Behaviour Diagrams:
Activity Diagrams

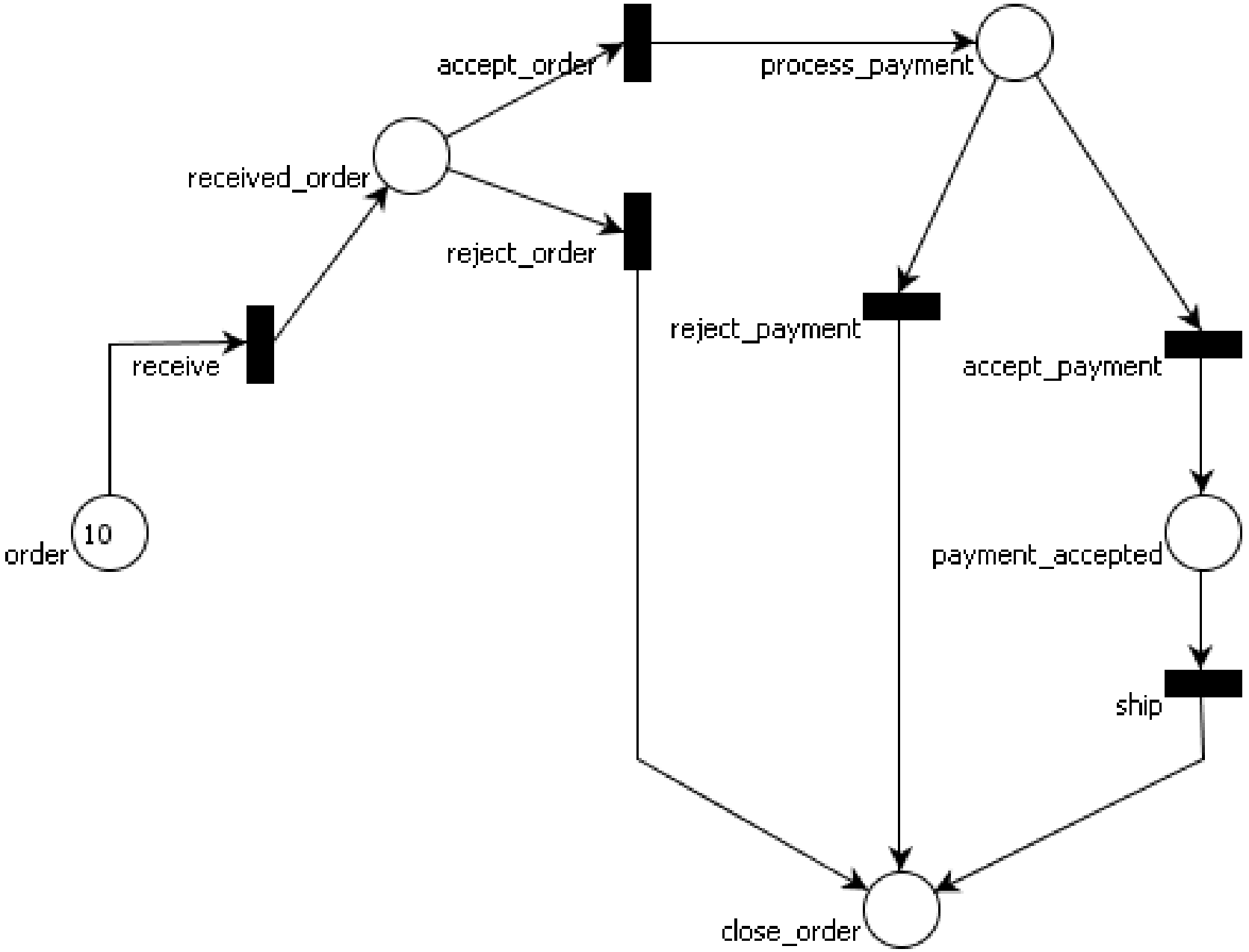
Activity Diagrams

- Describe behaviour
  - of overall system (not individual objects)
  - at high level of abstraction
  - focus on work flows (processes/activities)
  - elegant description of concurrency and interaction
  - can express non-determinism
  - as of UML 2.0 based on Petri Nets (before: state automata)

Petri Nets

- Formalism extending FSA, PetriNets are a graphical notation.
- They were developped by C.A. Petri in the 1960's as part of his PhD thesis.
- Additions to FSA:
  - Explicitly (graphically) represents when an event is enabled
    - → describe control logic
  - Elegant notation of concurrency, synchronization
  - Express non-determinism

