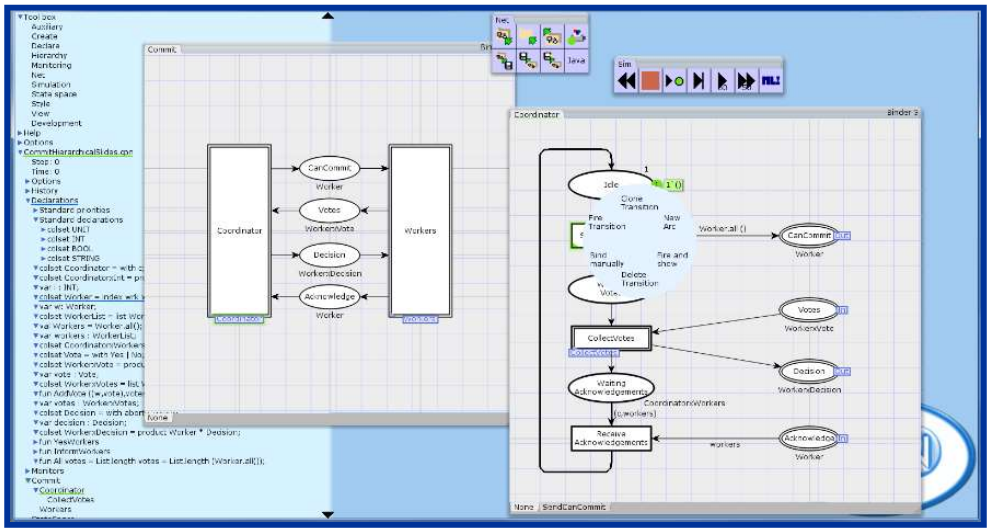
Lecture 2

Modelling with Place/Transition Nets



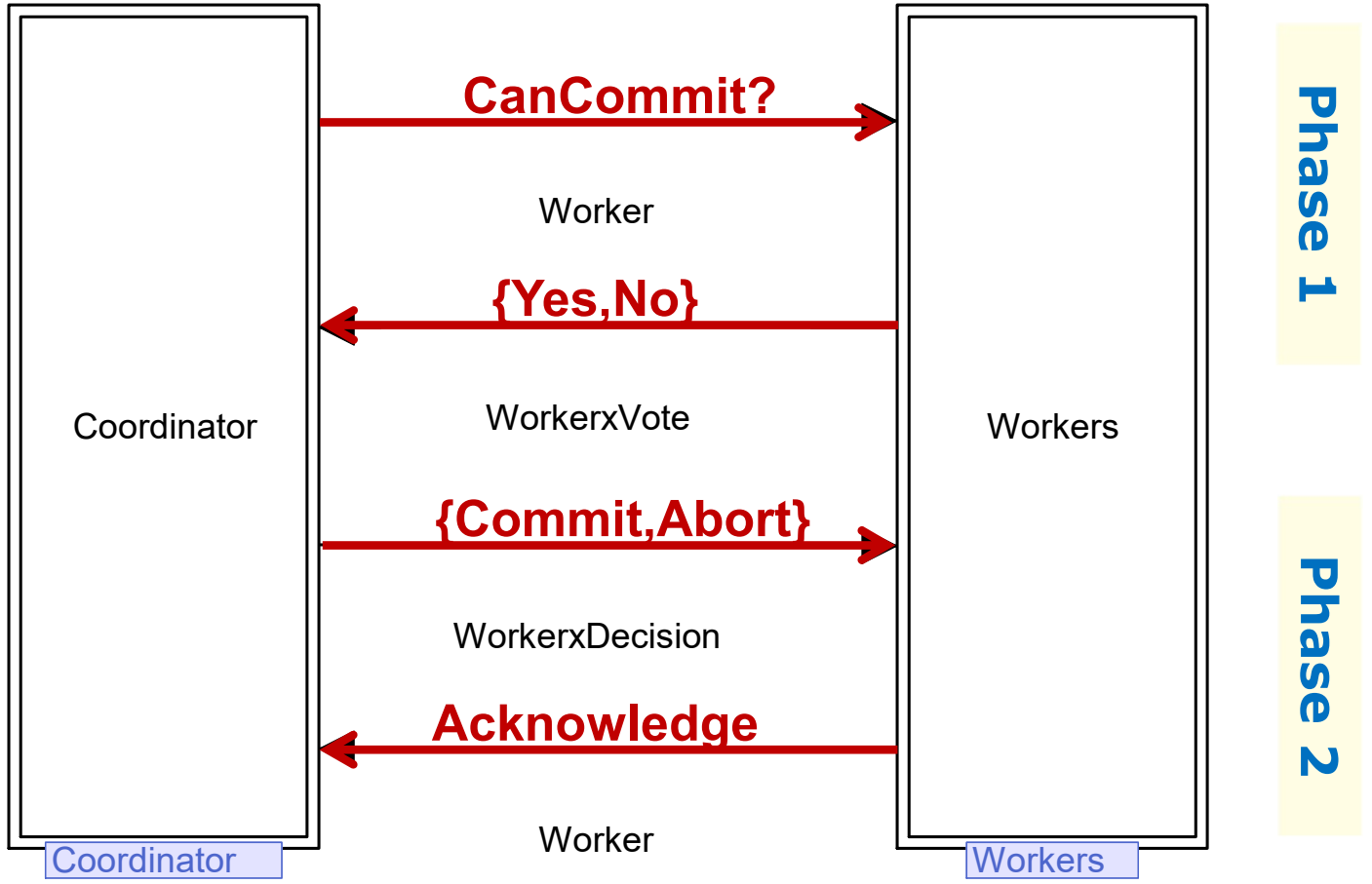
Lars M. Kristensen
Department of Computing, Mathematics, and Physics
Western Norway University of Applied Sciences
Email: lmkr@hvl.no / WWW: home.hib.no/ansatte/lmkr

