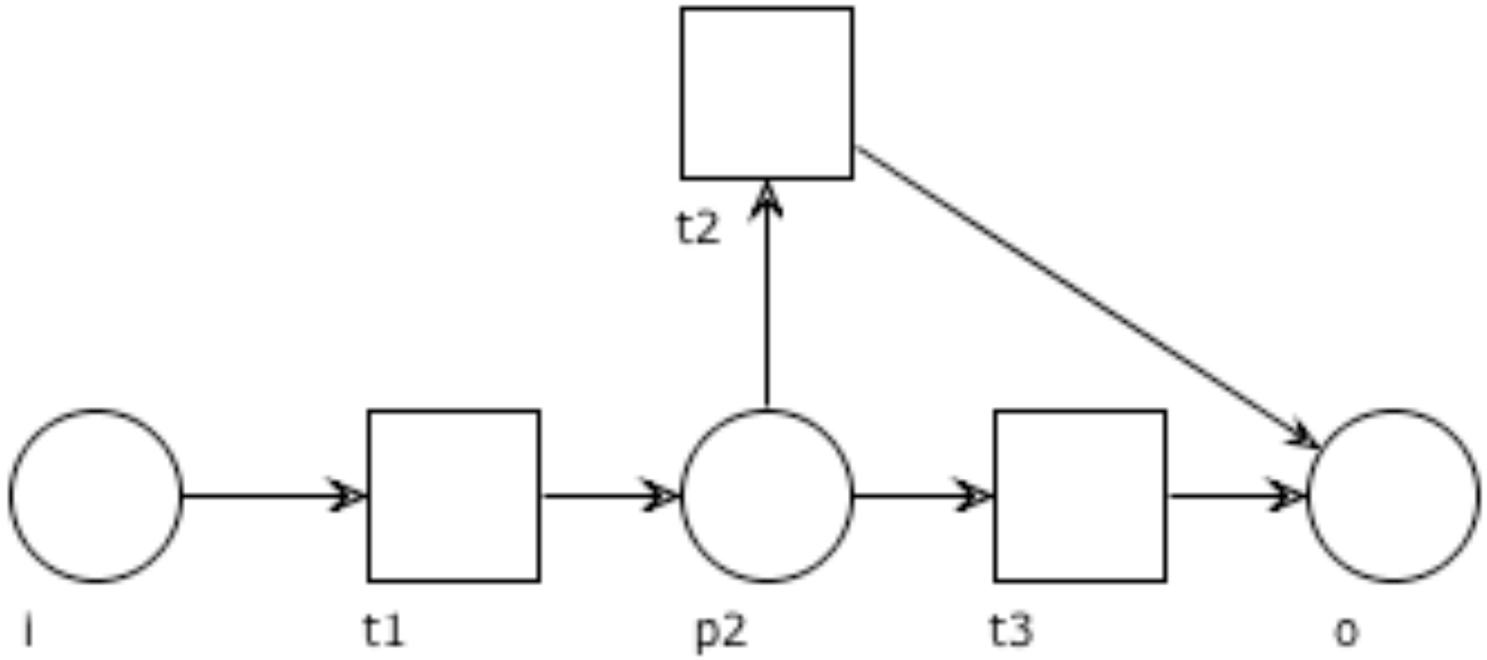
WF net: Example



Question time: WF net?

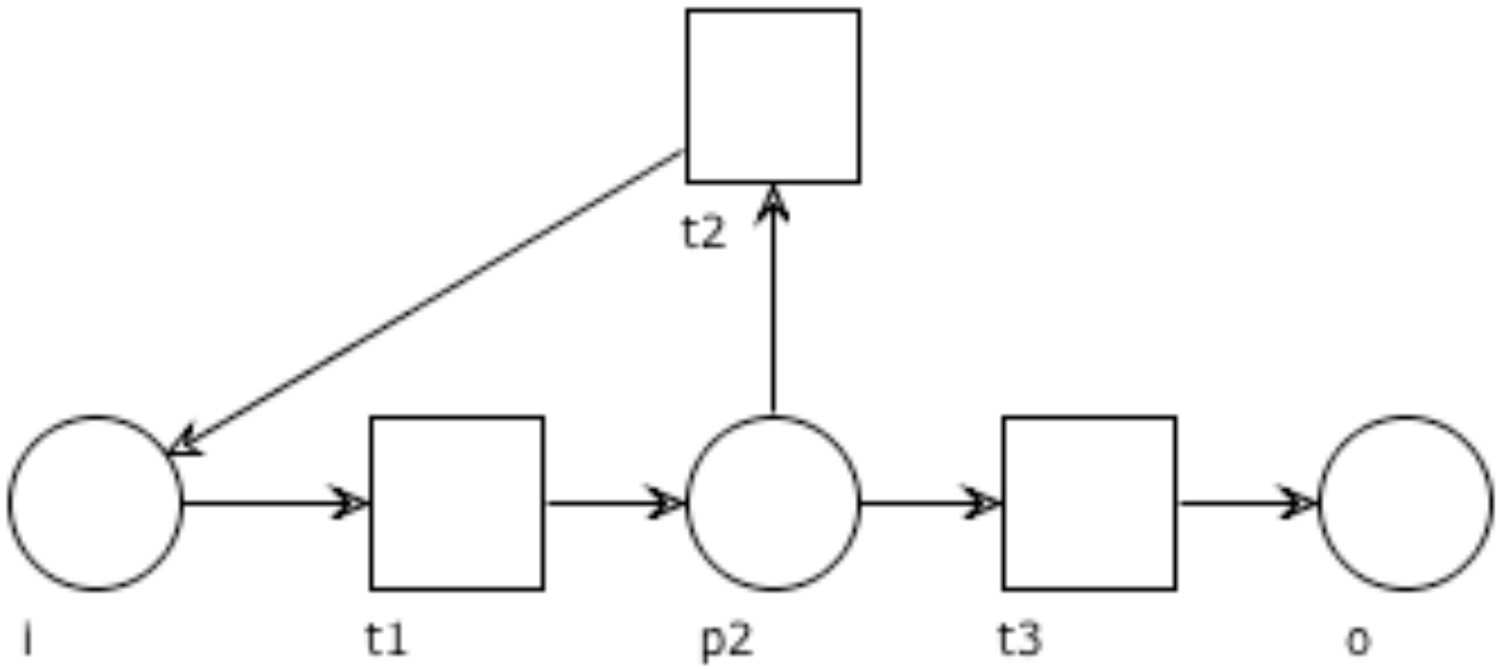


Question time: WF net?

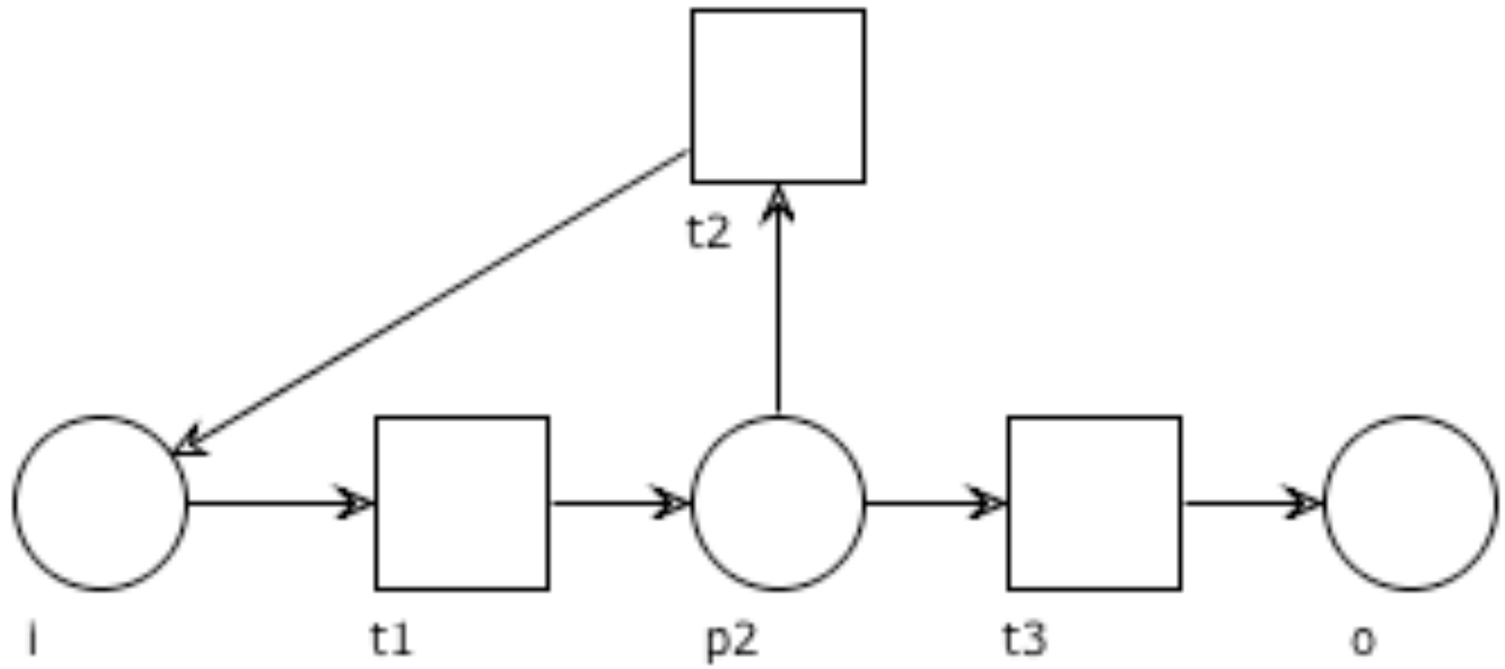


Yes

Question time: WF net?

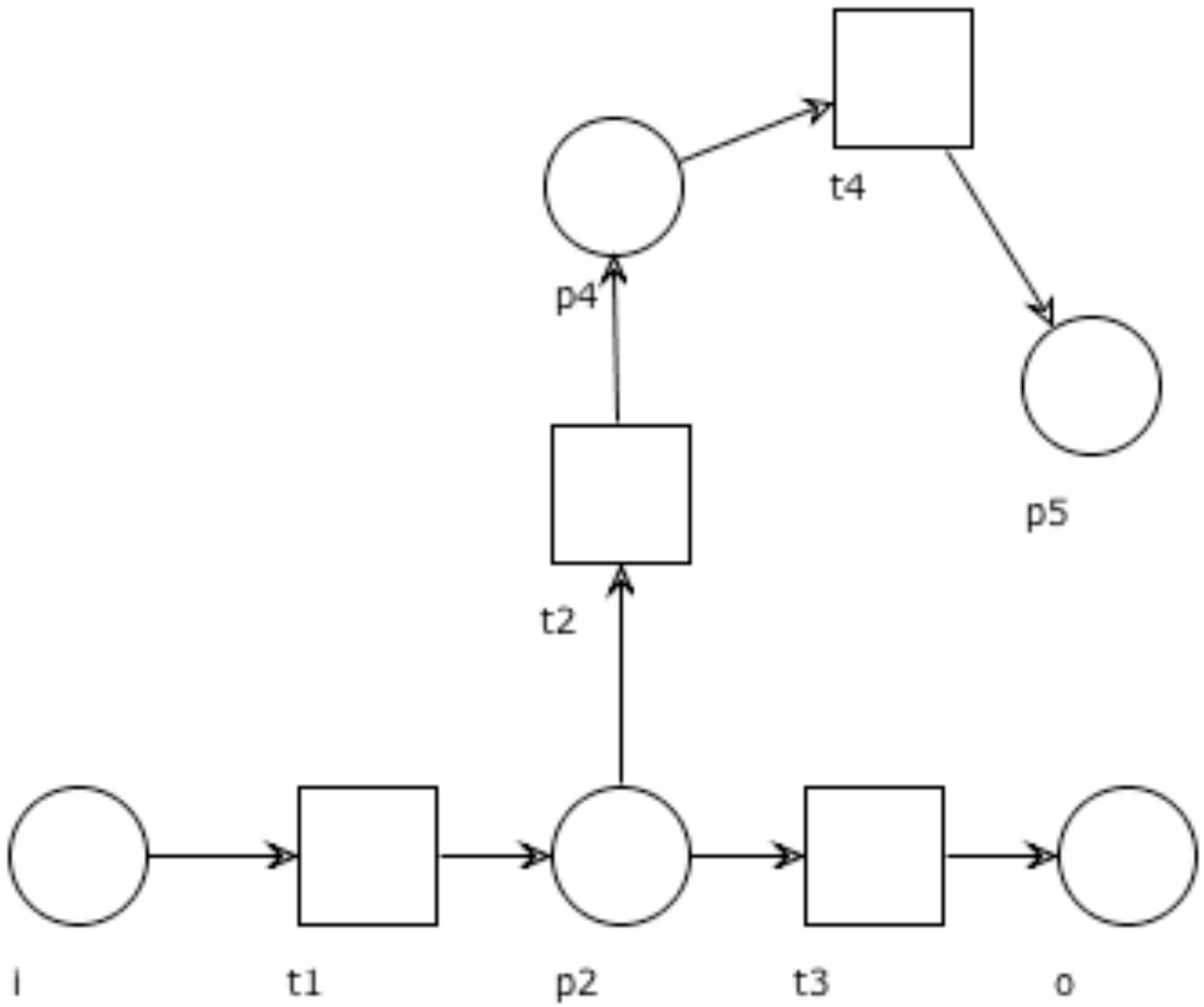


Question time: WF net?

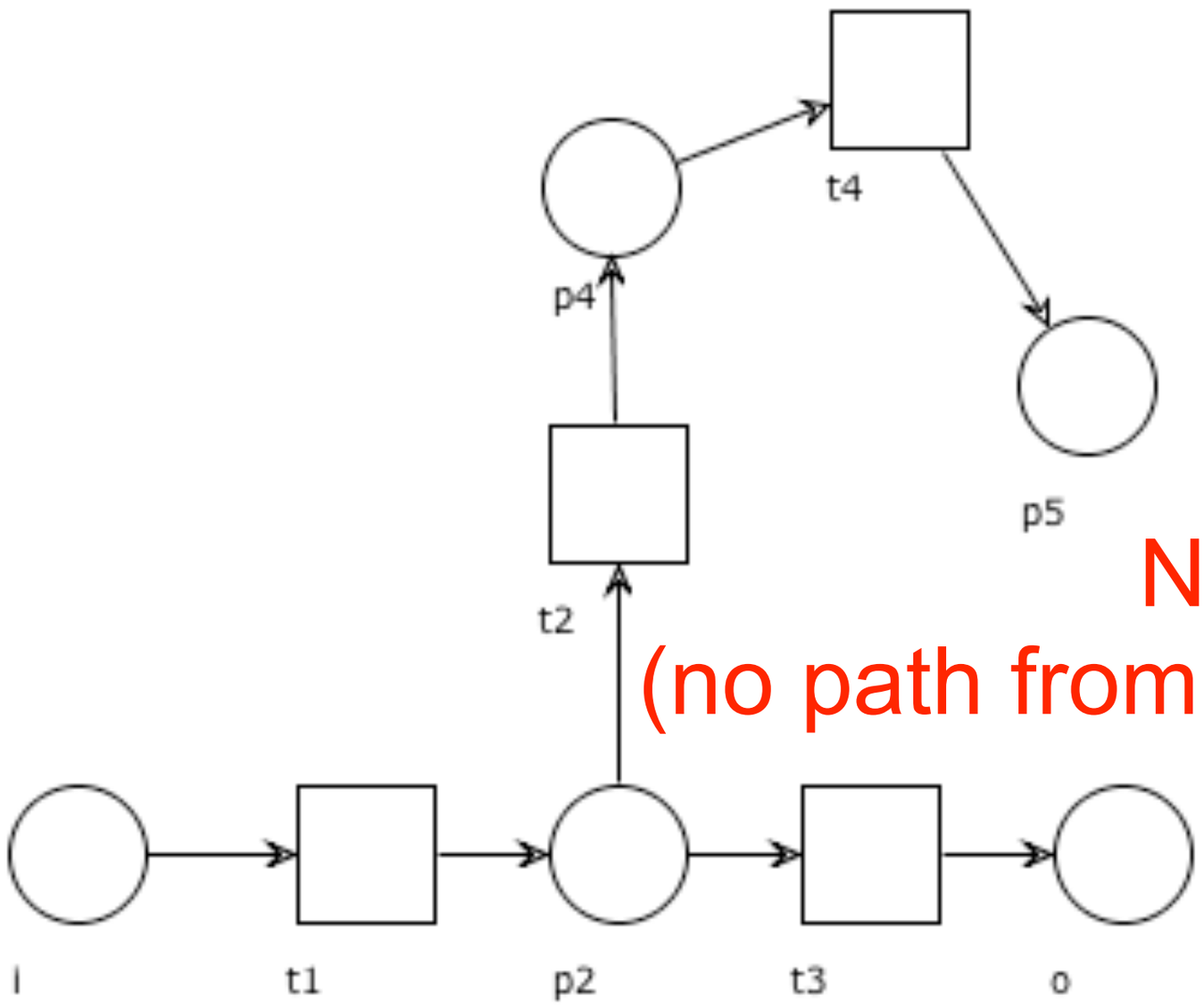


No
(no initial place)

Question time: WF net?

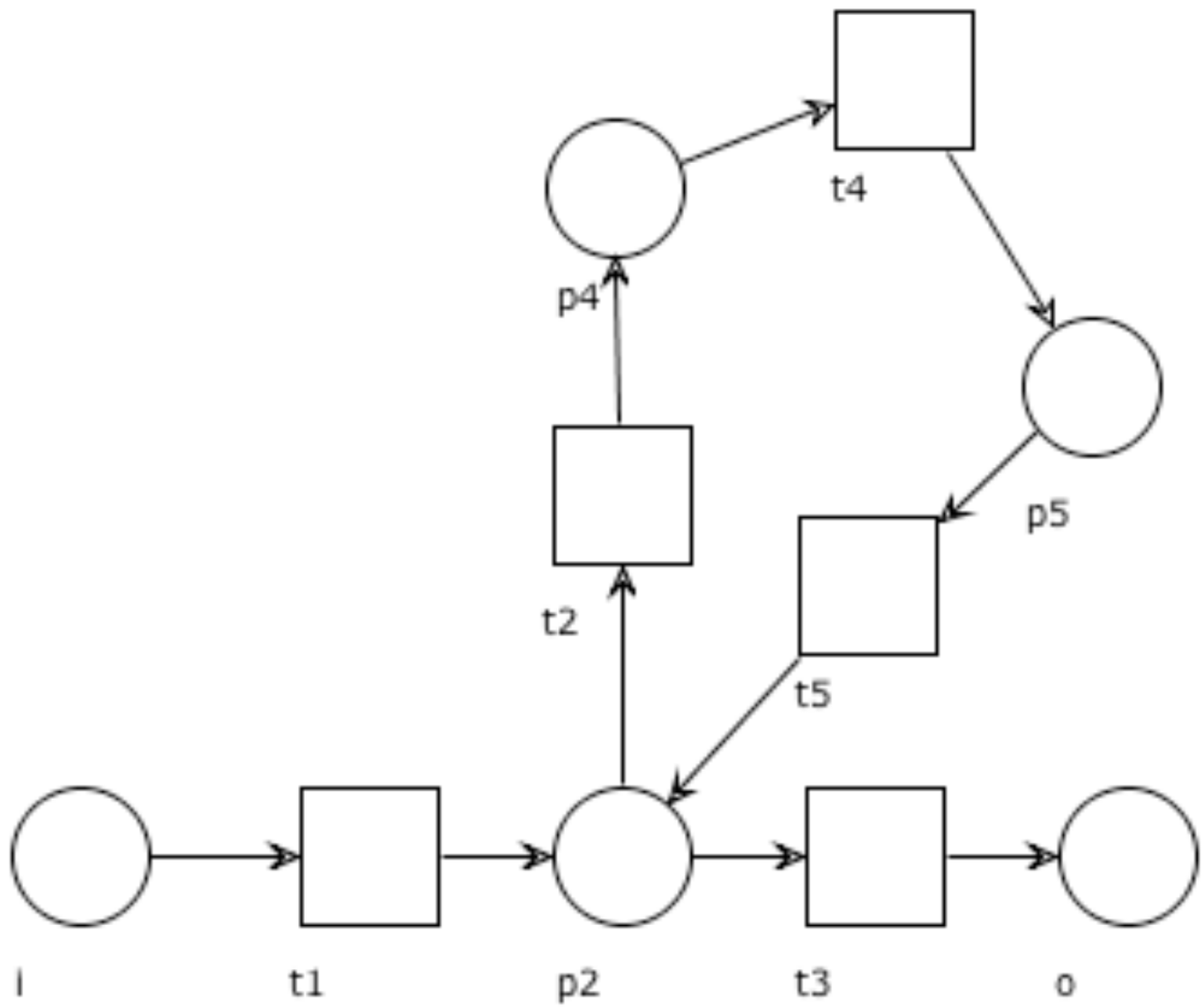


Question time: WF net?