Example


accept_order
process_payment
received_order
reject_order
reject_payment
accept_payment
receive
order
10
payment_accepted
ship
close_order

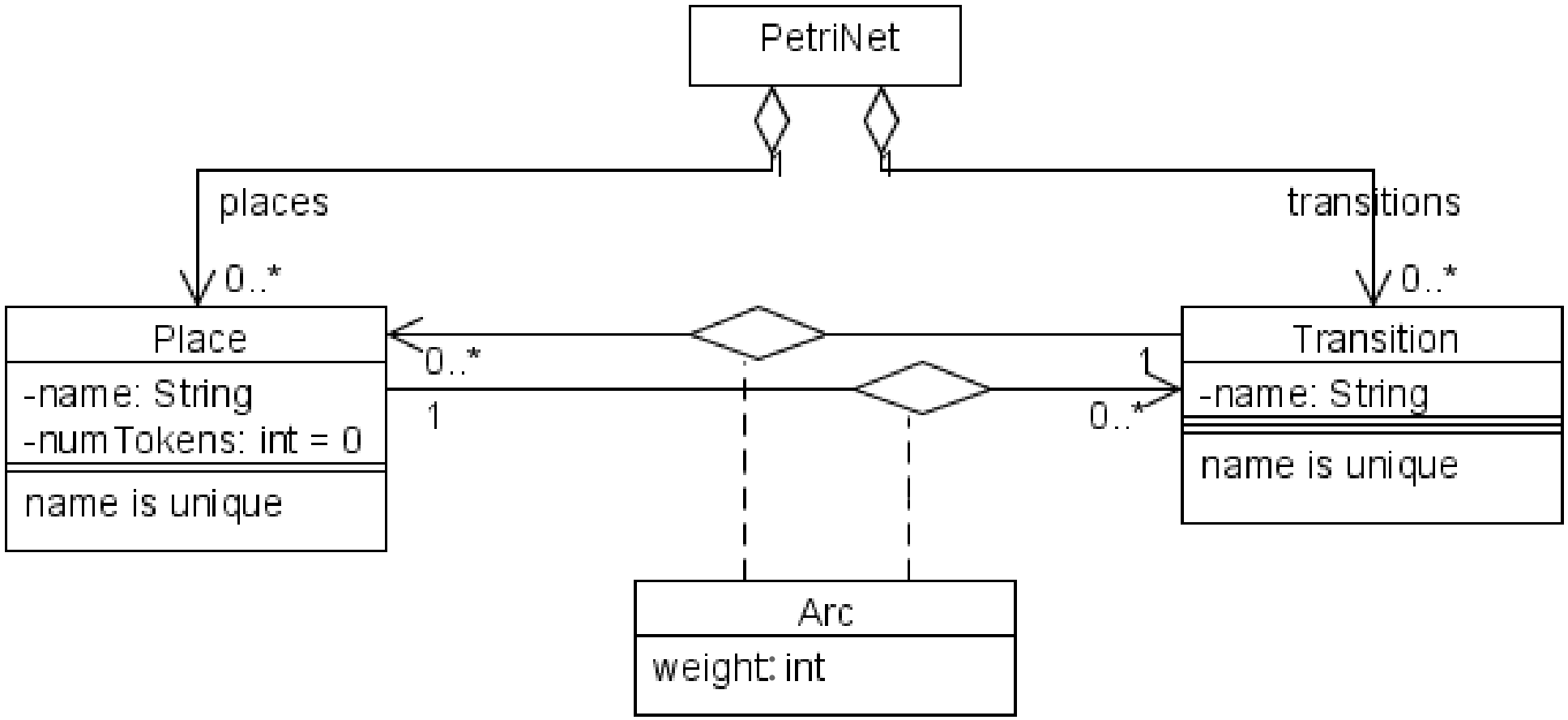
Petrinet Notation and Definition

- A Petrinet is defined by the following tuple

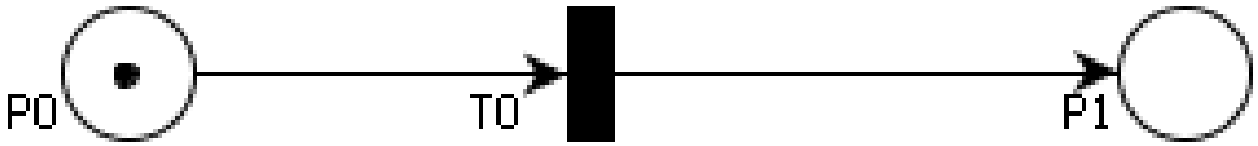
$$(P, T, A, w)$$

- $P = \{ p_1, p_2, \ldots \}$ is a finite set of places
- $T = \{ t_1, t_2, \ldots \}$ is a finite set of transitions
- $A \subseteq (P \times T) \cup (T \times P)$ is a set of arcs
- $w: A \rightarrow N$ is a weight function