Overview

- **Syntactical elements – model structure**
  - Places and transitions
  - Arcs and arc weights
  - Initial marking
- **Semantical concepts - dynamics/execution**
  - Tokens and current marking
  - Transition enabling and occurrence
  - Concurrency, conflict and non-determinism

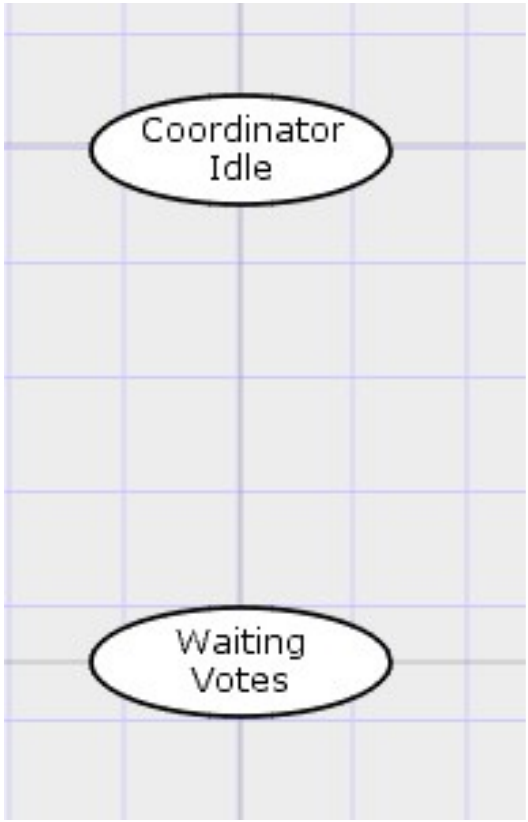