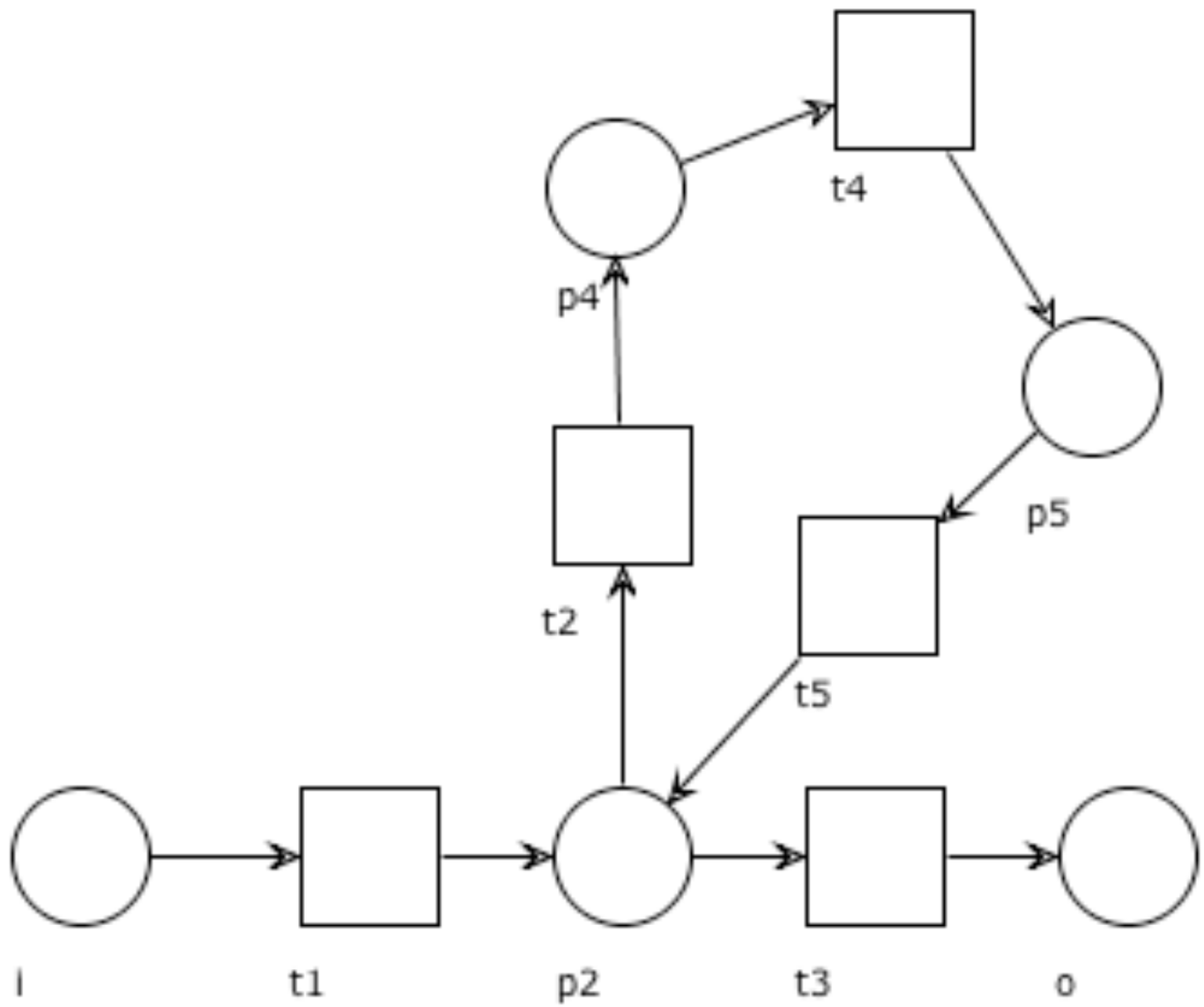


No
(no path from i to o with t2)

Question time: WF net?

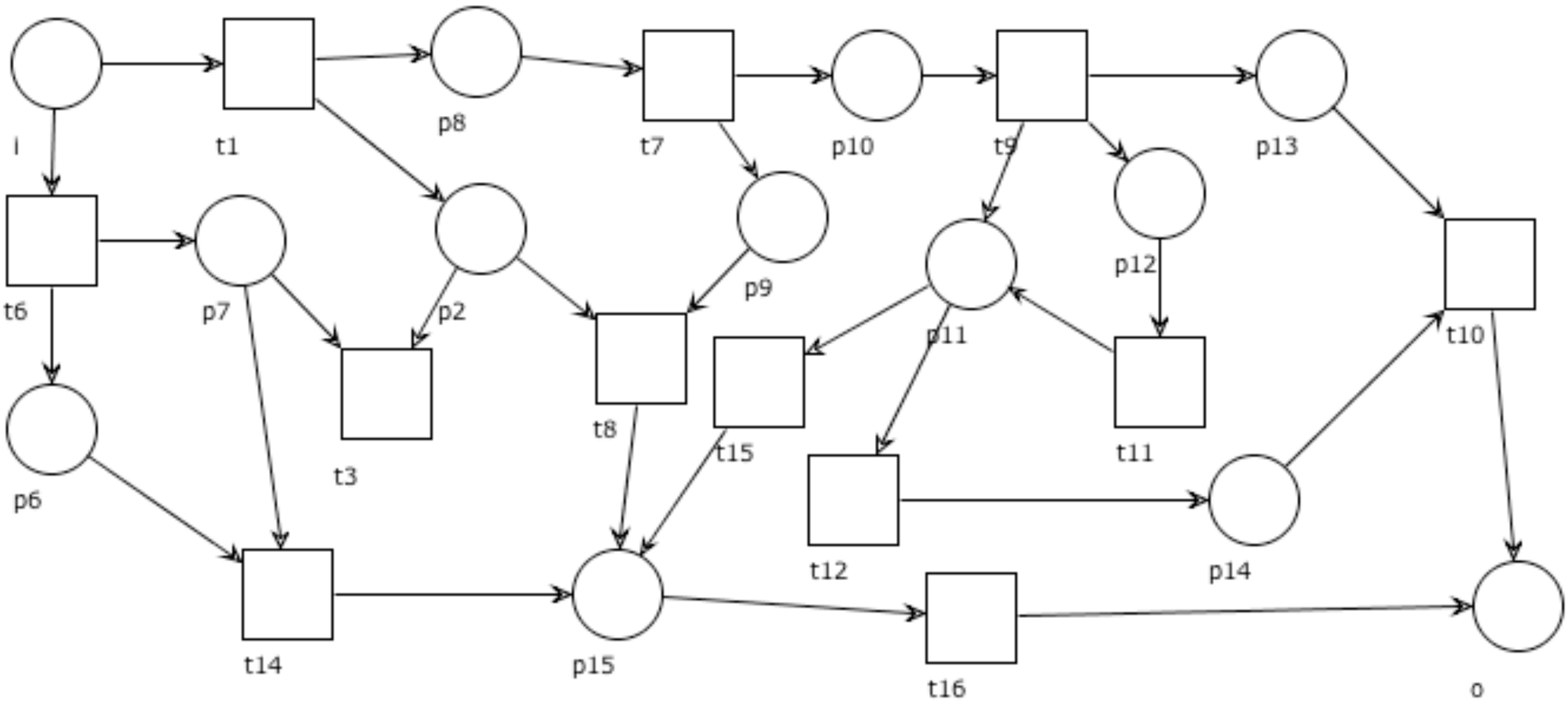


Question time: WF net?

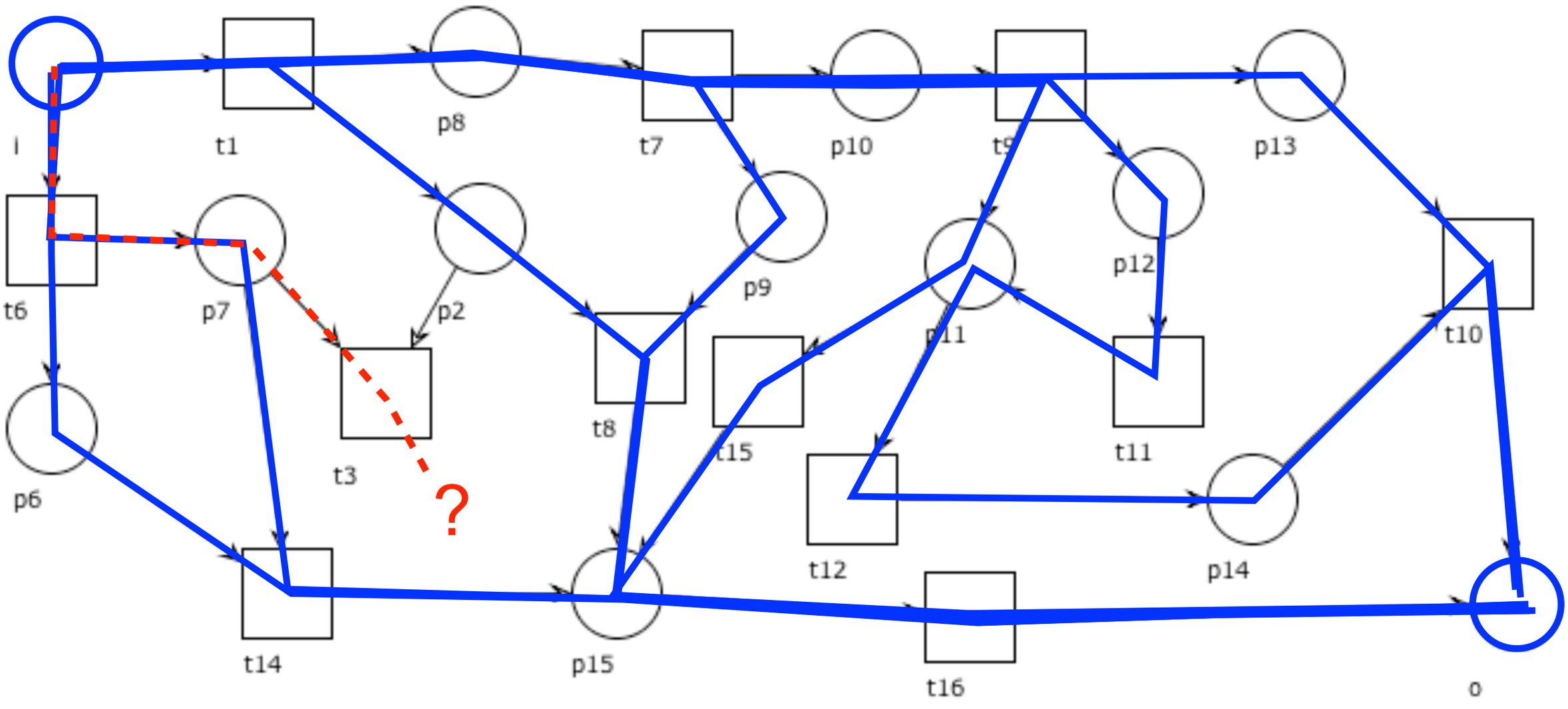


Yes

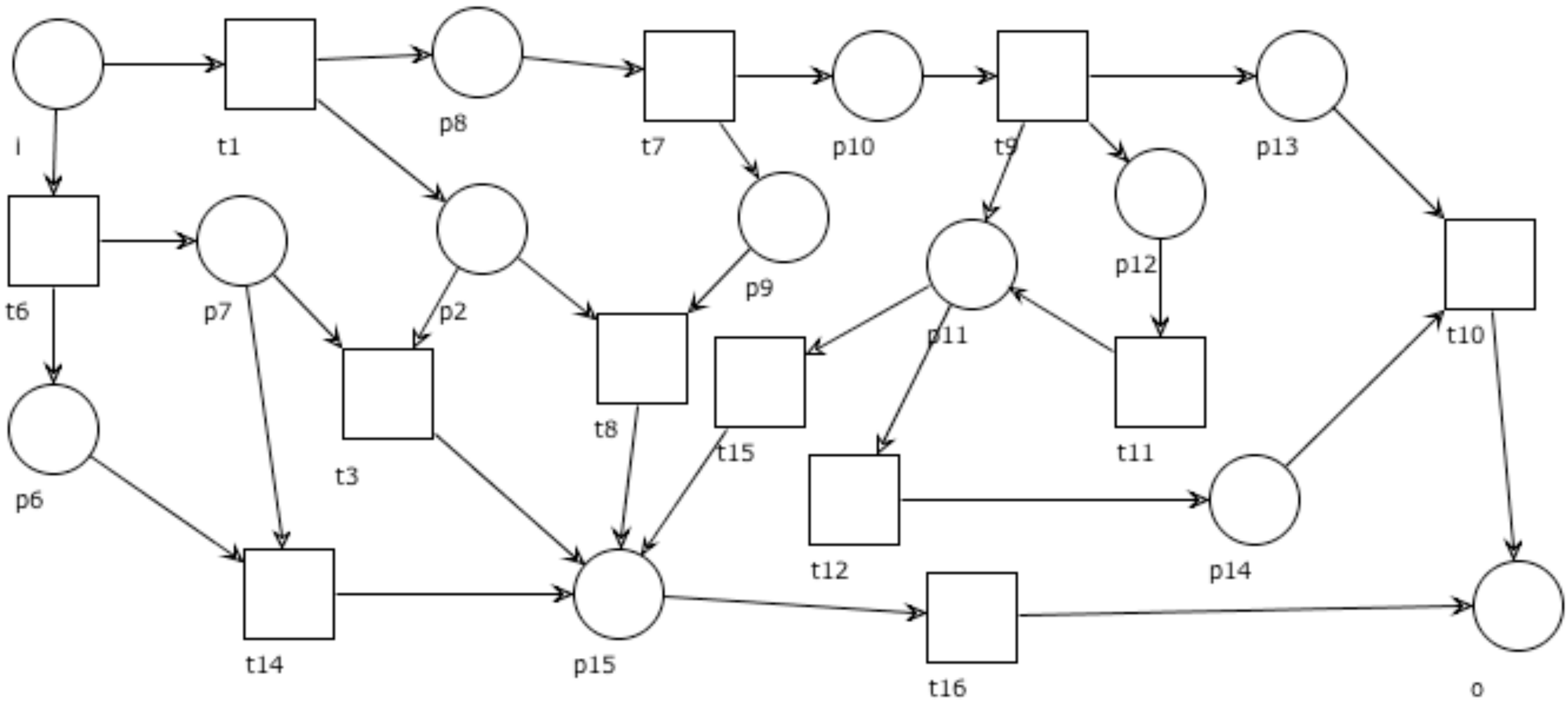
Question time: WF net?



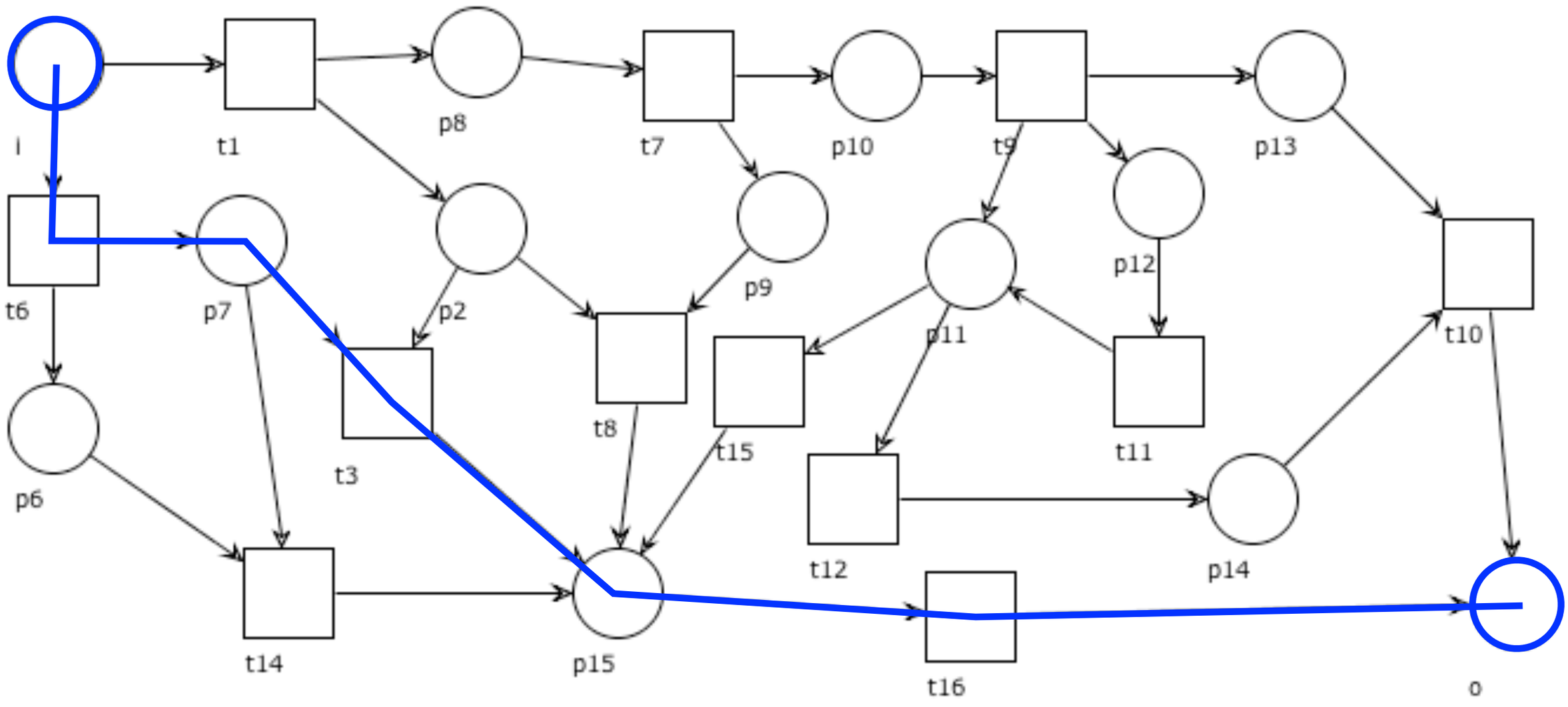
Question time: WF net?



Question time: WF net?



Question time: WF net?