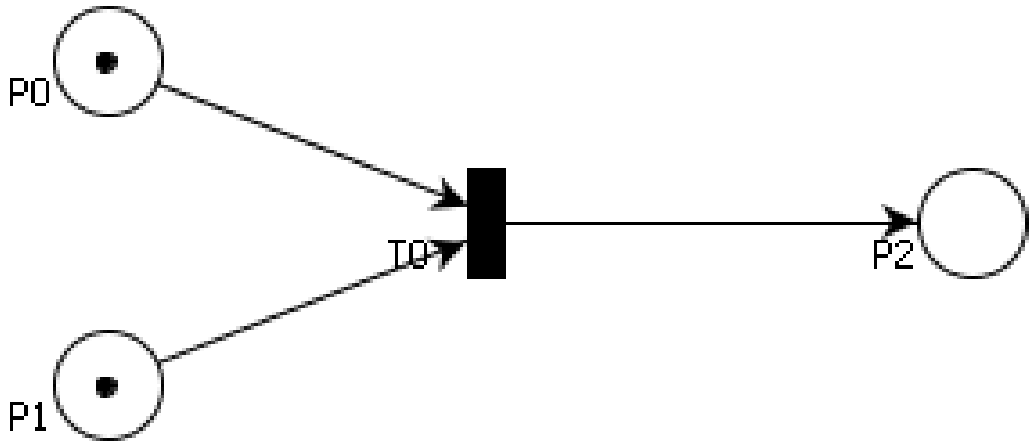
Petri Net (meta-)model



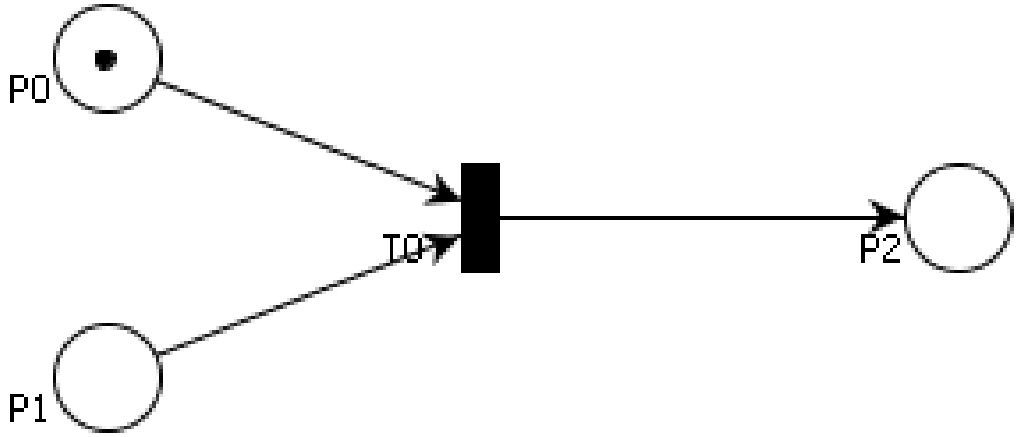
Simple



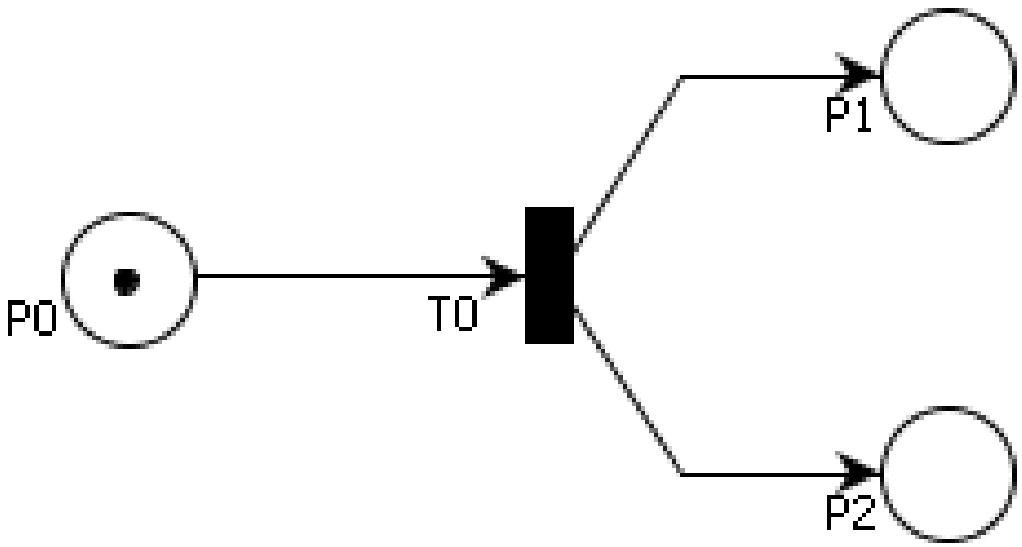
Join



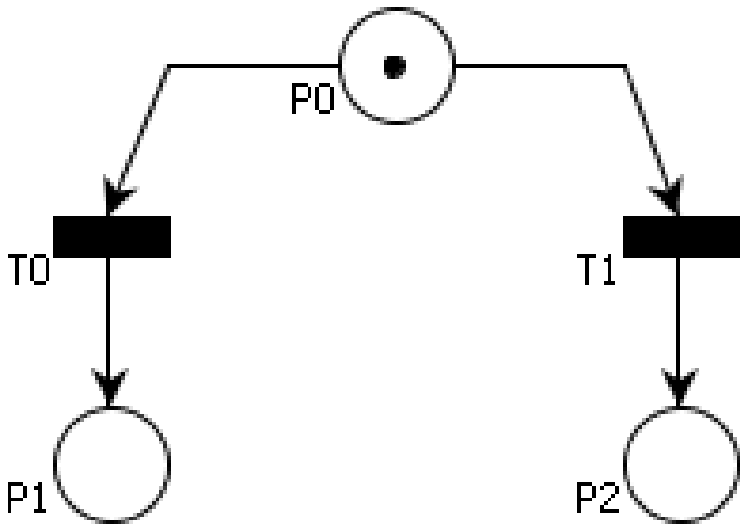
Not Live



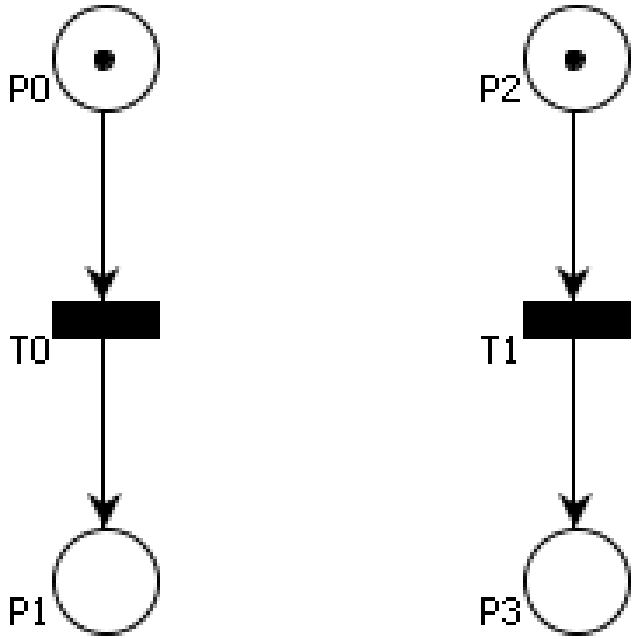
Fork



Conflict, Choice, Decision



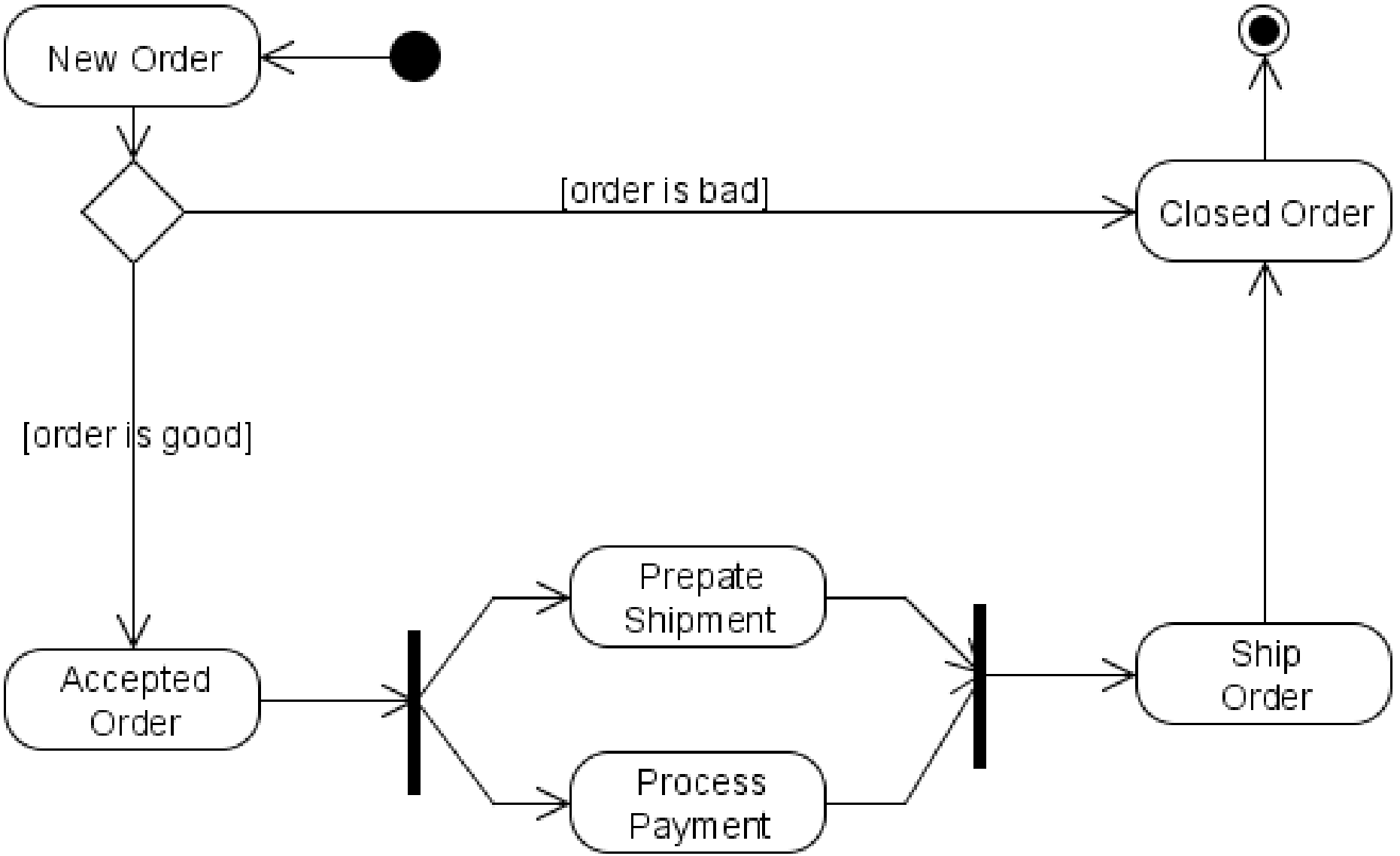
Concurrency



Back to Activity Diagrams

- As already mentioned, Activity Diagrams in UML 2.0 are based of Petri Nets.
- Although the notion of tokens is not used, critical elements such as places (called activities here) and transitions remain.

Activity Diagram



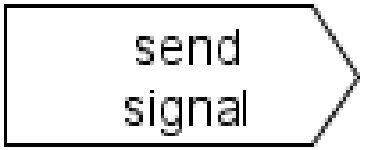
And a bit of this ...

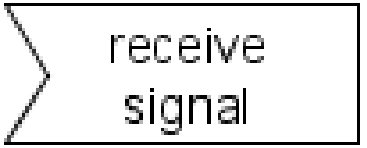
- As you can see, Activity diagrams use constructs from many formalisms.
- The core of the diagram is Petri Nets, using places and transitions.
- You can see a FSA influence, with the presence of an Initial state and an Accept state.
- Activity diagrams also provide additional constructs, such as conditionals, signaling and timing.