Two-phase Commit Transaction Protocol

- **We will focus on modelling the first phase**



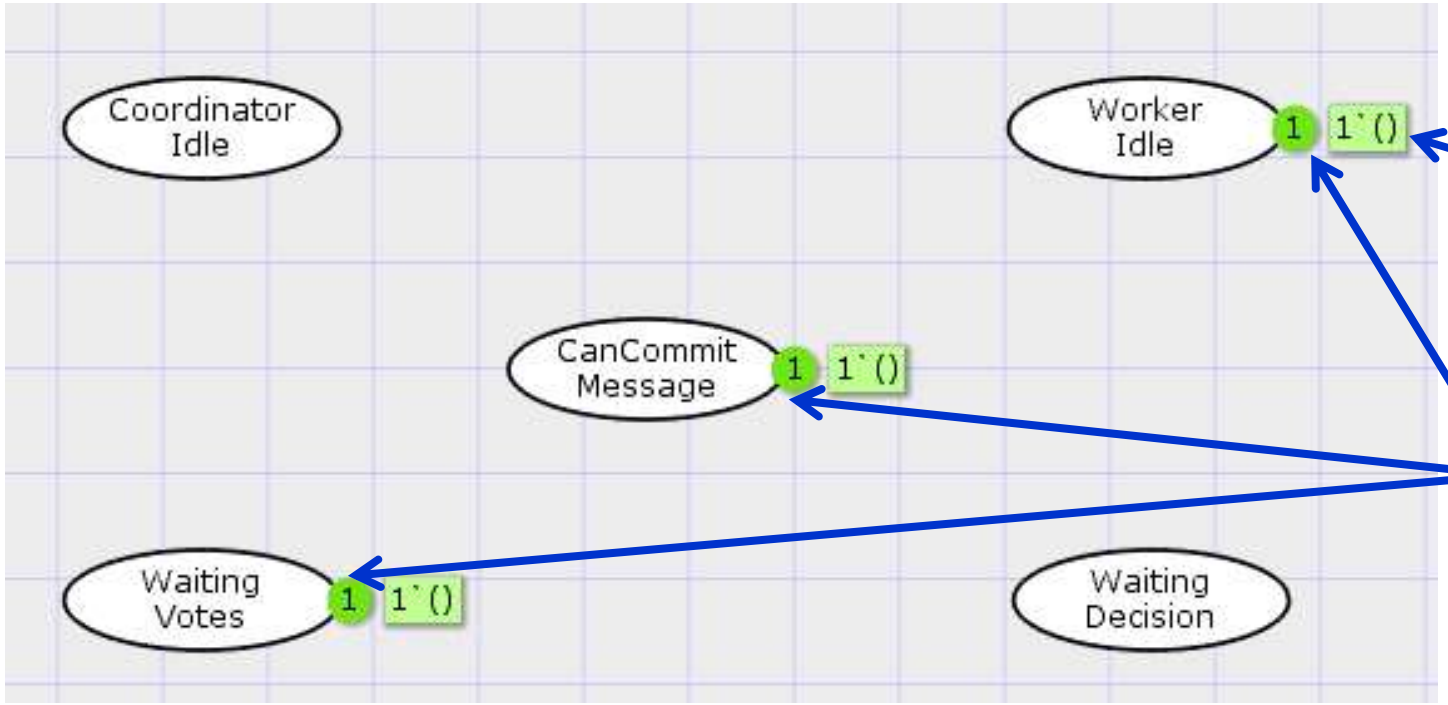
Places

- **Used to model the state of the system - drawn as ellipses**



Tokens and Markings

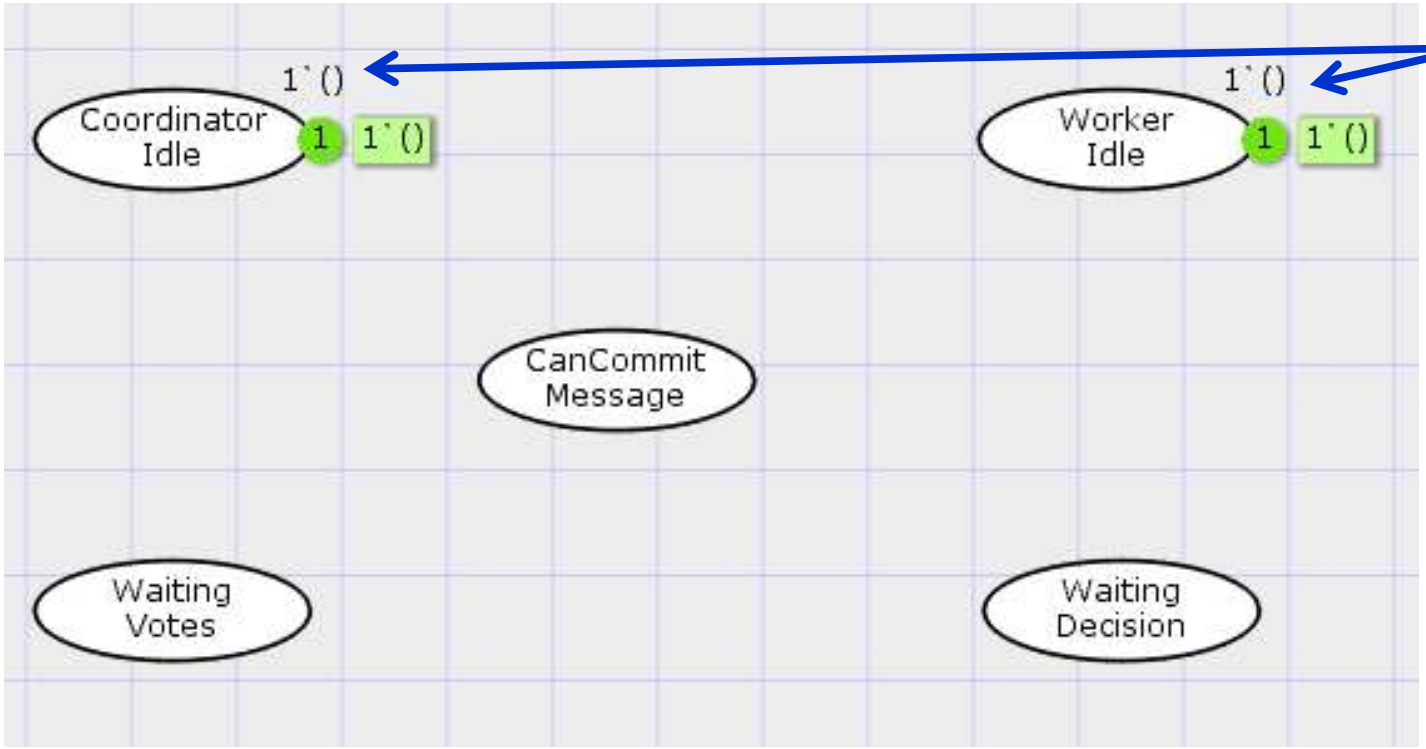
- A place can contain a number of tokens