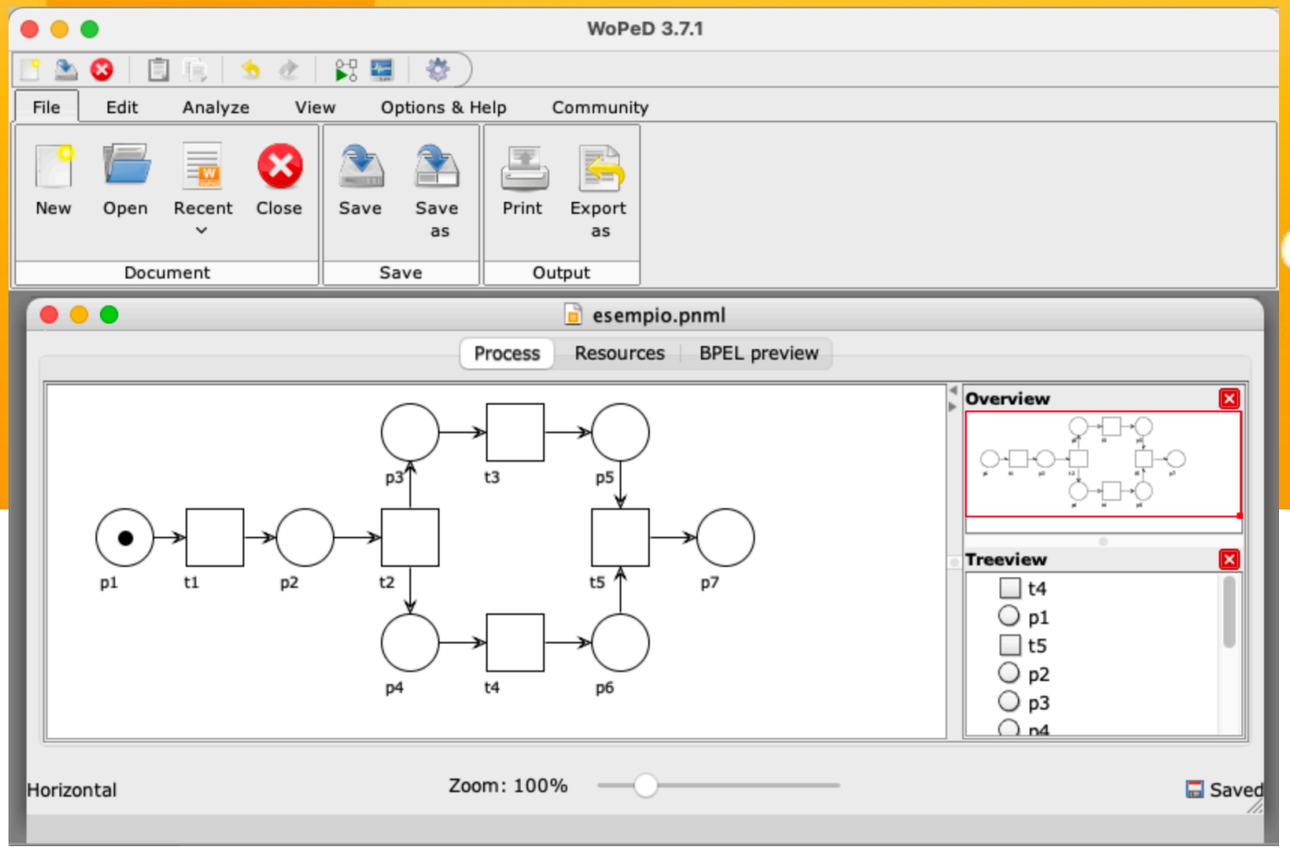


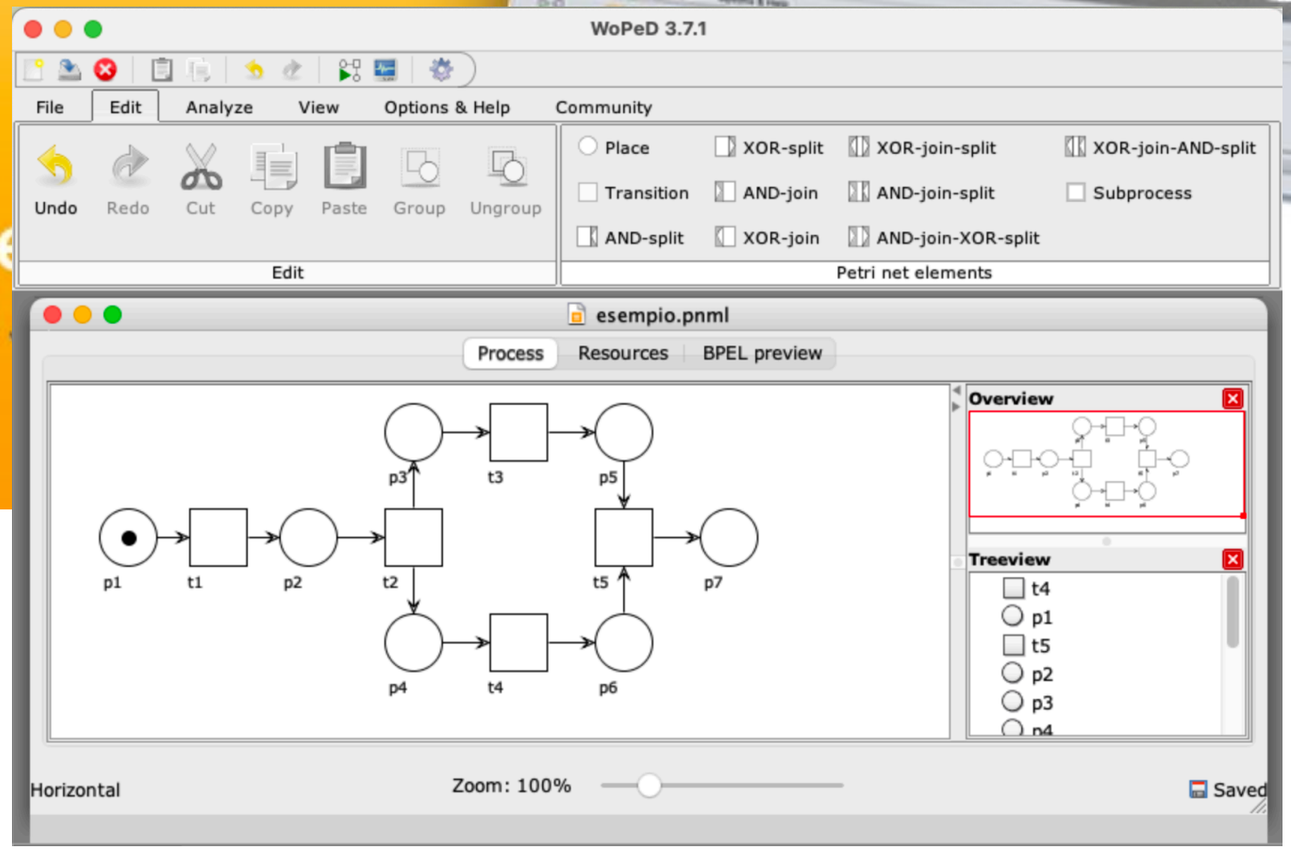
Yes

WoPeD

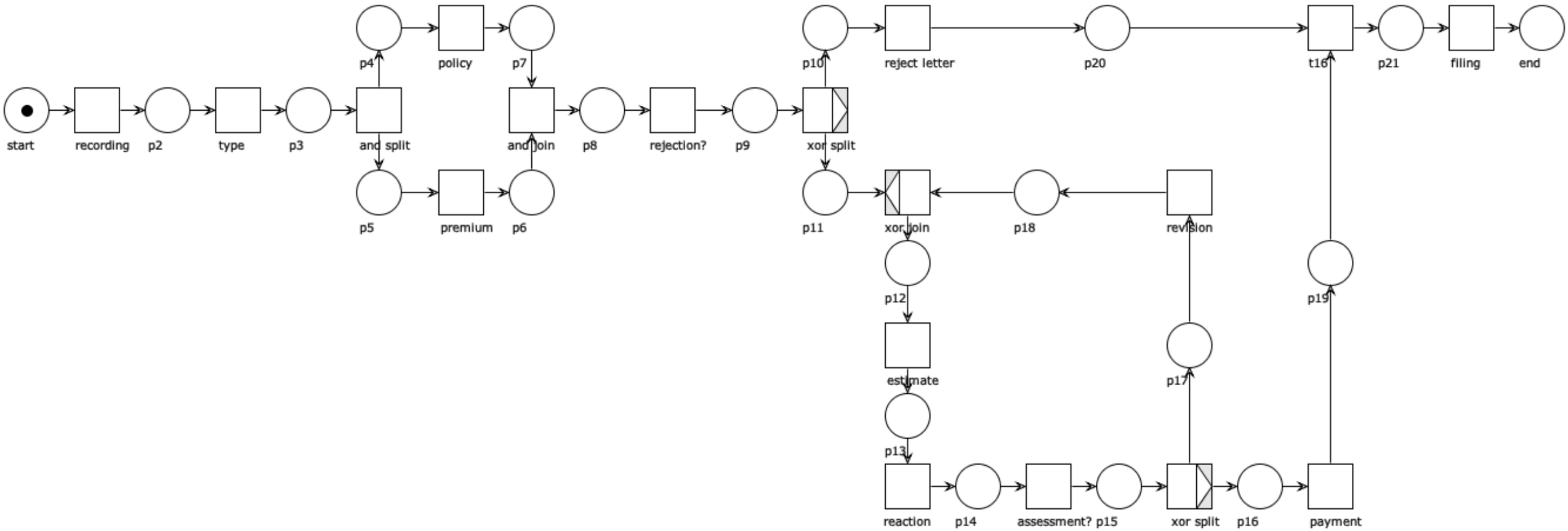
WoPeD

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





Is it a Wf net?



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

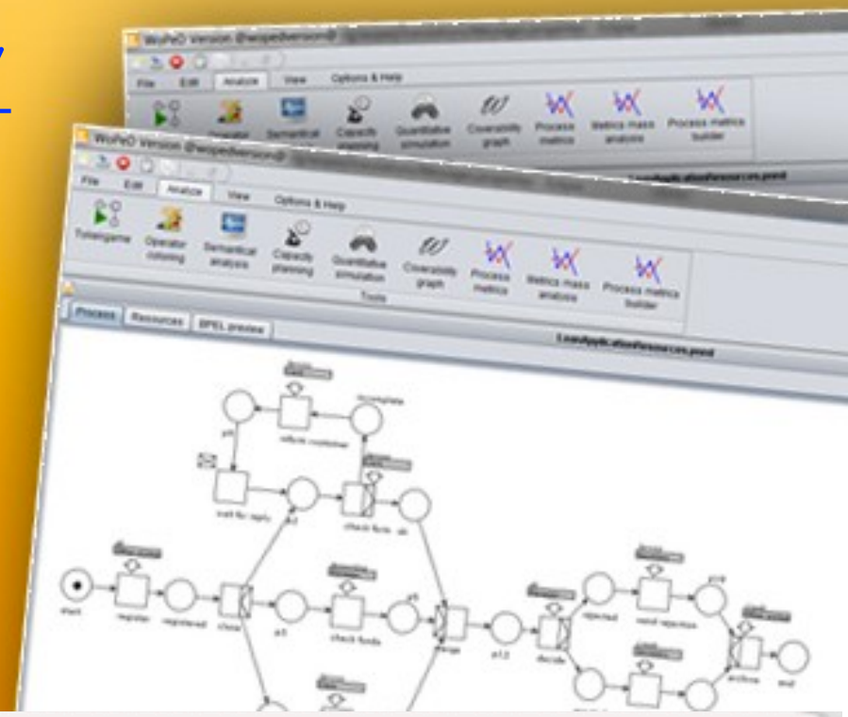
Syntax sugar (denotations)

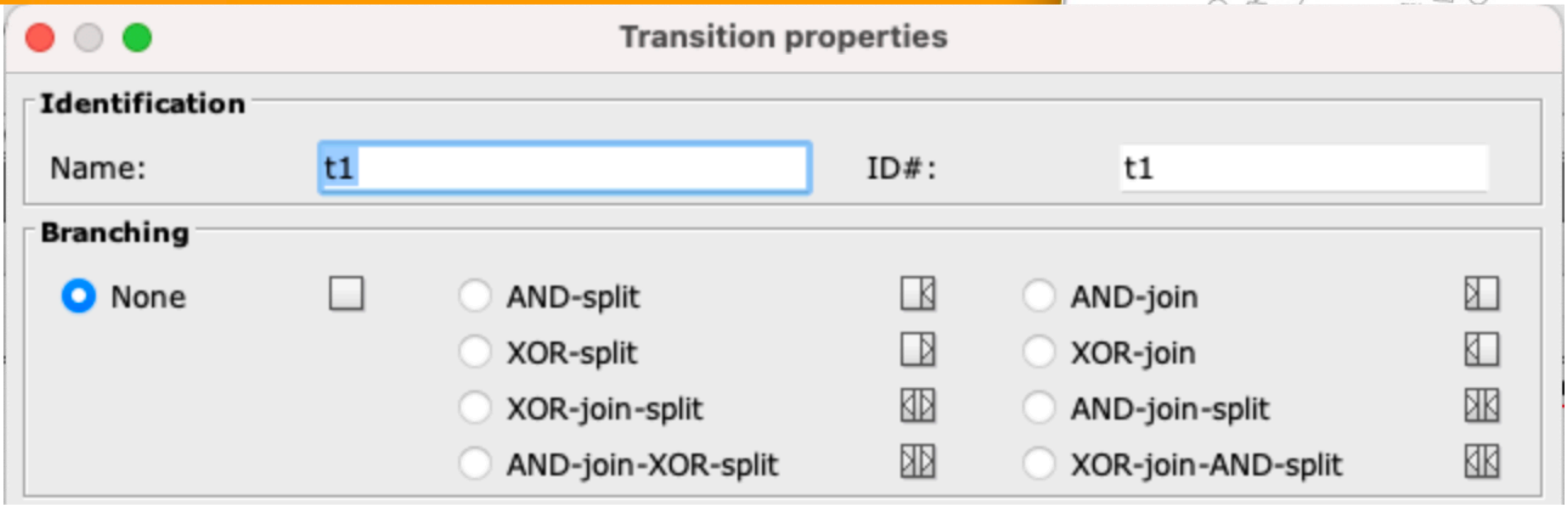
http://woped.dhbw-karlsruhe.de/woped/

WoPeD

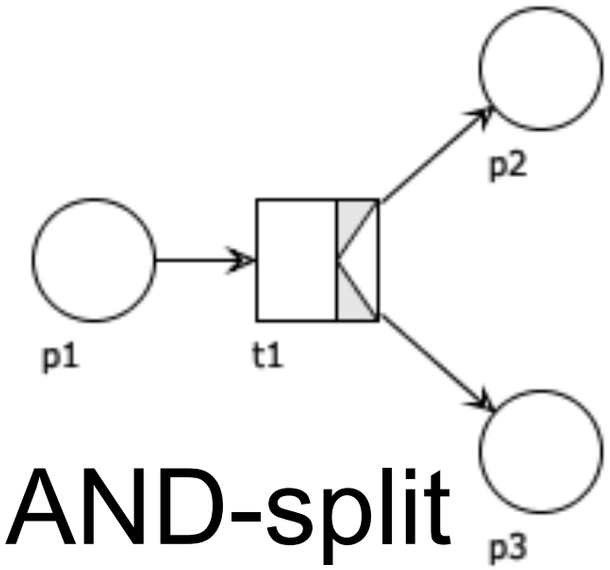
Workflow Petri Net Designer

Download WoPeD at sourceforge!