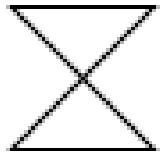
Signals and Timing

- The signal construct allows a Activity diagram to interact with external components.
  - The block with the outwards triangle indicates that a signal is sent to an external component.
  - The block with an inwards triangle indicates that the activity is blocked until a message is received.
- The timing construct allows an activity diagram to trigger certain tasks at specific times.

  ex: 8h00 am, April 1st, etc

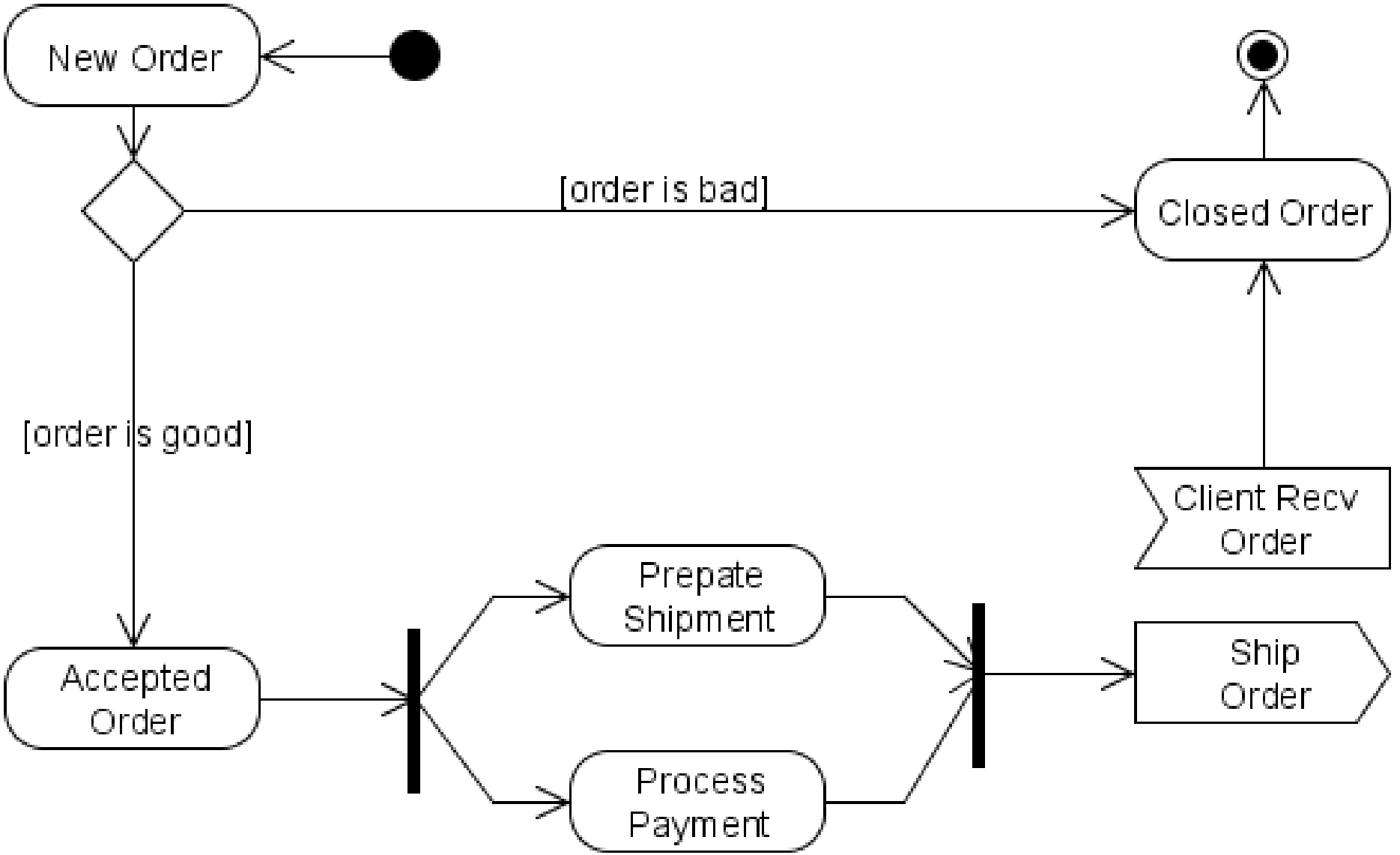






time signal

Signal Example



Timing Example