- A marking is a distribution of tokens on the places - represents a system state.

Initial Marking

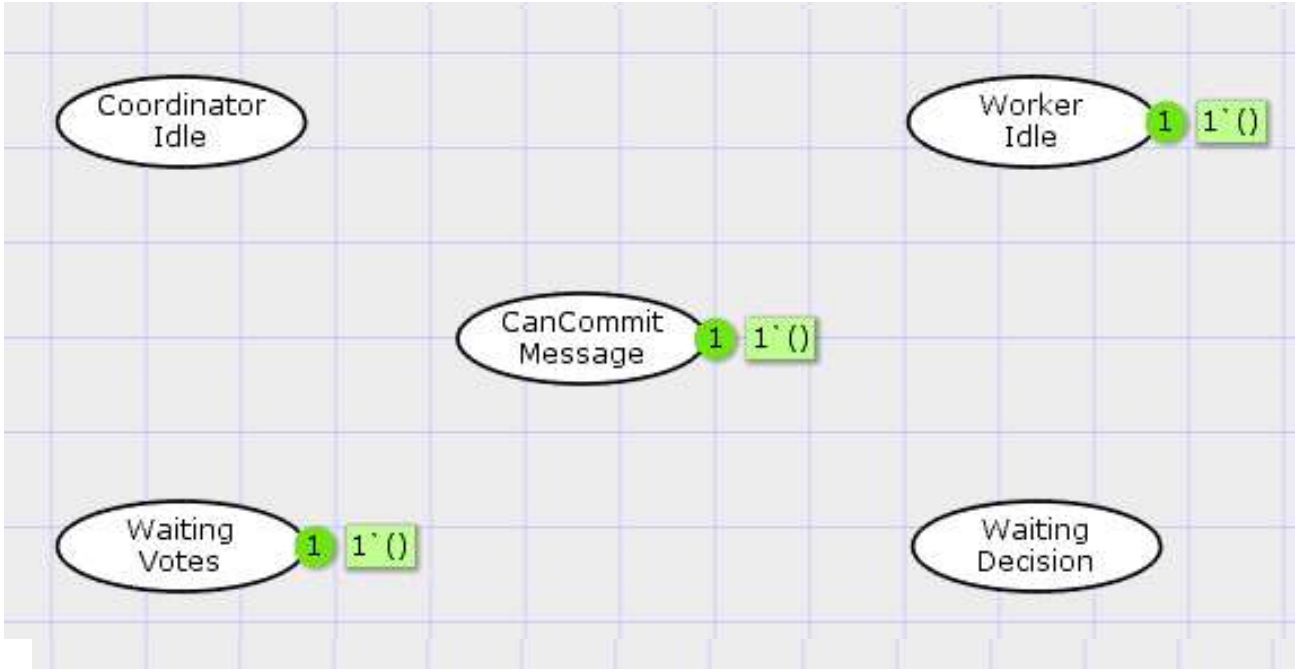
- The initial marking (token distribution) represents the initial system state.
- Specified by giving the number of tokens that are initially present on a place