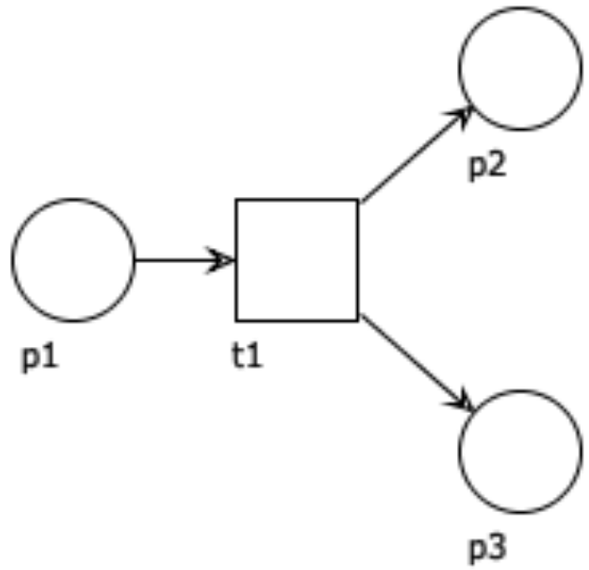


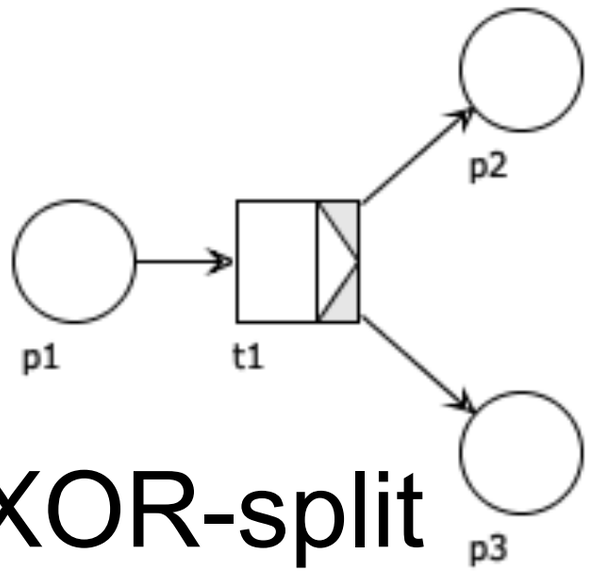
Syntax sugar: split



AND-split

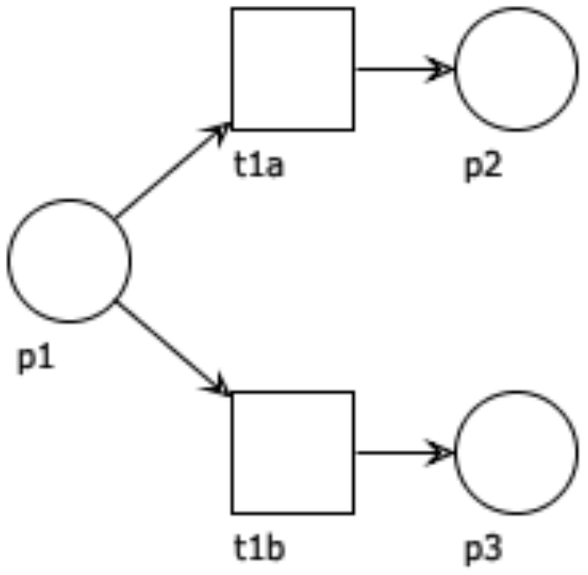
stands for



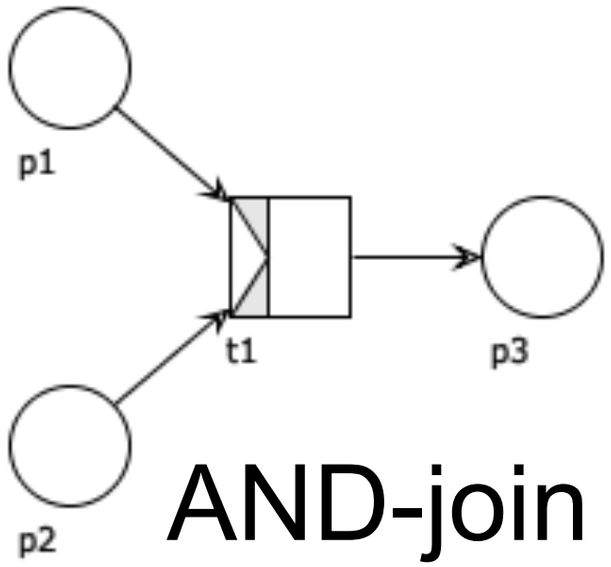


XOR-split

stands for

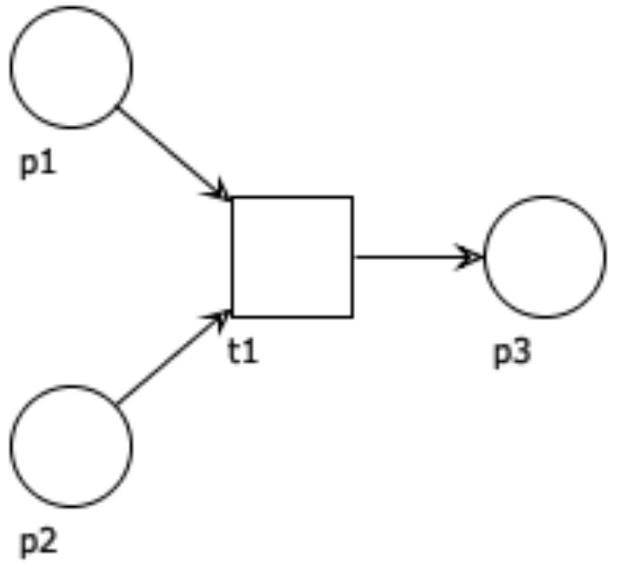


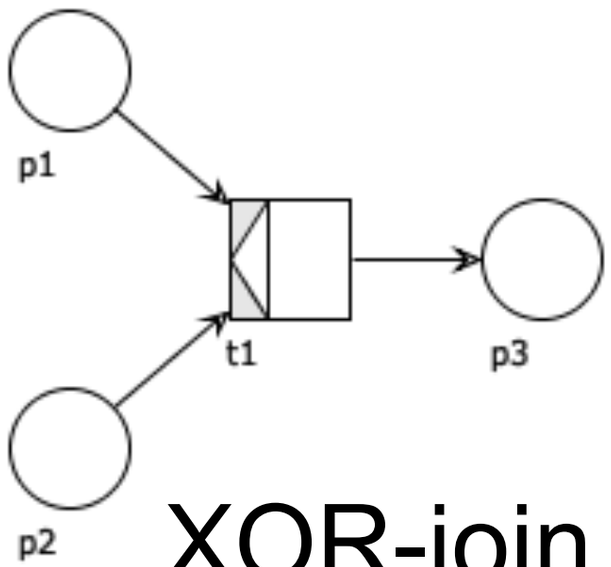
Syntax sugar: join



AND-join

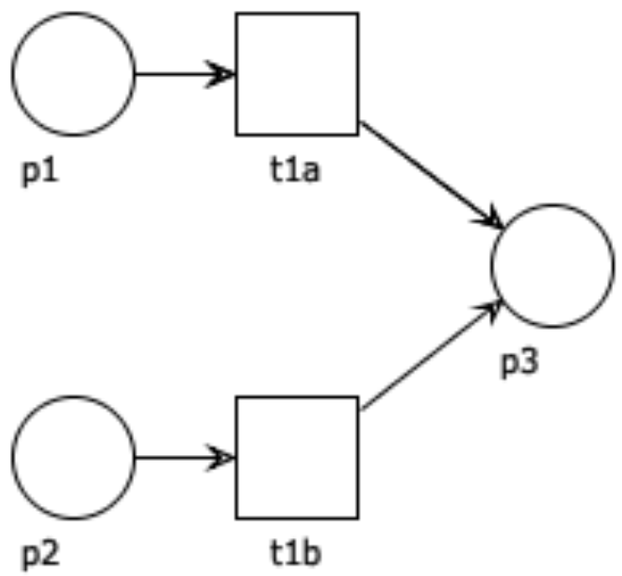
stands for



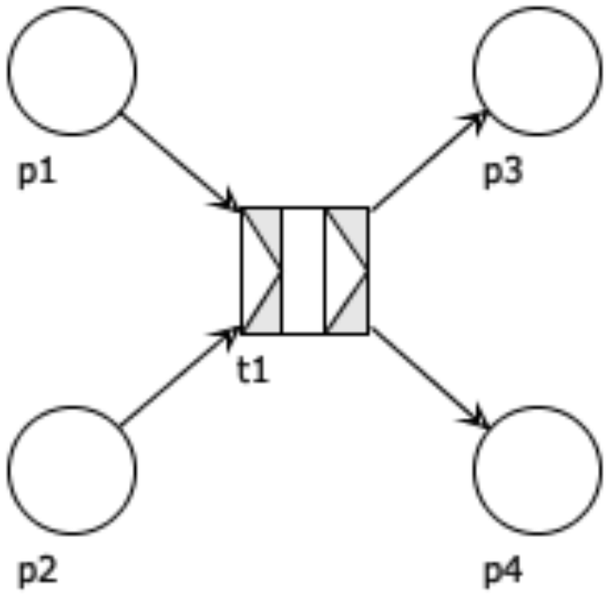


XOR-join

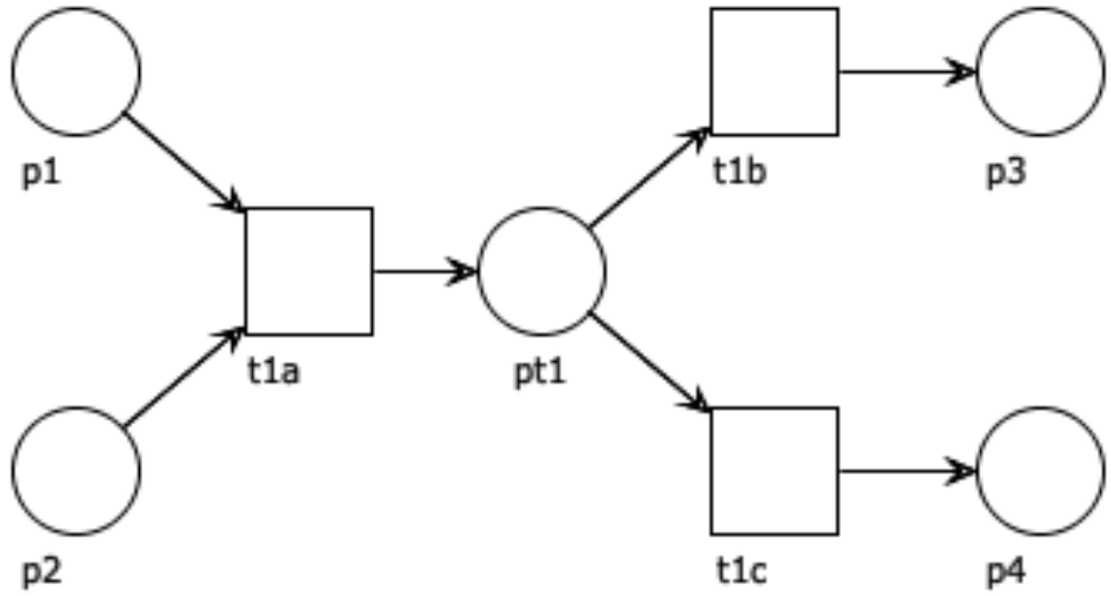
stands for

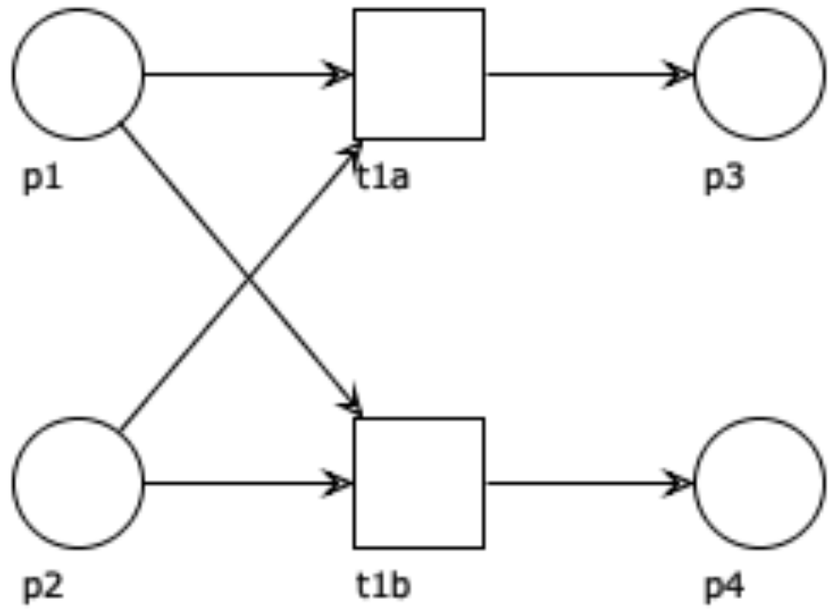


Syntax sugar: any combination is also possible

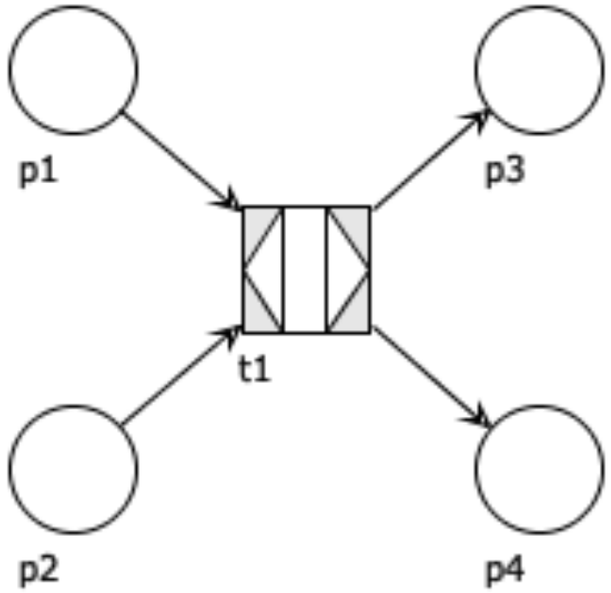


stands for

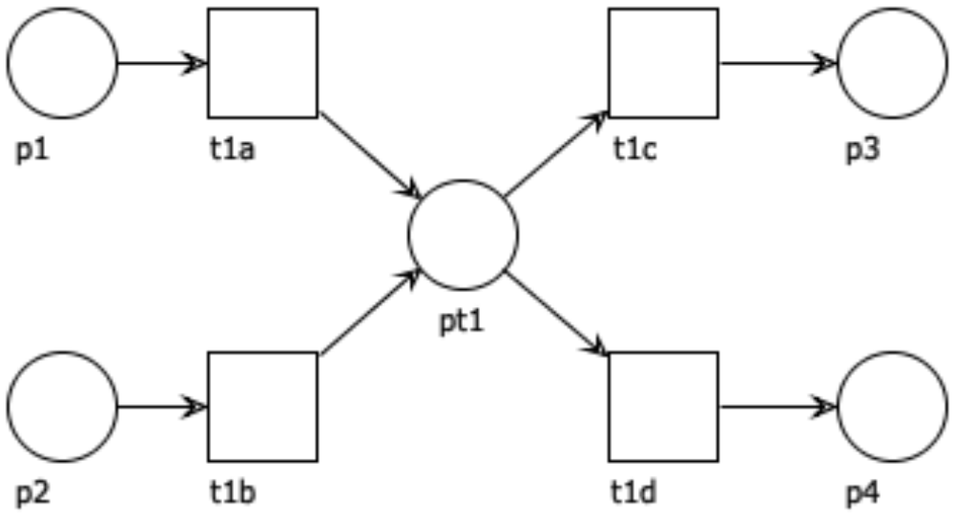


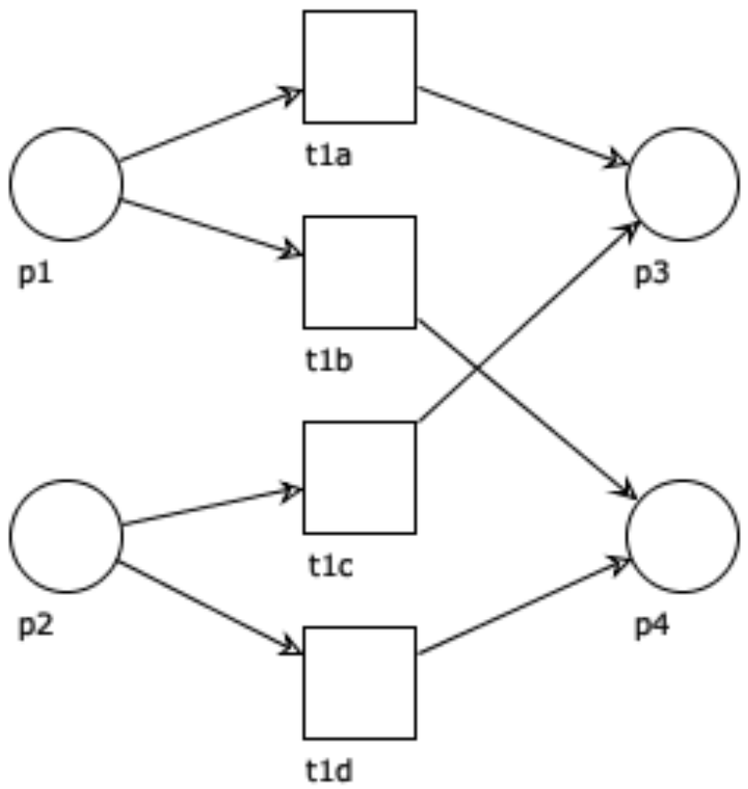


Syntax sugar: any combination is also possible

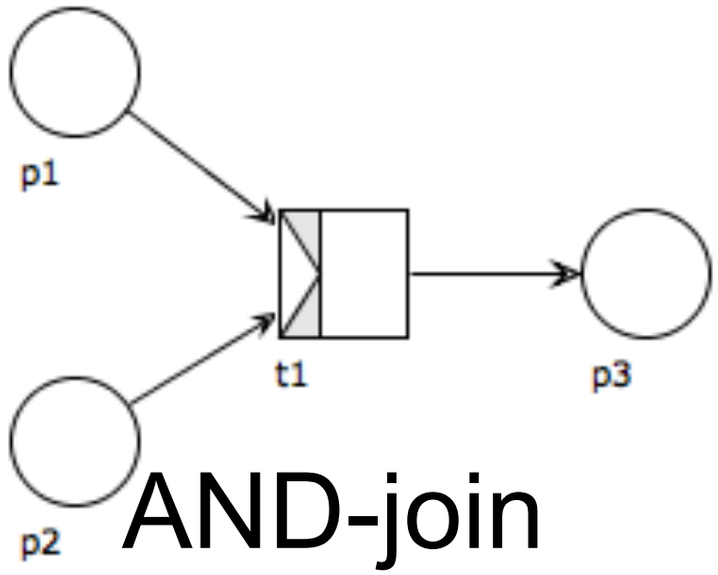


stands for

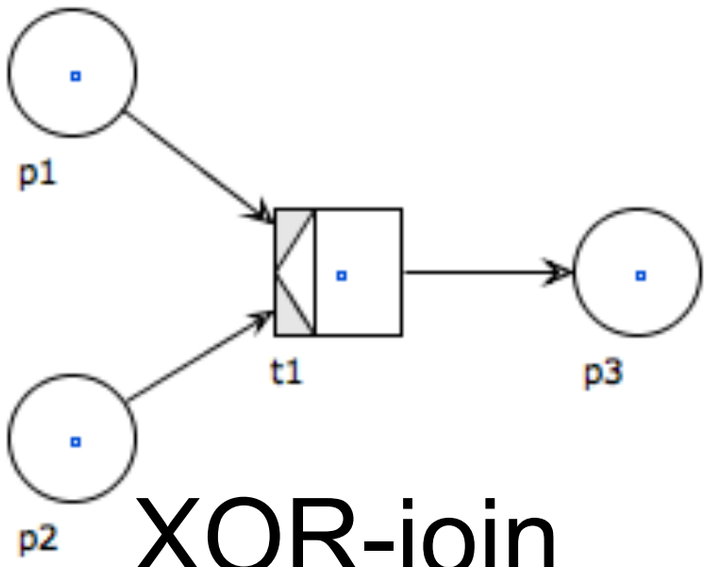




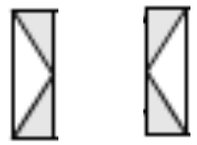
Syntax sugar: a personal note



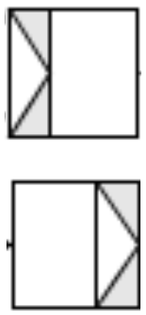
AND-join



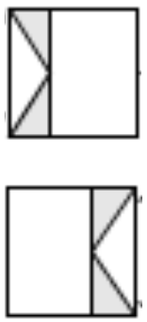
XOR-join



Chosen decorations are too similar!



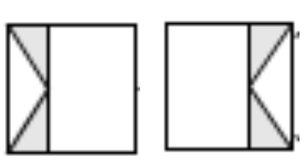
Different meanings if differently placed!

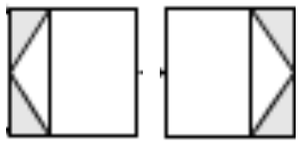


Unnecessary for AND (redundant)!

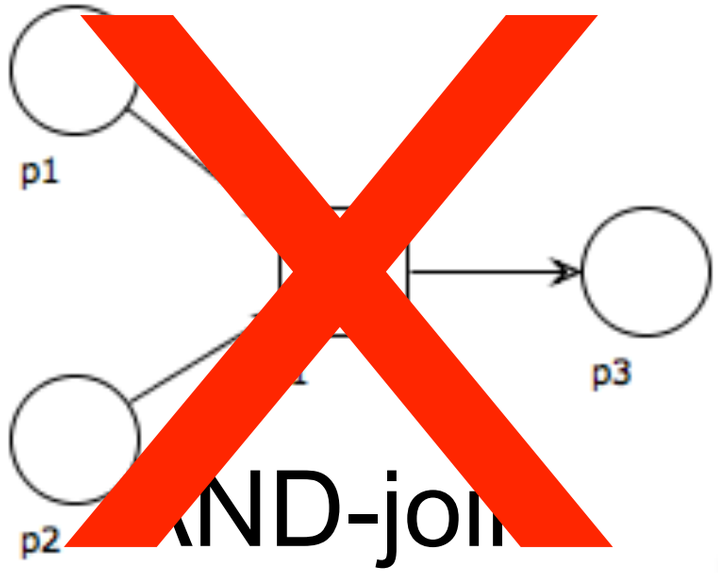
Syntax sugar:
a personal note

Why there? Because of gateways

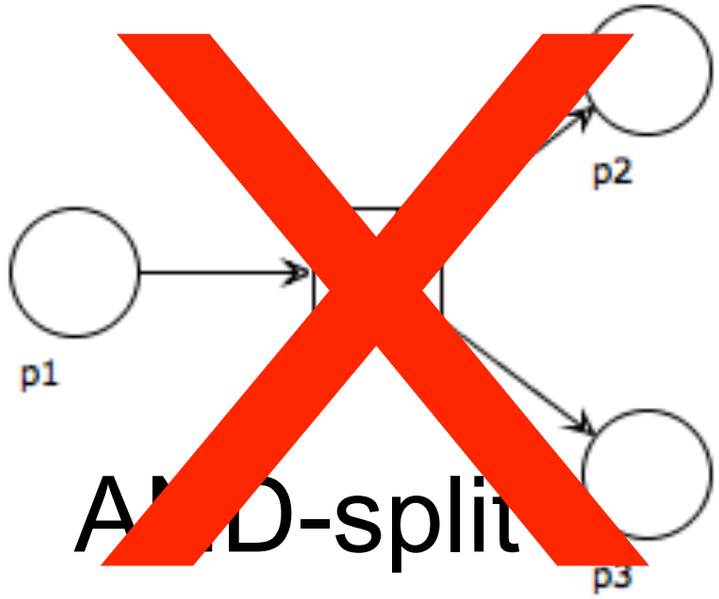




Syntax sugar:
a personal note

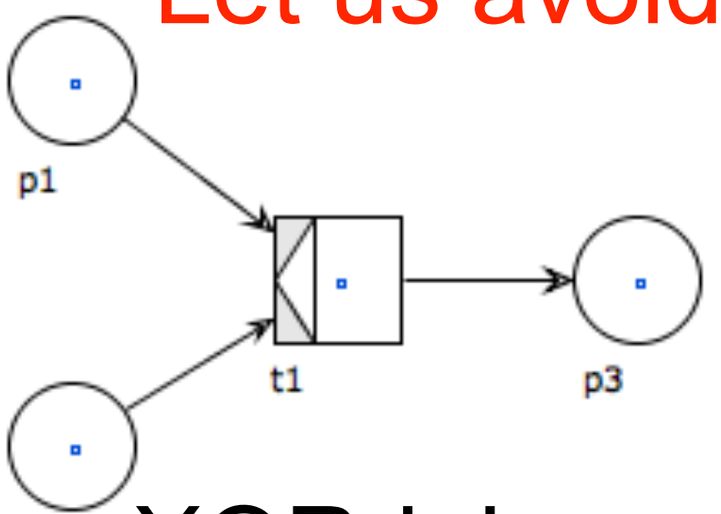


AND-join

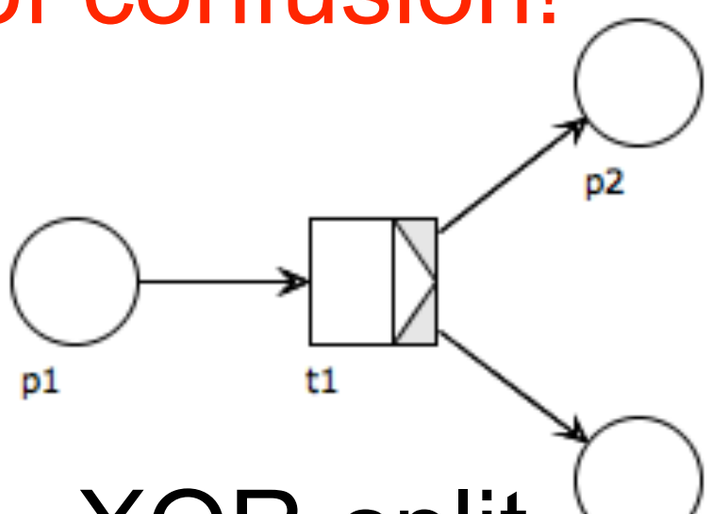


AND-split

Let us avoid any source of confusion!