Initial marking is by conventional specified above the place

Current Marking

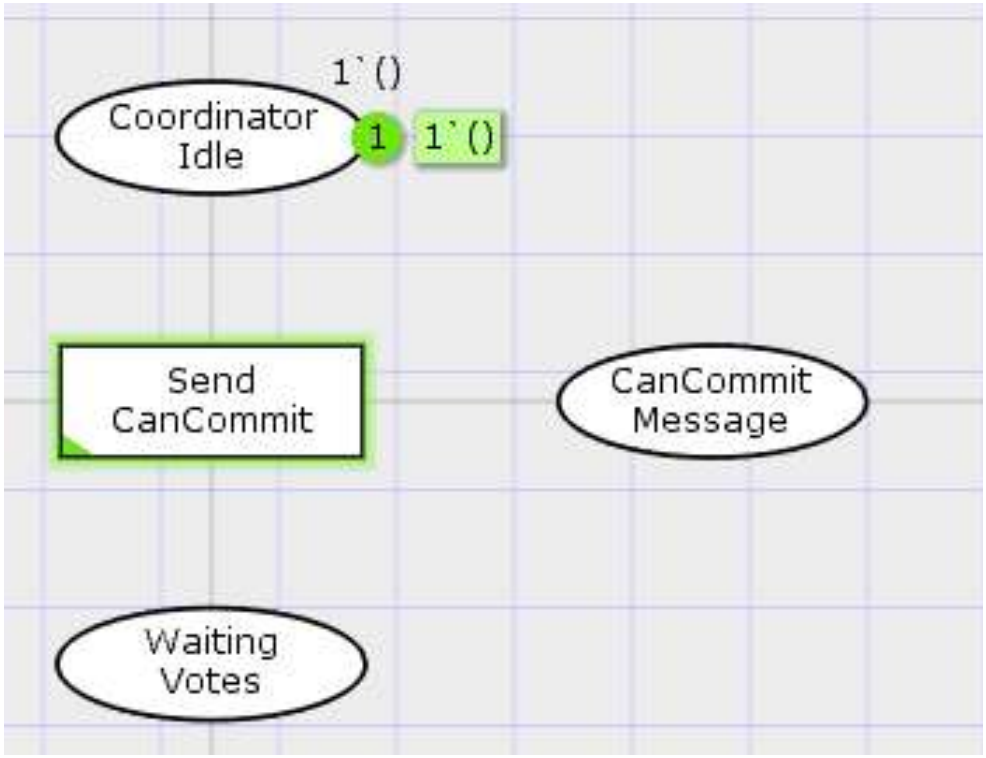
- **Current marking is representing the state that the system is currently in**



- **Starts being equal to the initial marking - changes when the model is executed.**

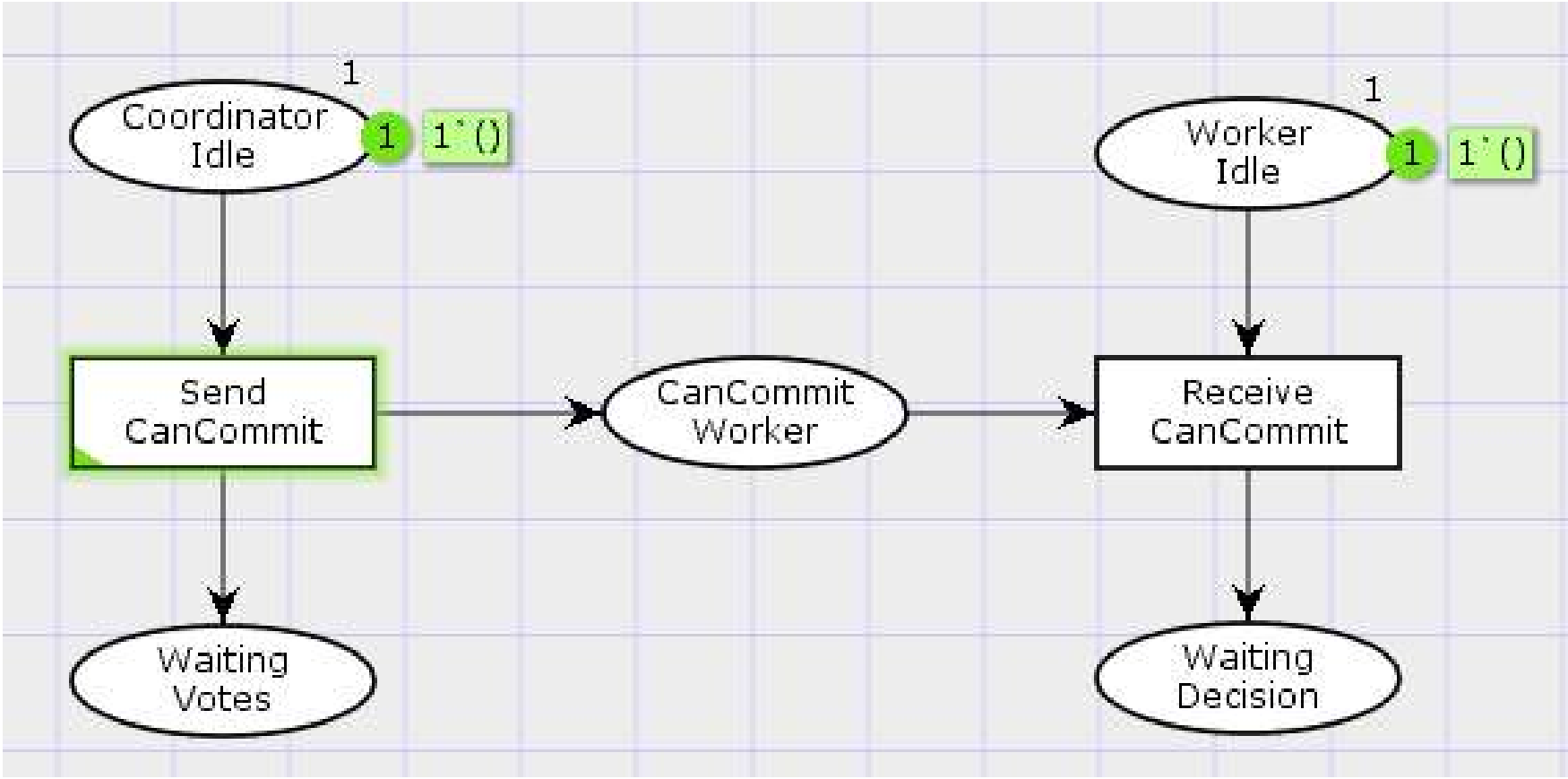
Transitions

- **Used to model the actions/events in