XOR-join



XOR-split

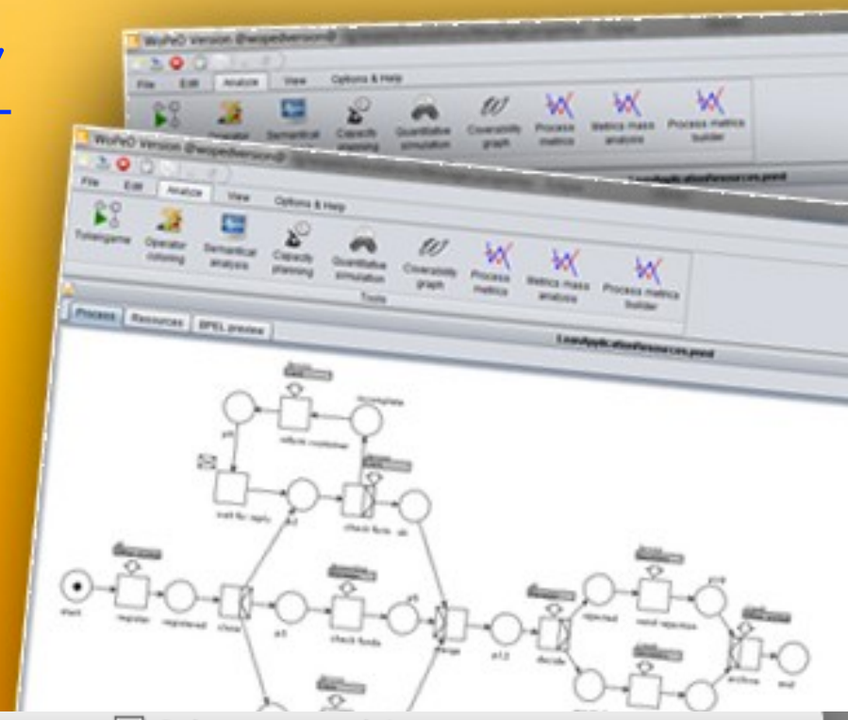
Subprocesses

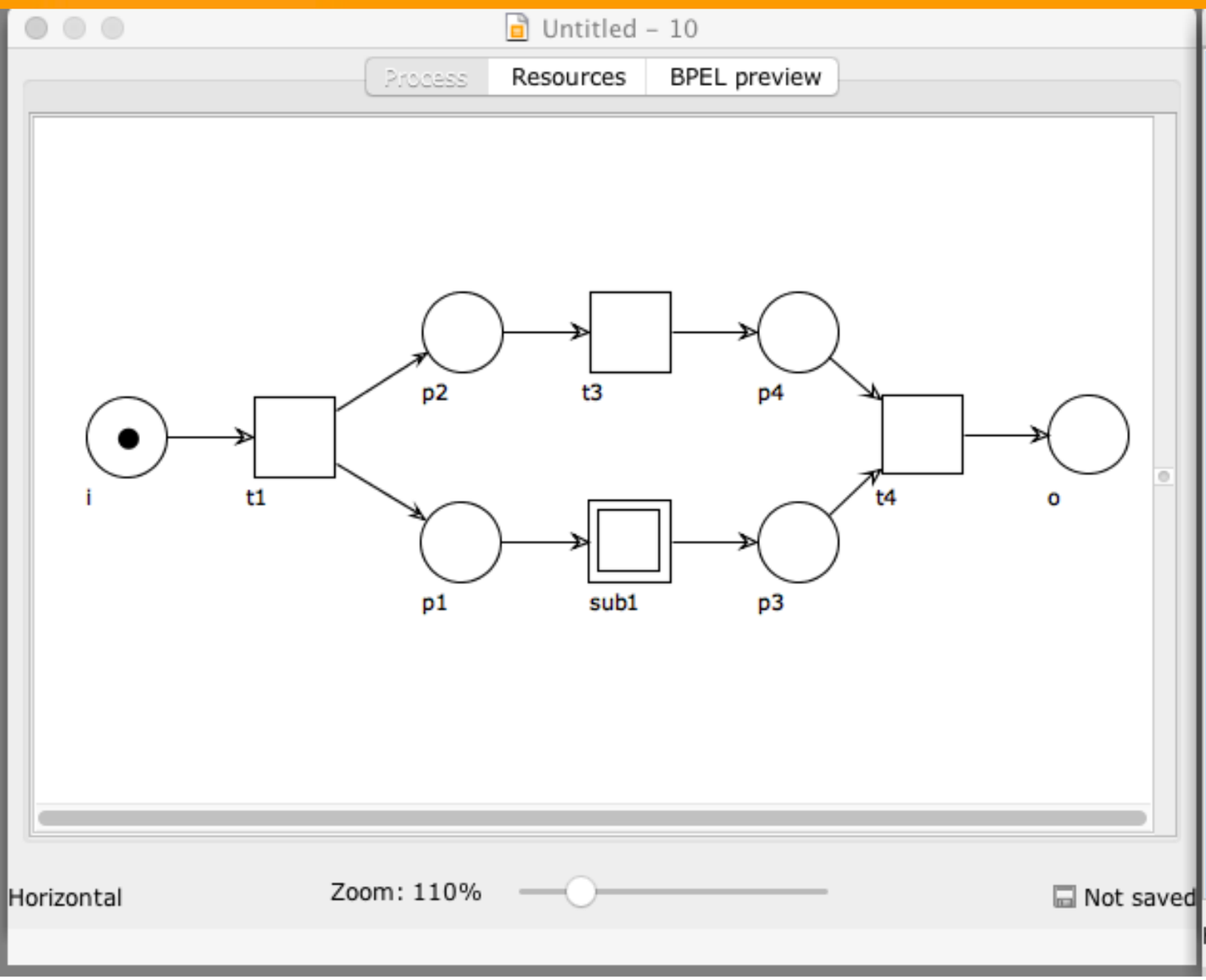
http://woped.dhbw-karlsruhe.de/woped/

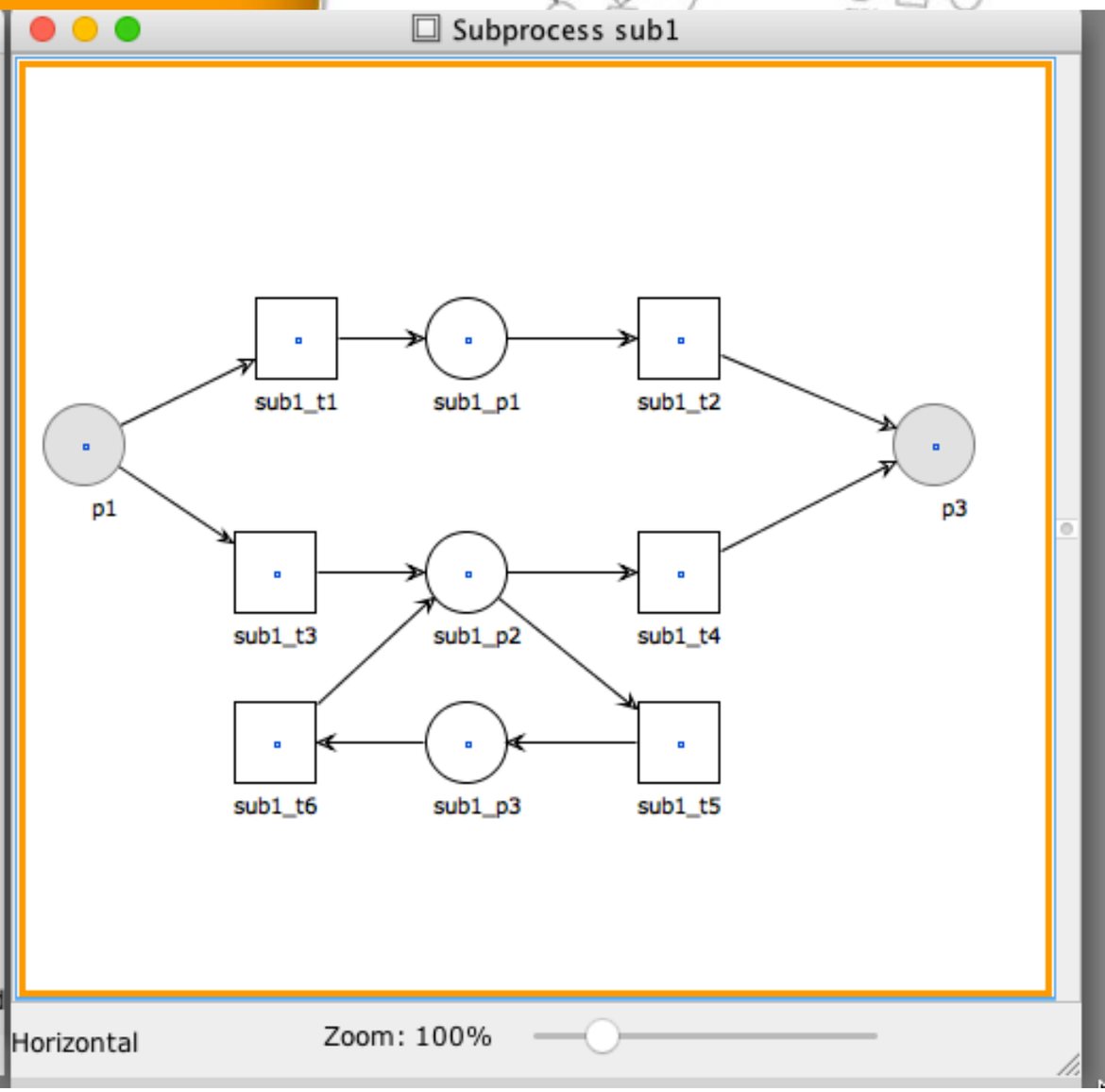
WoPeD

Workflow Petri Net Designer

Download WoPeD at sourceforge!



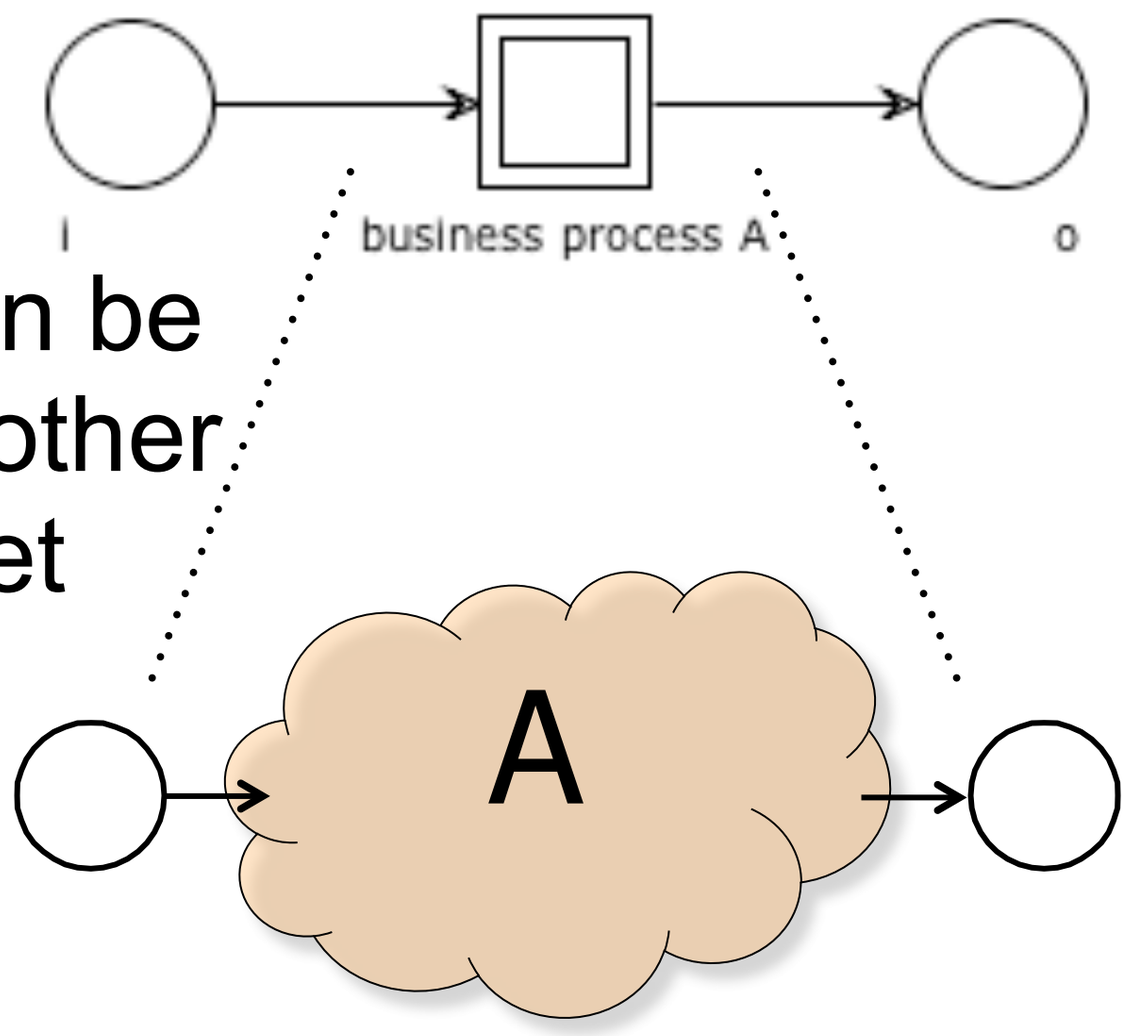




Hierarchical structuring

Uniqueness of entry / exit point facilitate the hierarchical structuring of WF nets

a transition can be realized by another workflow net



Some patterns

Typical control flow aspects

Sequencing

Parallelism (AND-split + AND-join)

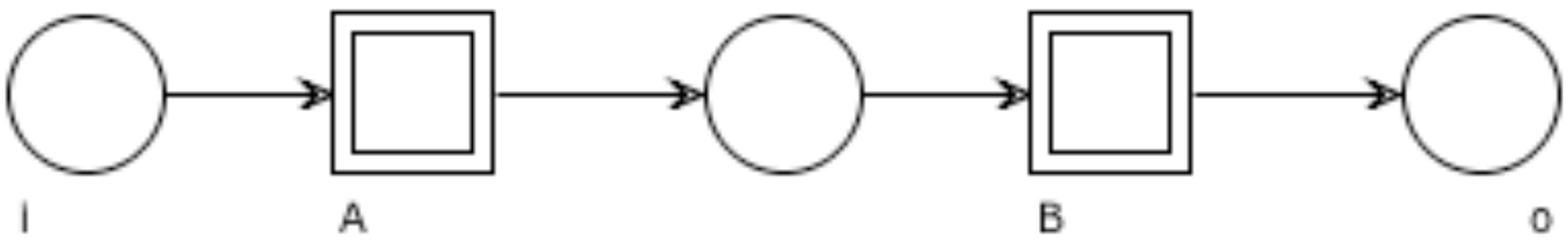
Selection (XOR-split + XOR-join)

Iteration (XOR-join + XOR-split)

Capacity constraints:
Feedback loop
Mutual exclusion
Alternating

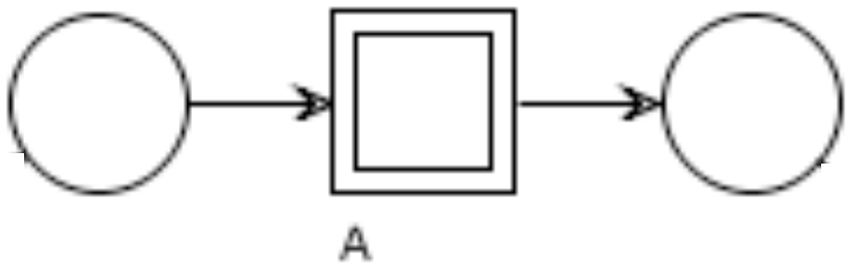
Sequencing

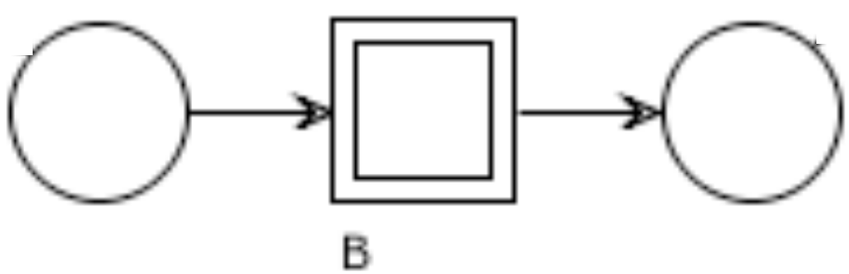
B is executed after A



Parallelism?
(AND-split + AND-join)

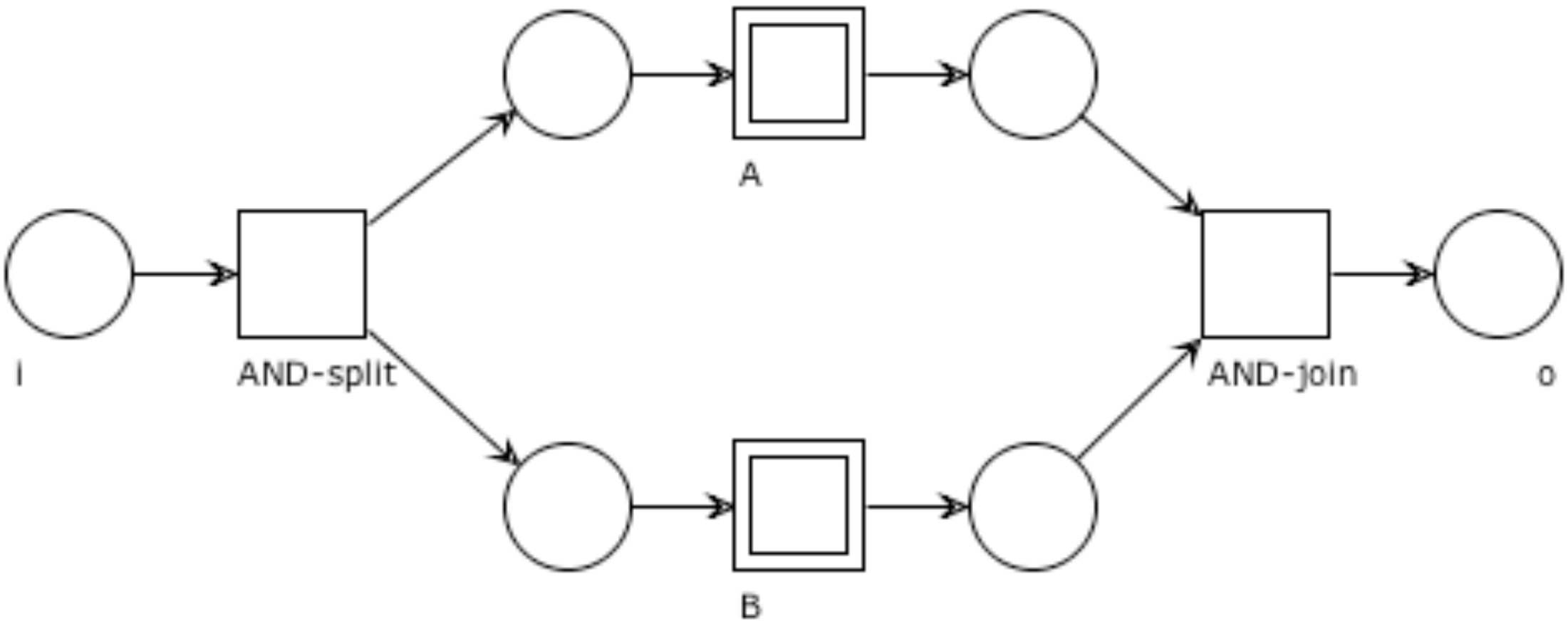
A and B to be executed both, in no particular order





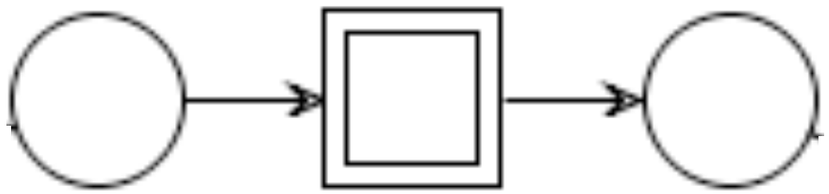
Parallelism (AND-split + AND-join)

A and B are both executed in no particular order

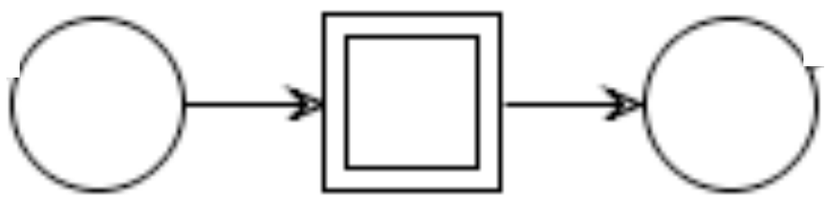


Explicit choice?
(XOR-split + XOR-join)

Either A or B is executed (choice is **explicit**)



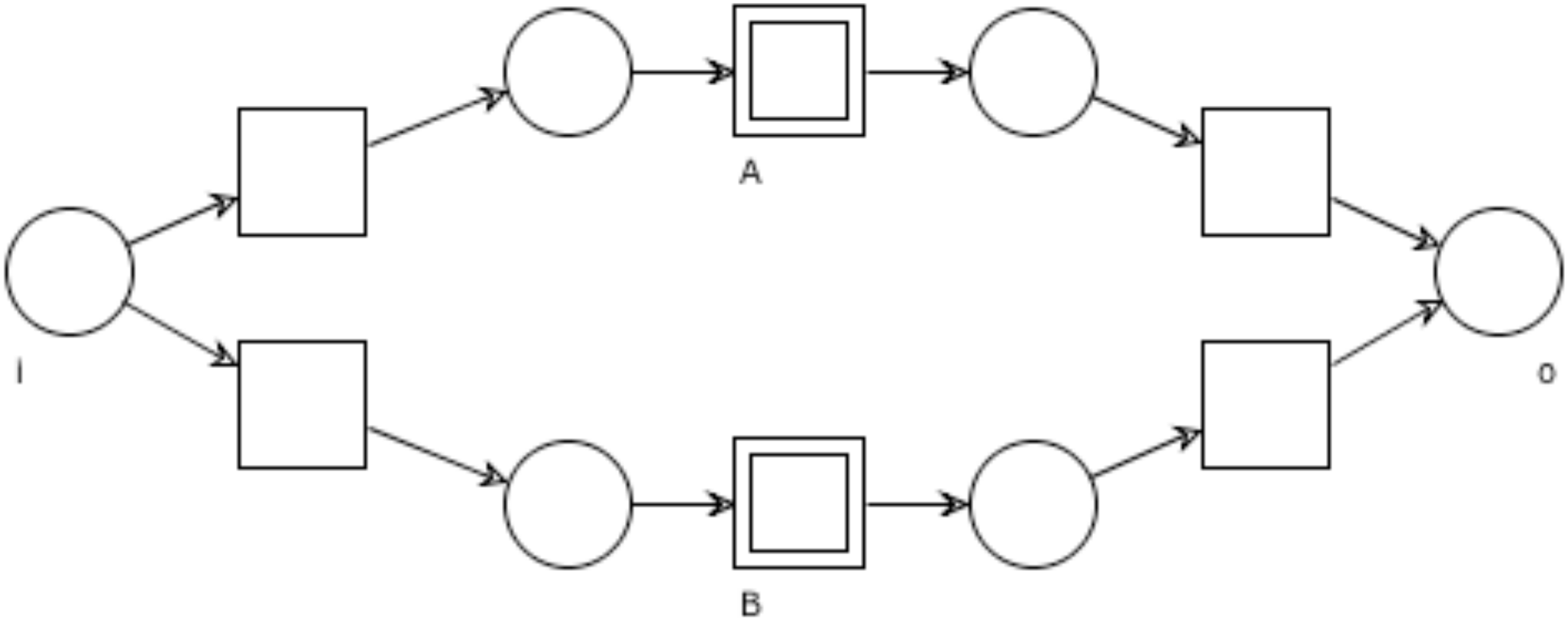
A



B

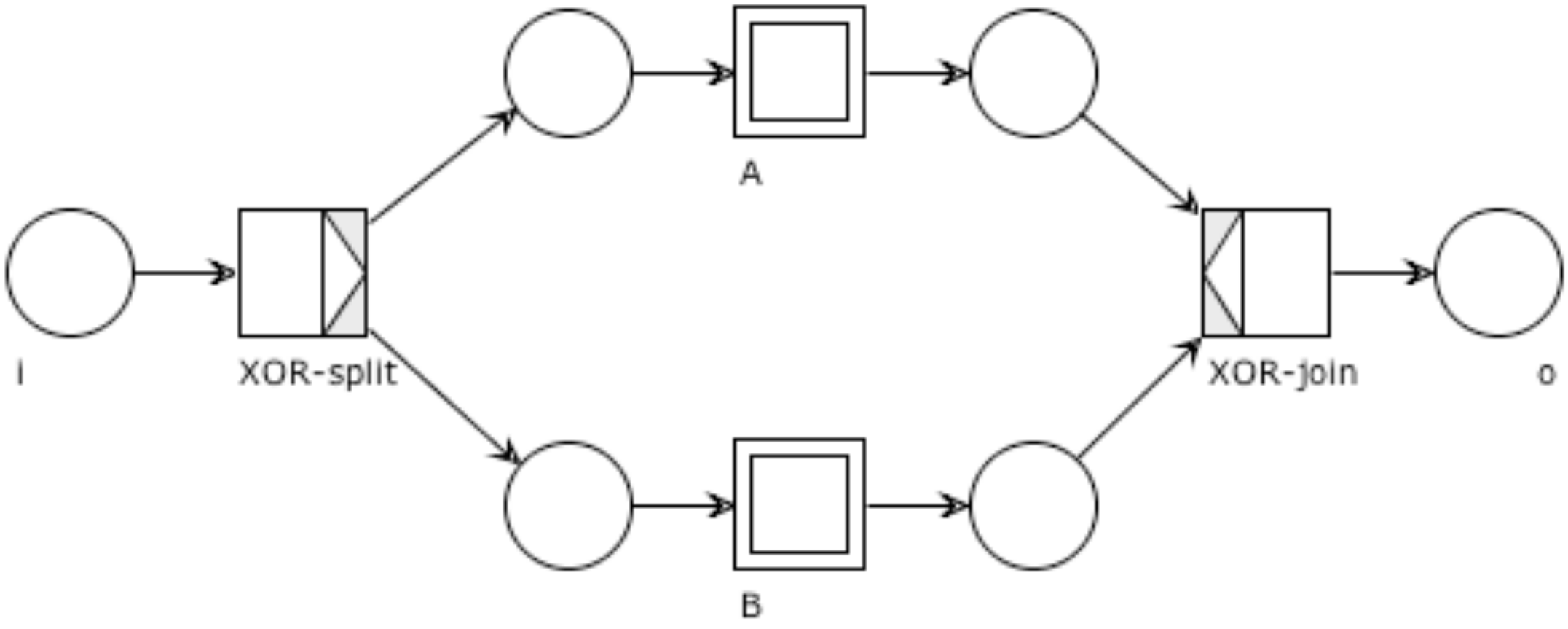
Explicit choice (XOR-split + XOR-join)

Either A or B is executed (choice is **explicit**)



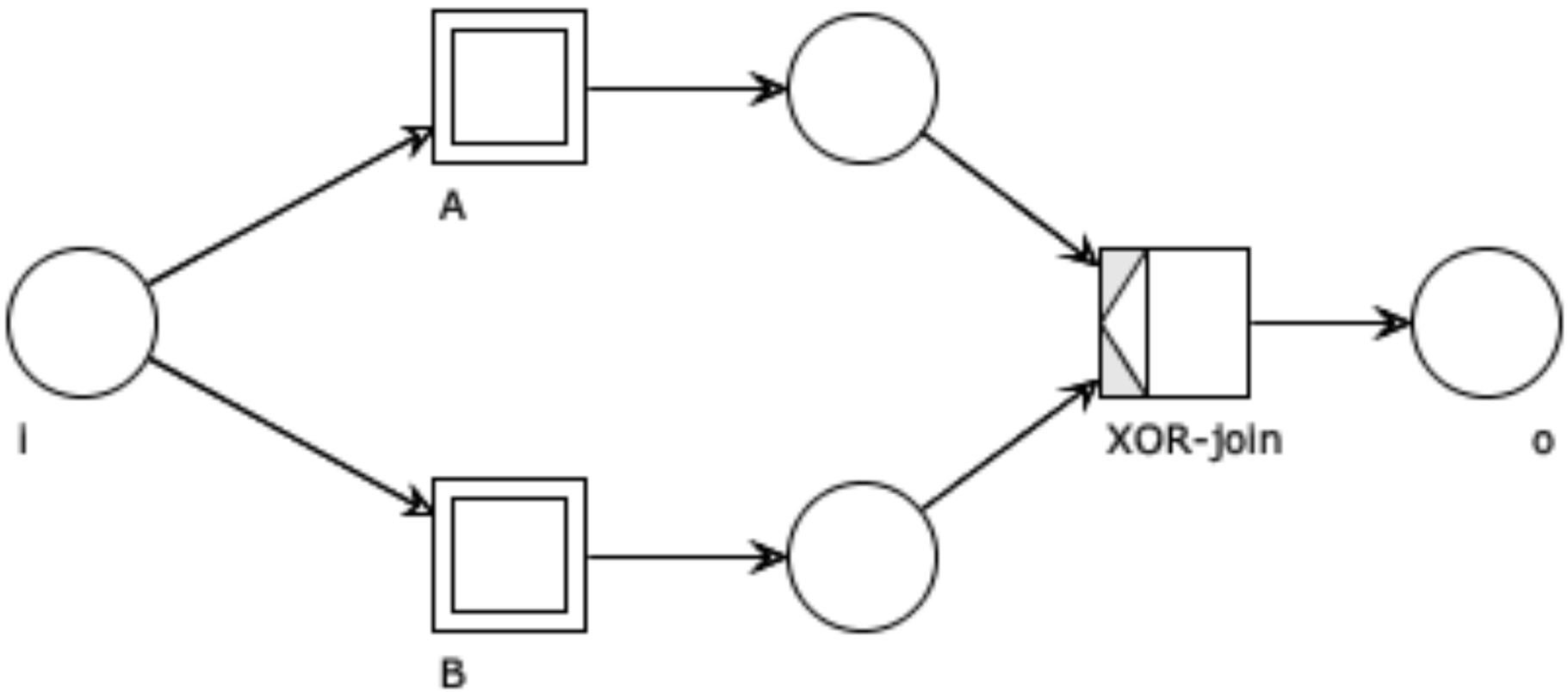
Explicit choice ("sugared" version)

Decorated version



Deferred choice

Either A or B is executed (choice is **implicit**)

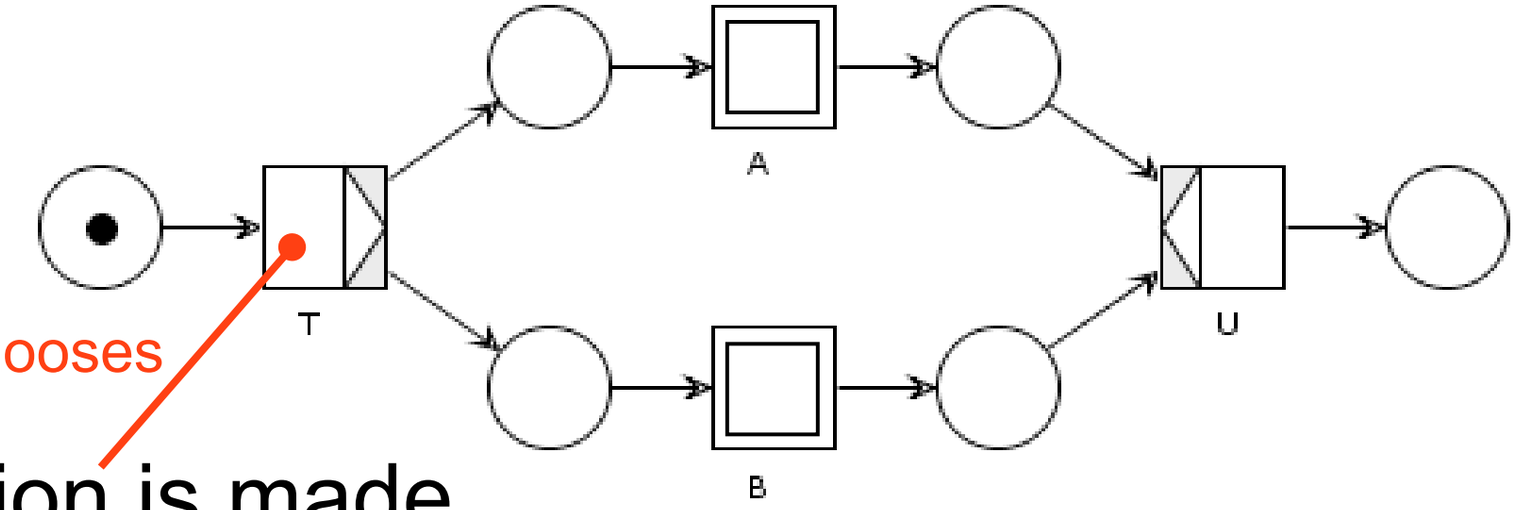


Remember

BPMN elements

XOR

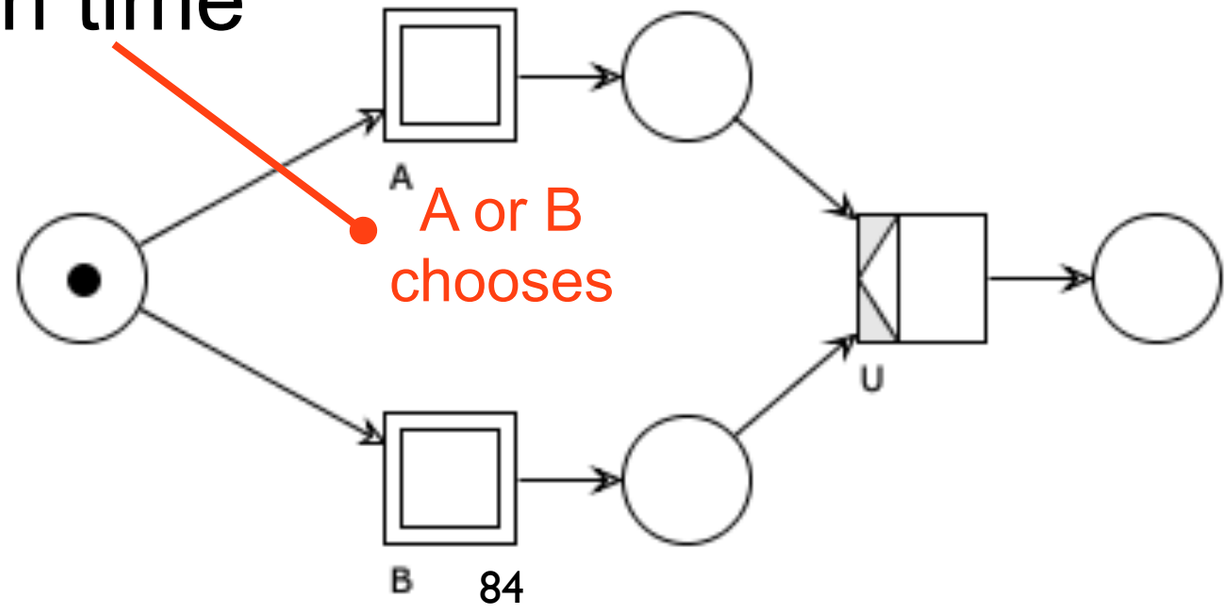
Explicit choice ≠ Implicit choice



T chooses

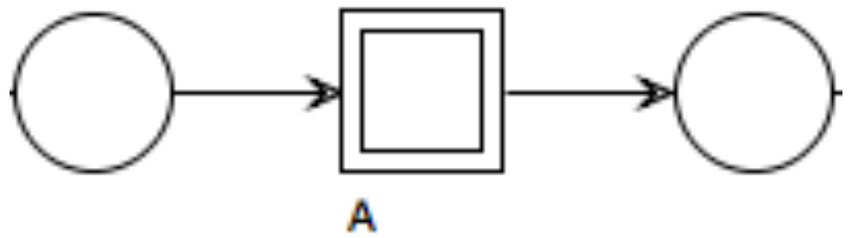
The decision is made at different points in time

event based



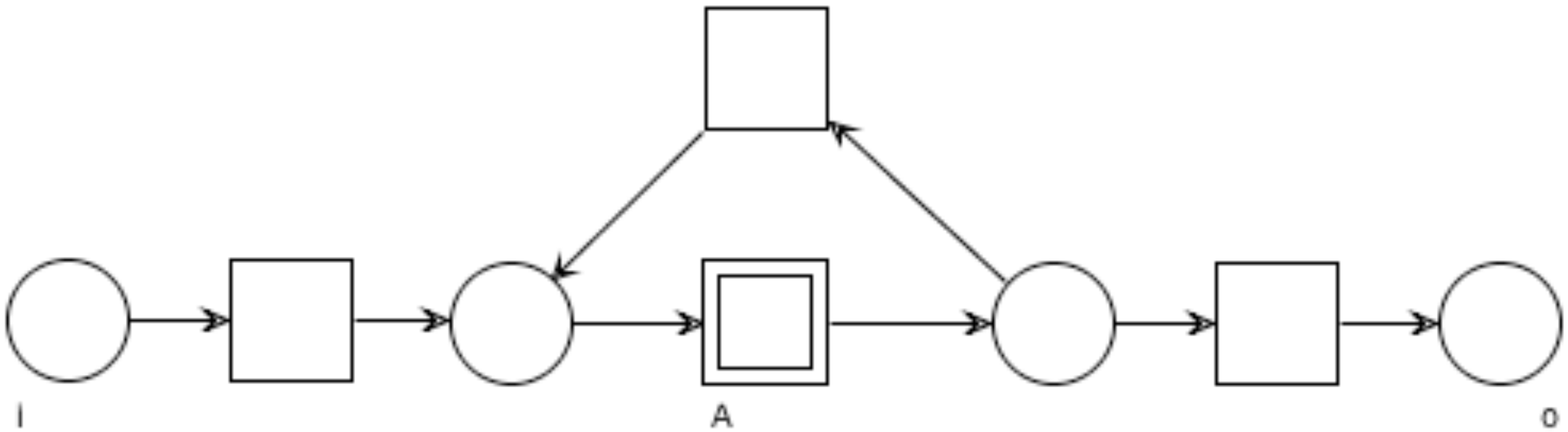
Iteration
(one or more times)

A to be executed 1 or more times



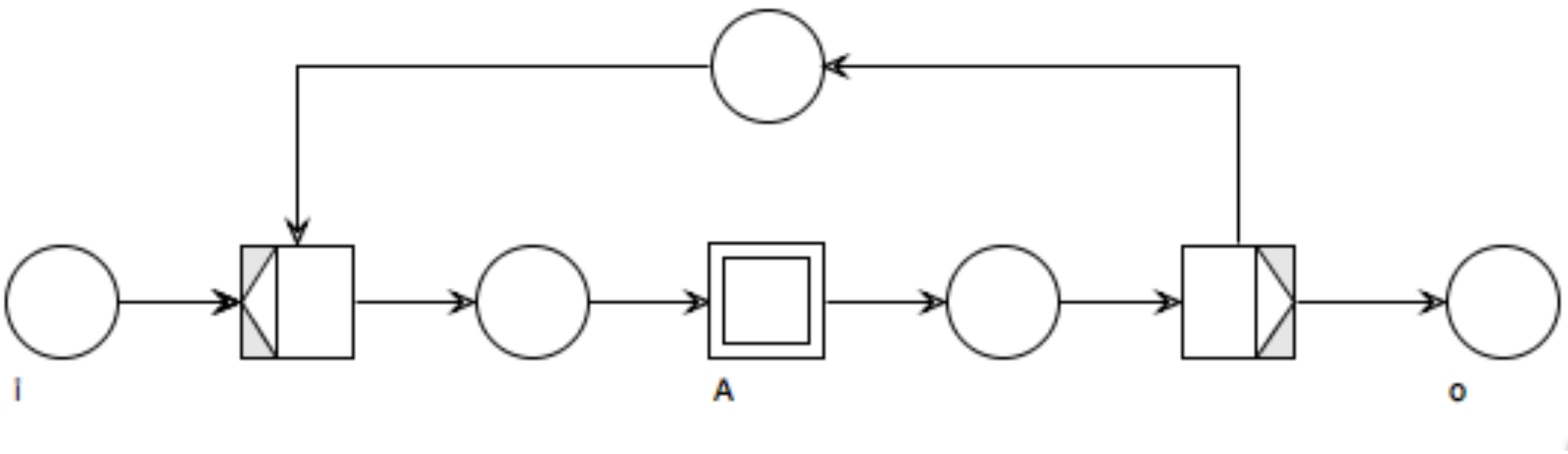
Iteration
(one or more times)

A is executed 1 or more times



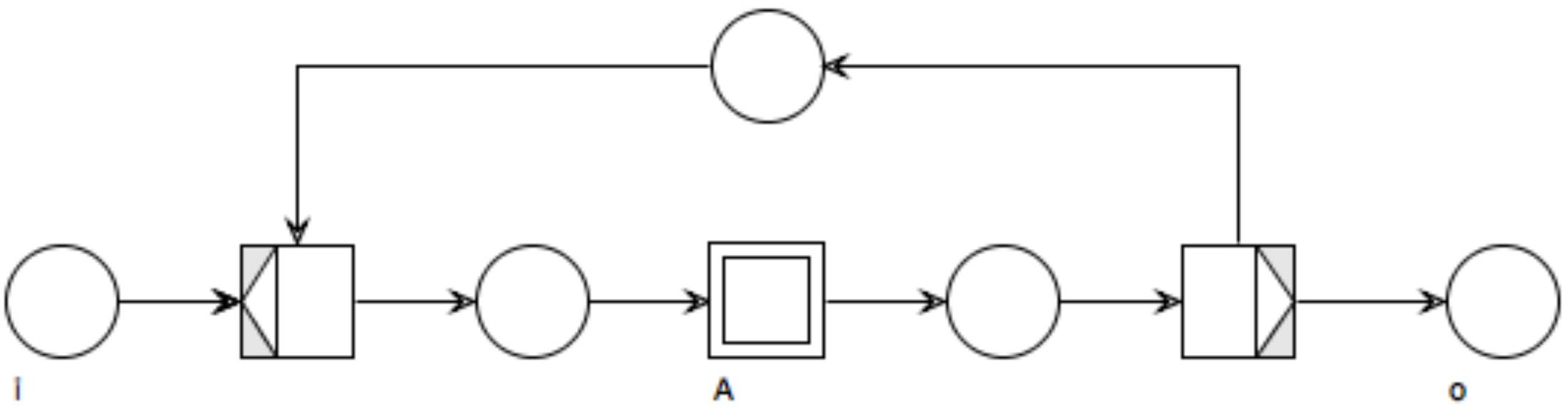
One-or-more iteration ("sugared" version)

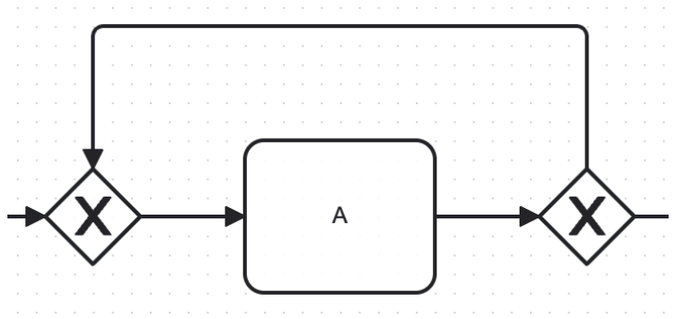
Decorated version



One-or-more iteration BPMN-like version

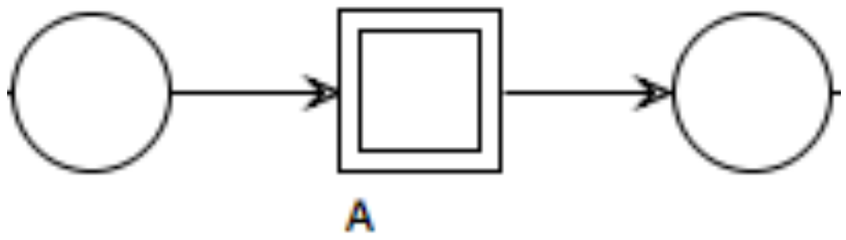
Decorated version





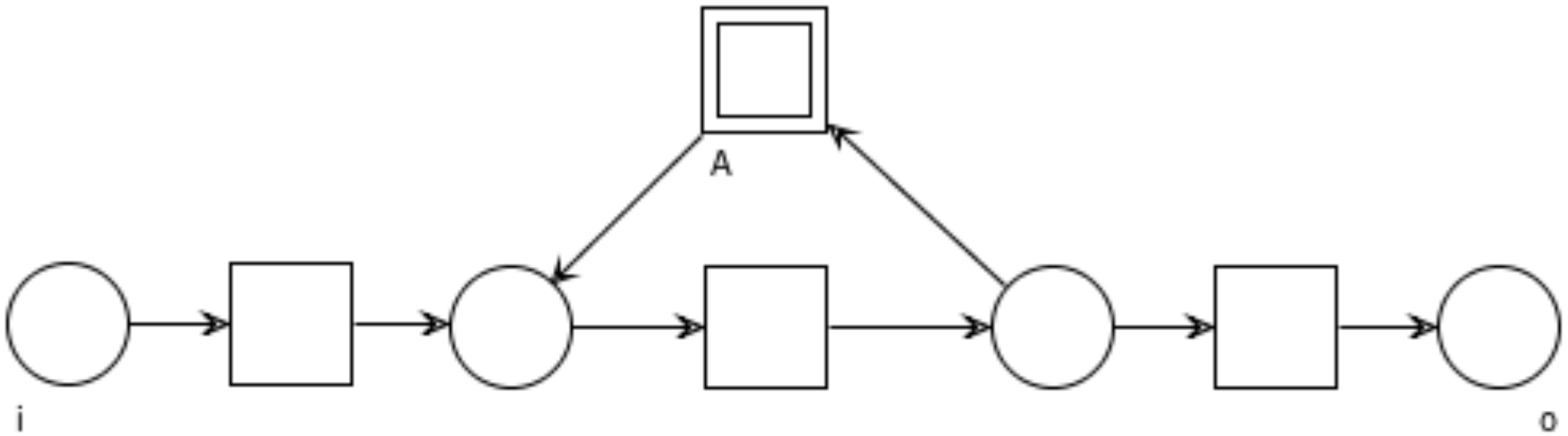
Iteration
(zero or more times)

A to be executed 0 or more times



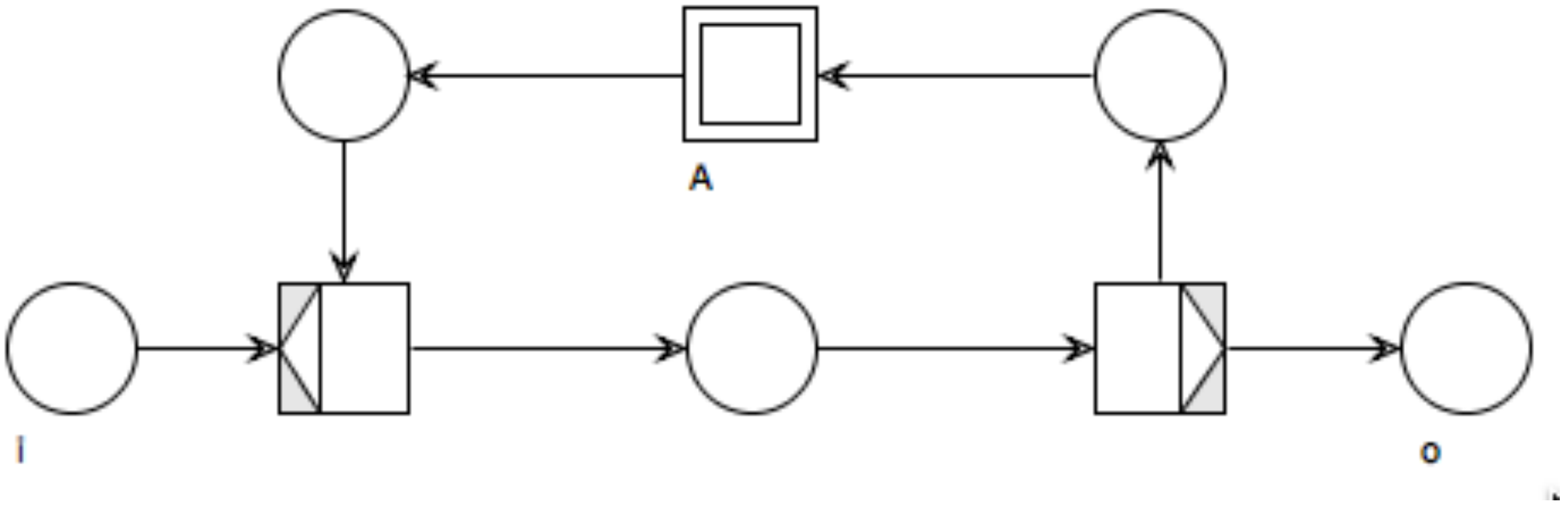
Iteration
(zero or more times)

A is executed 0 or more times



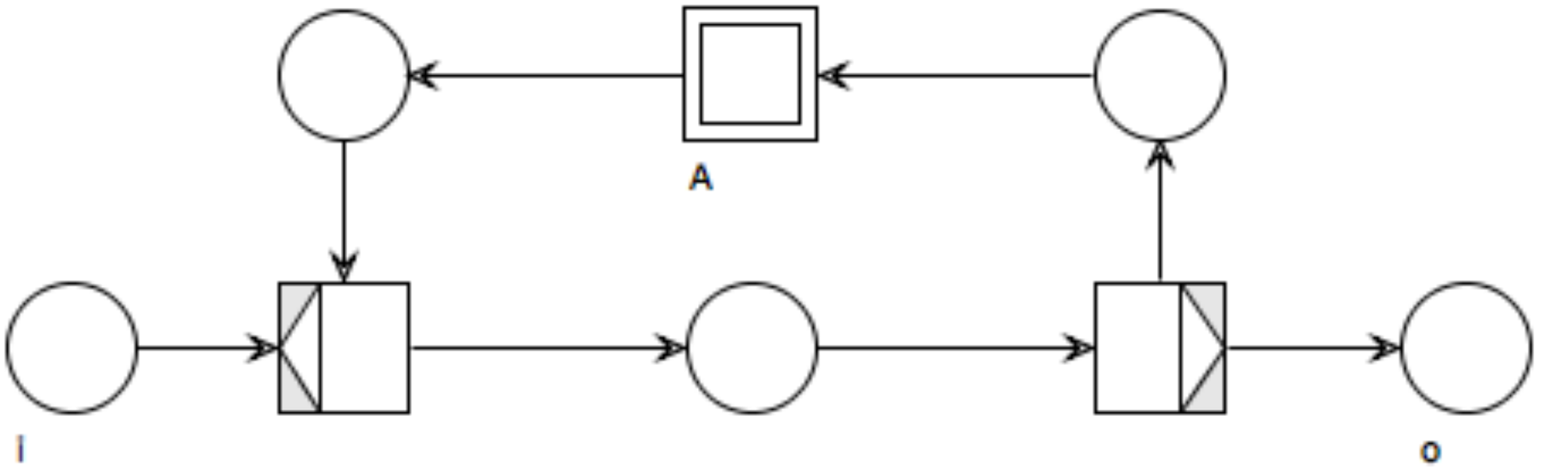
Zero-or-more iteration ("sugared" version)

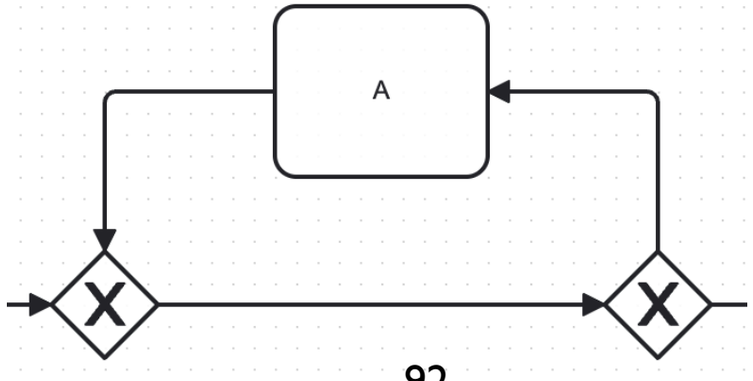
Decorated version



Zero-or-more iteration
BPMN-like version

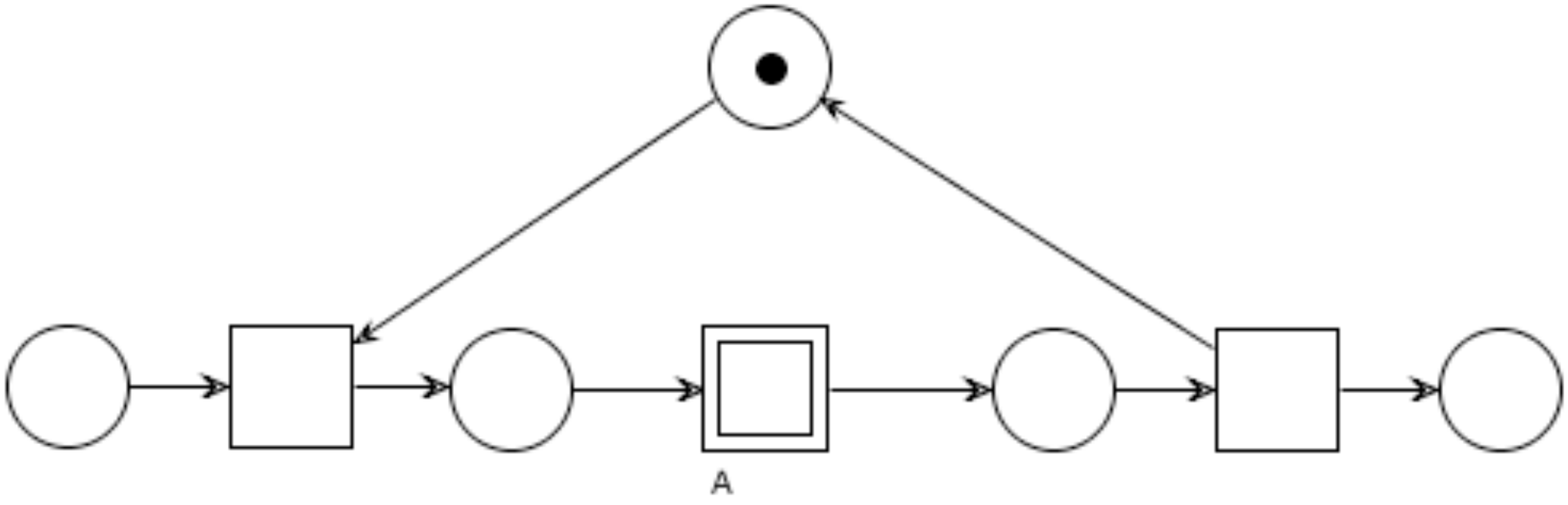
Decorated version





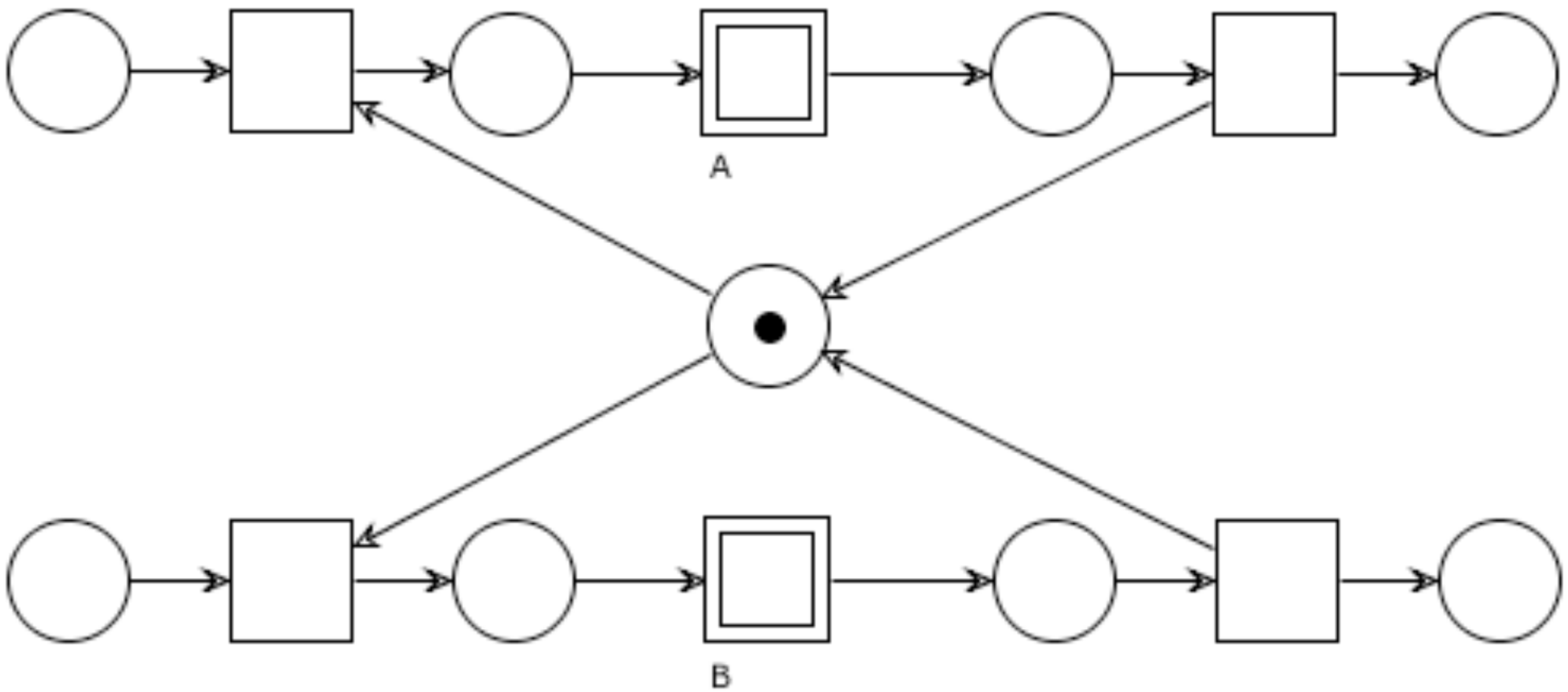
One serve per time

Multiple activations are handled one by one



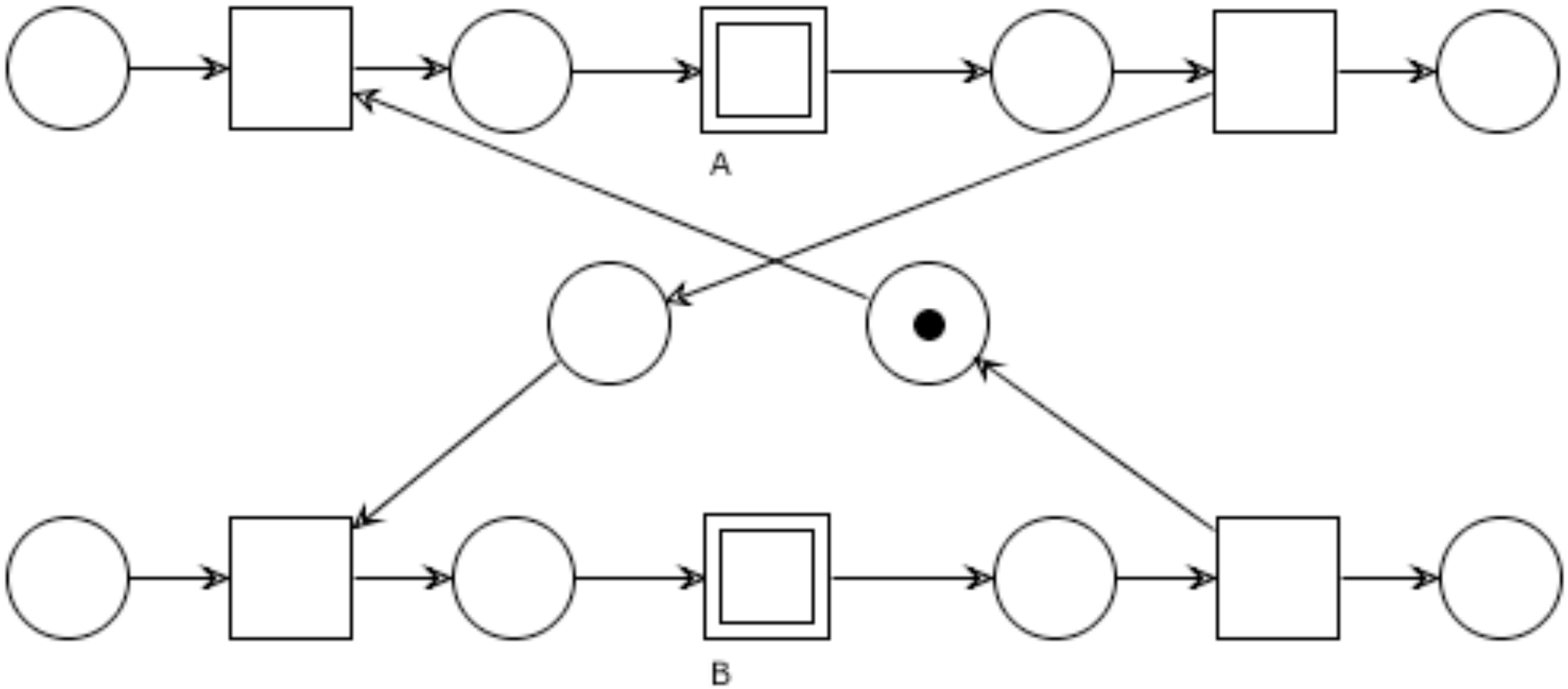
Mutual exclusion

A and B cannot execute concurrently

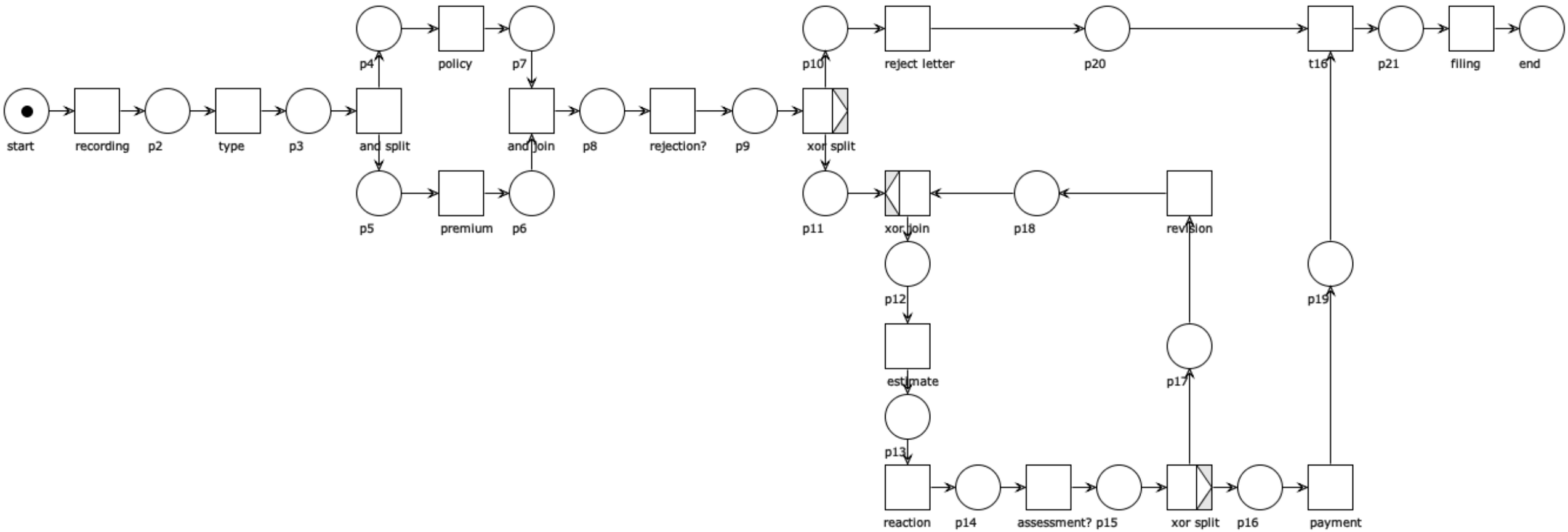


Alternation

A and B execute one time each (A first)

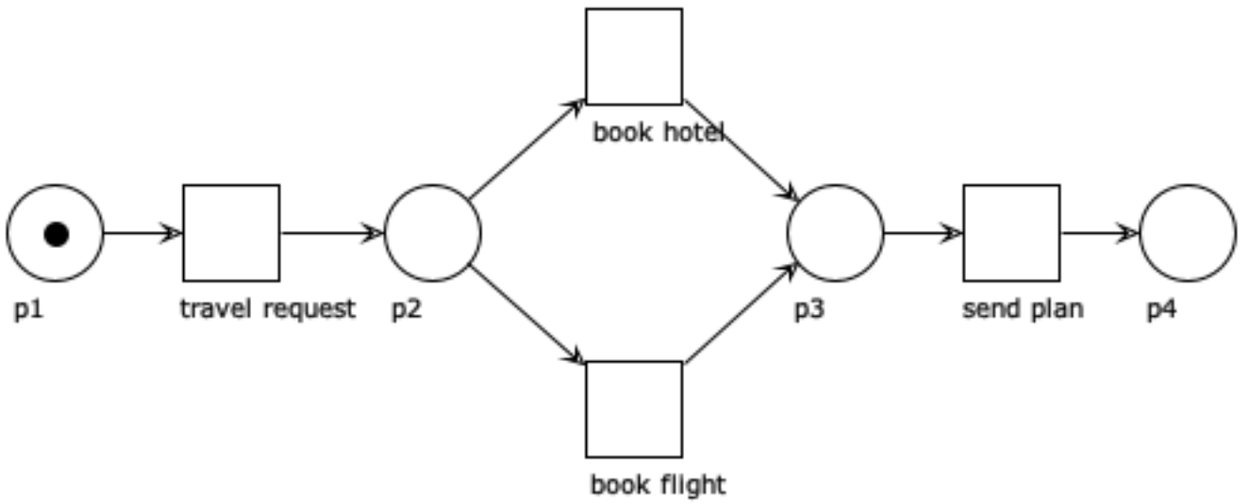


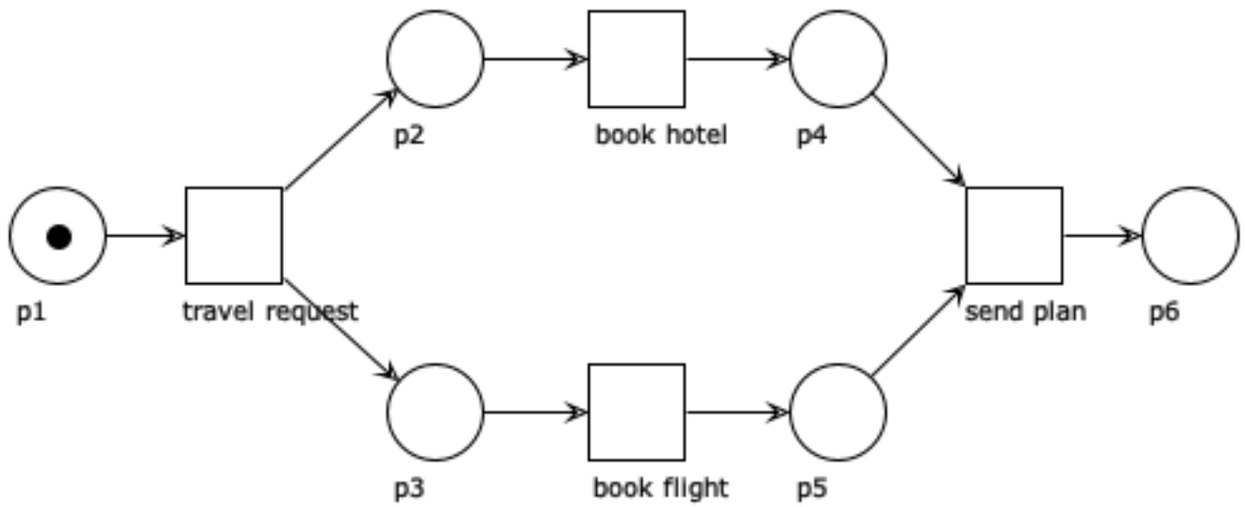
Which patterns?



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

Question time





Which model looks more reasonable?

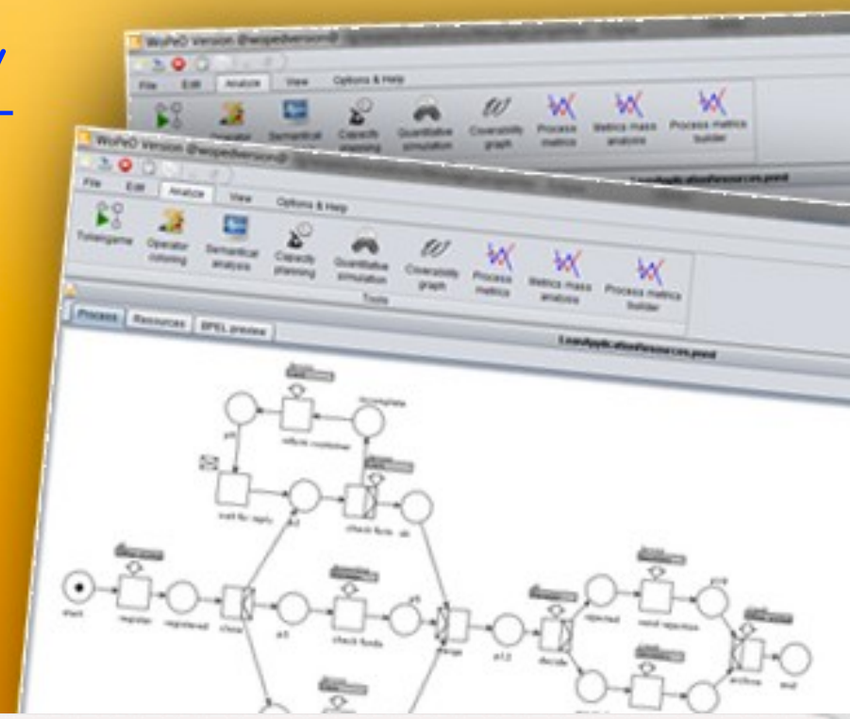
Triggers

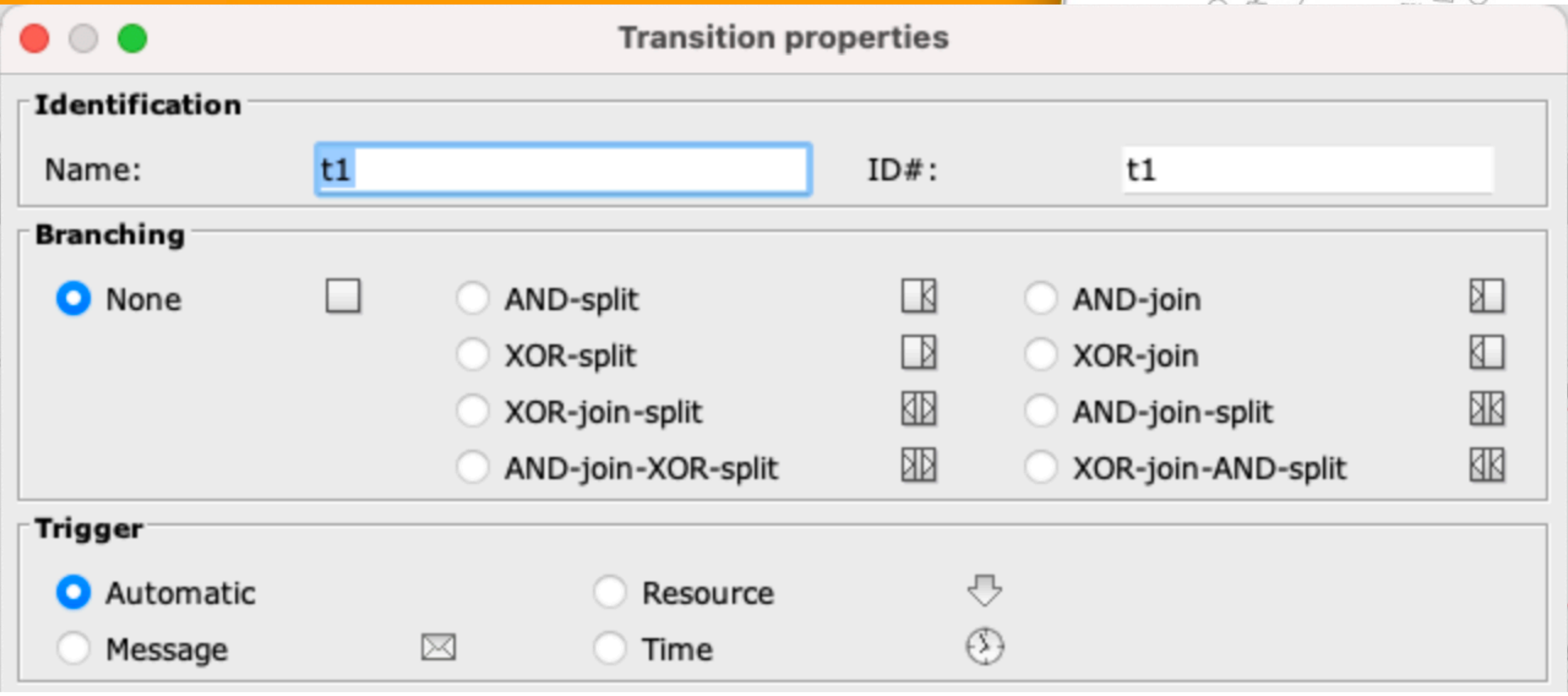
http://woped.dhbw-karlsruhe.de/woped/

WoPeD

Workflow Petri Net Designer

Download WoPeD at sourceforge!





Triggers

Execution constraints can depend on the environment in which processes are enacted.

In workflow nets, transitions can be decorated with the information on who (or what) is responsible for the "firing" of that task.

Such annotations are called **triggers**

Triggers

Triggers can be:

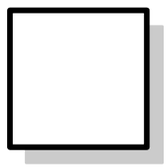
a human interaction

the receipt of a message

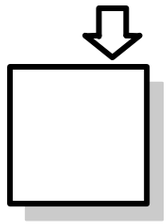
the expiration of a time-out

Transitions with no trigger can fire automatically

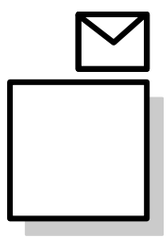
Symbols for triggers



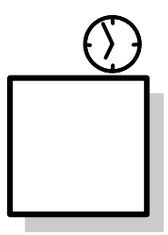
Automatic Trigger: Task enacted automatically



User Trigger: A human user takes initiative and starts activity

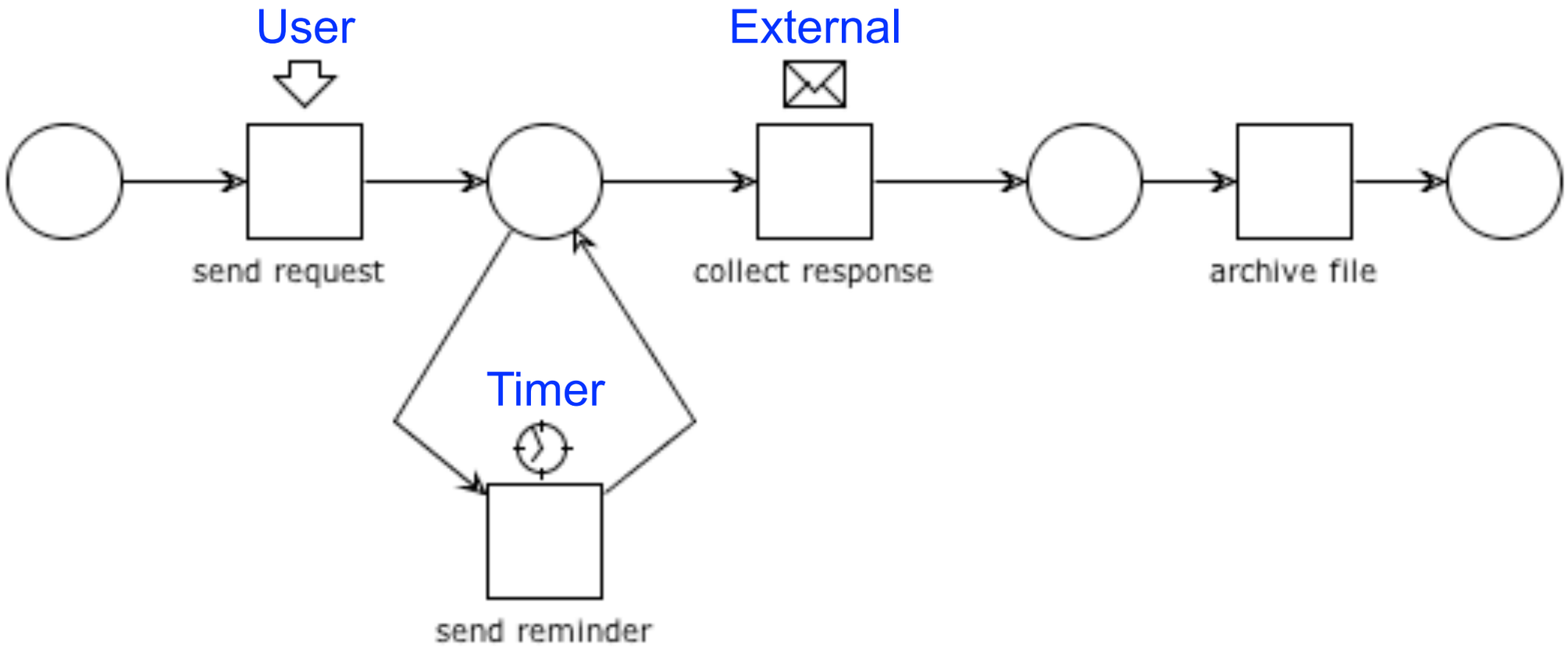


External Trigger: External event required to start activity

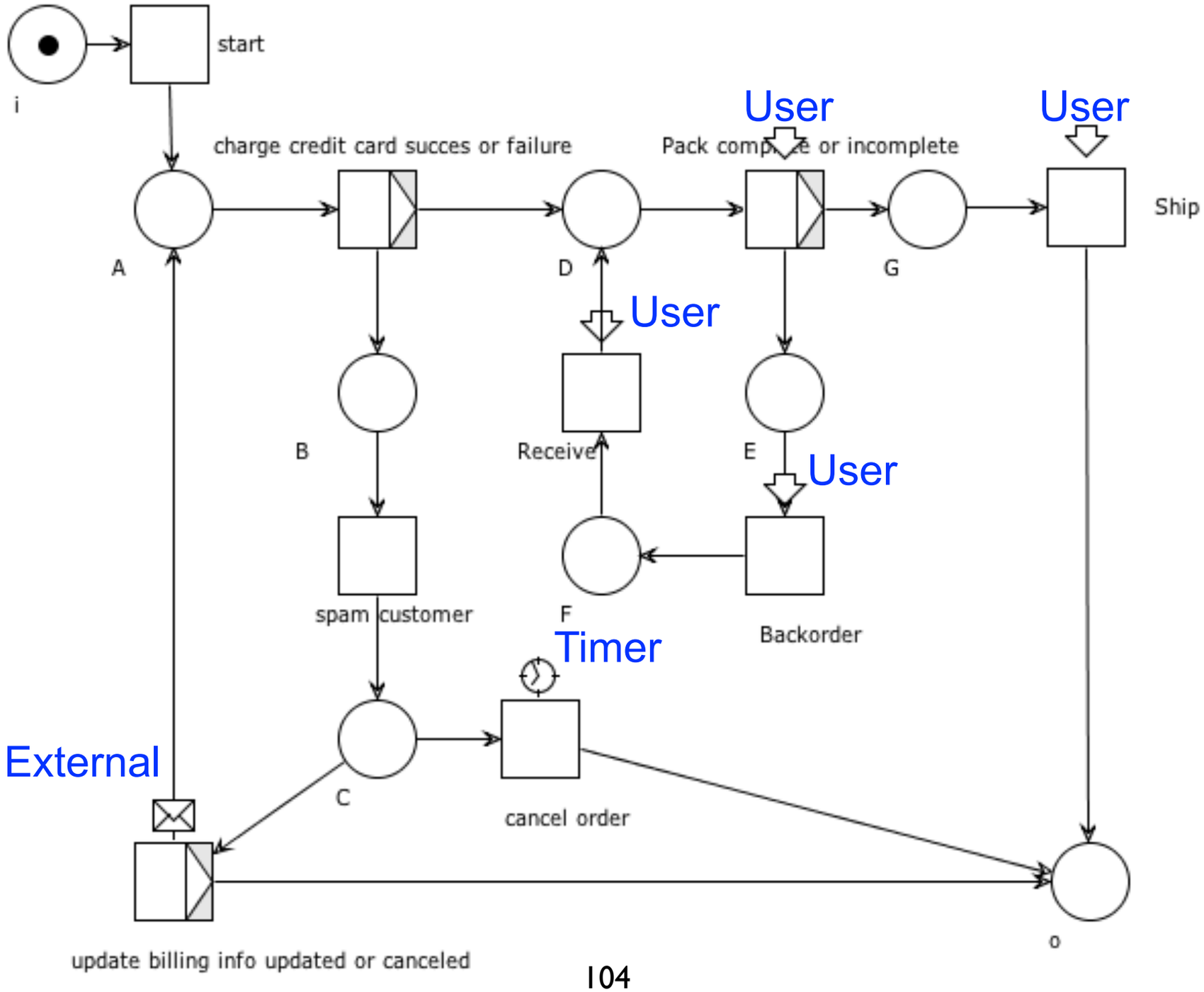


Time Trigger: Activity started when timer elapses

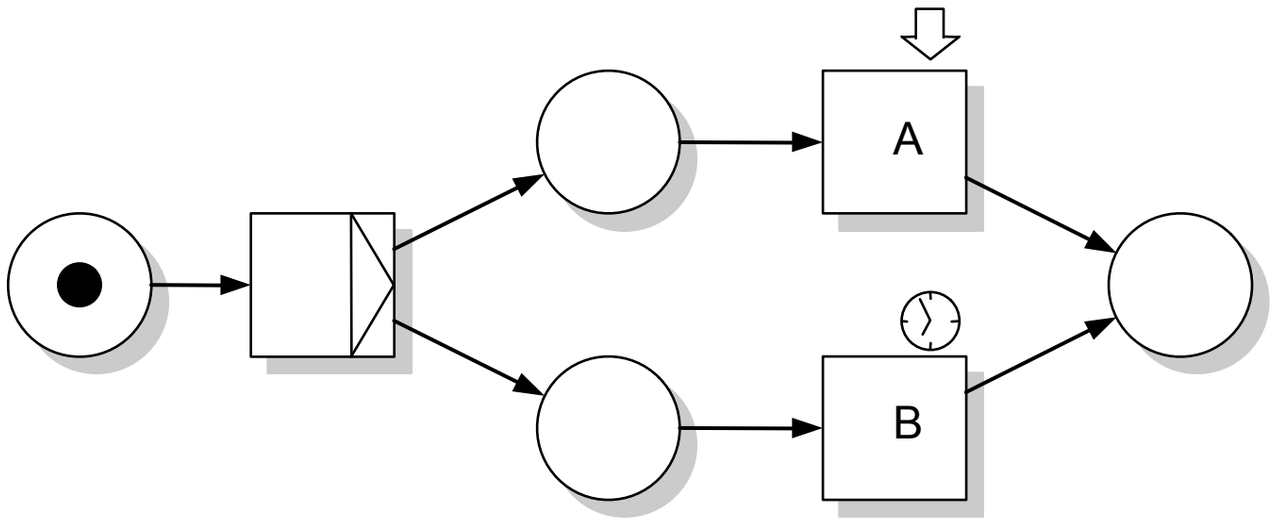
Triggers: example



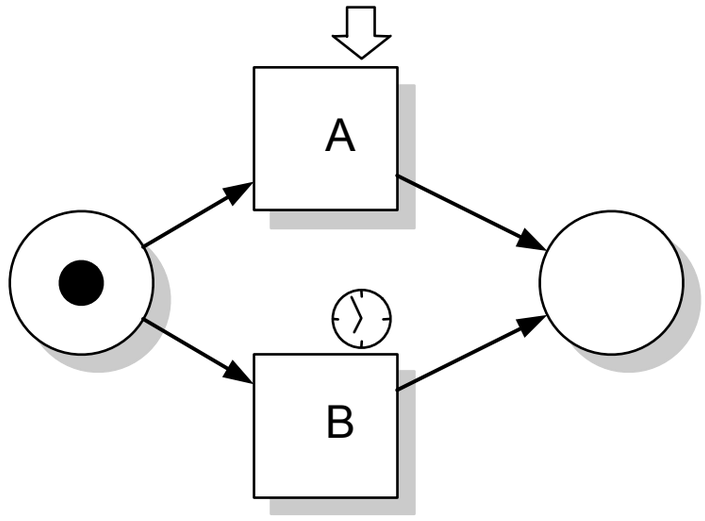
Triggers: example



Explicit vs Implicit choices (again)



(a) *Explicit xor split* does not enable A and B concurrently



(b) *Implicit xor split* enables A and B concurrently

XOR

event based

Question time
Net design: Car Damage

- An insurance company uses the following procedure for the processing of the claims
- Every claim, reported by a customer, is registered
- After the registration, the claim is classified
- There are two categories: simple and complex claims.
  - For simple claims two tasks need to be executed:
    check insurance and phone garage.
    These tasks are *independent* of each other.
  - The complex claims require three tasks:
    check insurance, check damage history and phone garage.
    These tasks need to be *executed sequentially* in the order specified.
- After executing the two/three tasks a decision is taken with two possible outcomes: OK (positive) or NOK (negative).
- If the decision is positive, then insurance company will pay.
- In any event, the insurance company sends a letter to the customer.

Question time
Net design: Car Damage

- An insurance company uses the following procedure for the processing of the claims
- Every claim, reported by a customer, is **registered**
- After the registration, the claim is **classified**
- There are two categories: **simple** and **complex** claims.
  - For simple claims two tasks need to be executed:
    check insurance and **phone garage**.
    These tasks are *independent* of each other.
  - The complex claims require three tasks:
    check insurance, **check damage history** and **phone garage**.
    These tasks need to be *executed sequentially* in the order specified.
- After executing the two/three tasks a **decision** is taken with two possible outcomes: **OK** (positive) or **NOK** (negative).
- If the decision is positive, then insurance company will **pay**.
- In any event, the insurance company **sends a letter** to the customer.

Question time

Net design: Car Damage