the system - drawn as rectangles**



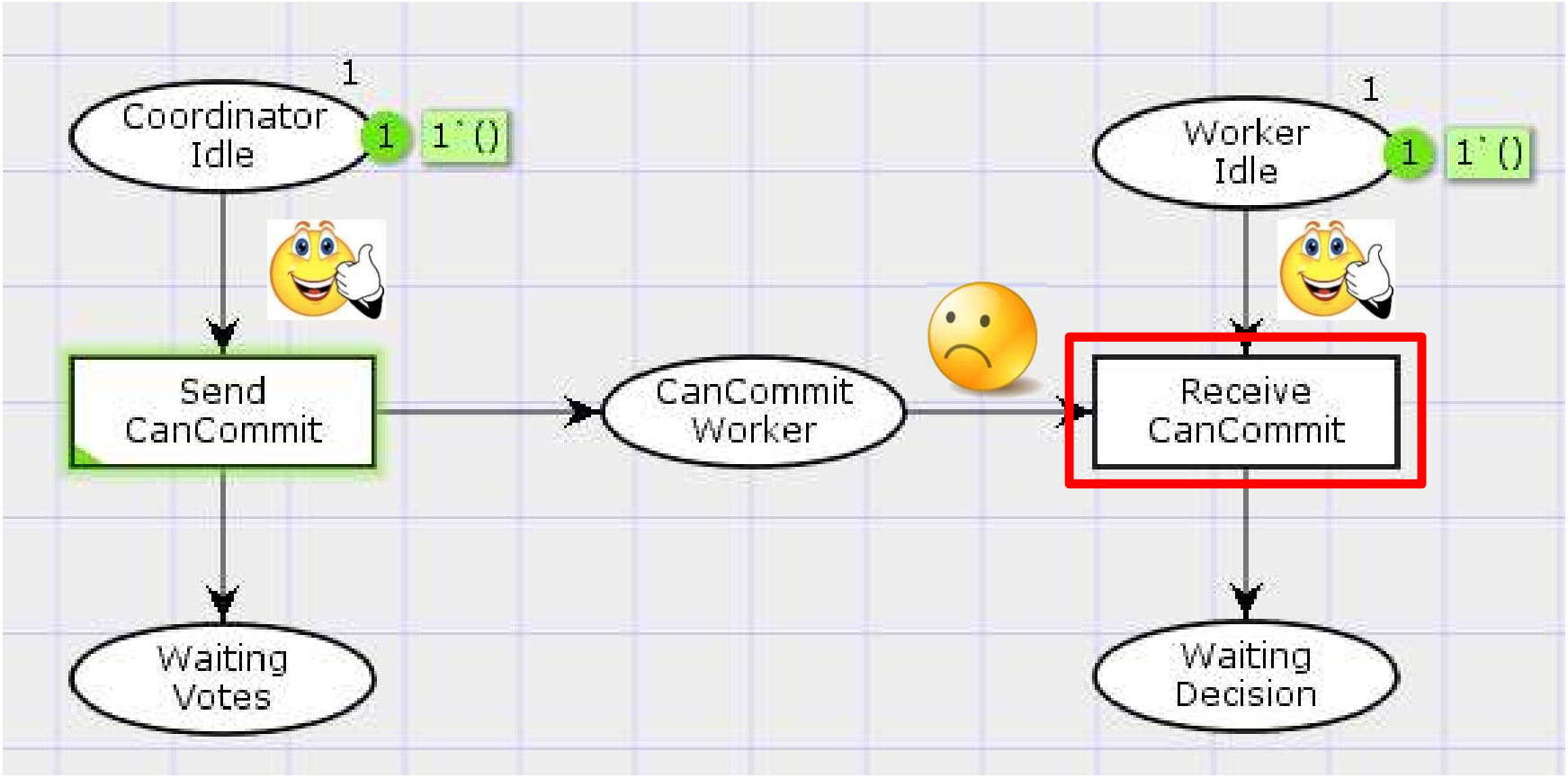
Arcs

- **Connect places and transitions and determine transition enabling and occurrence (firing):**



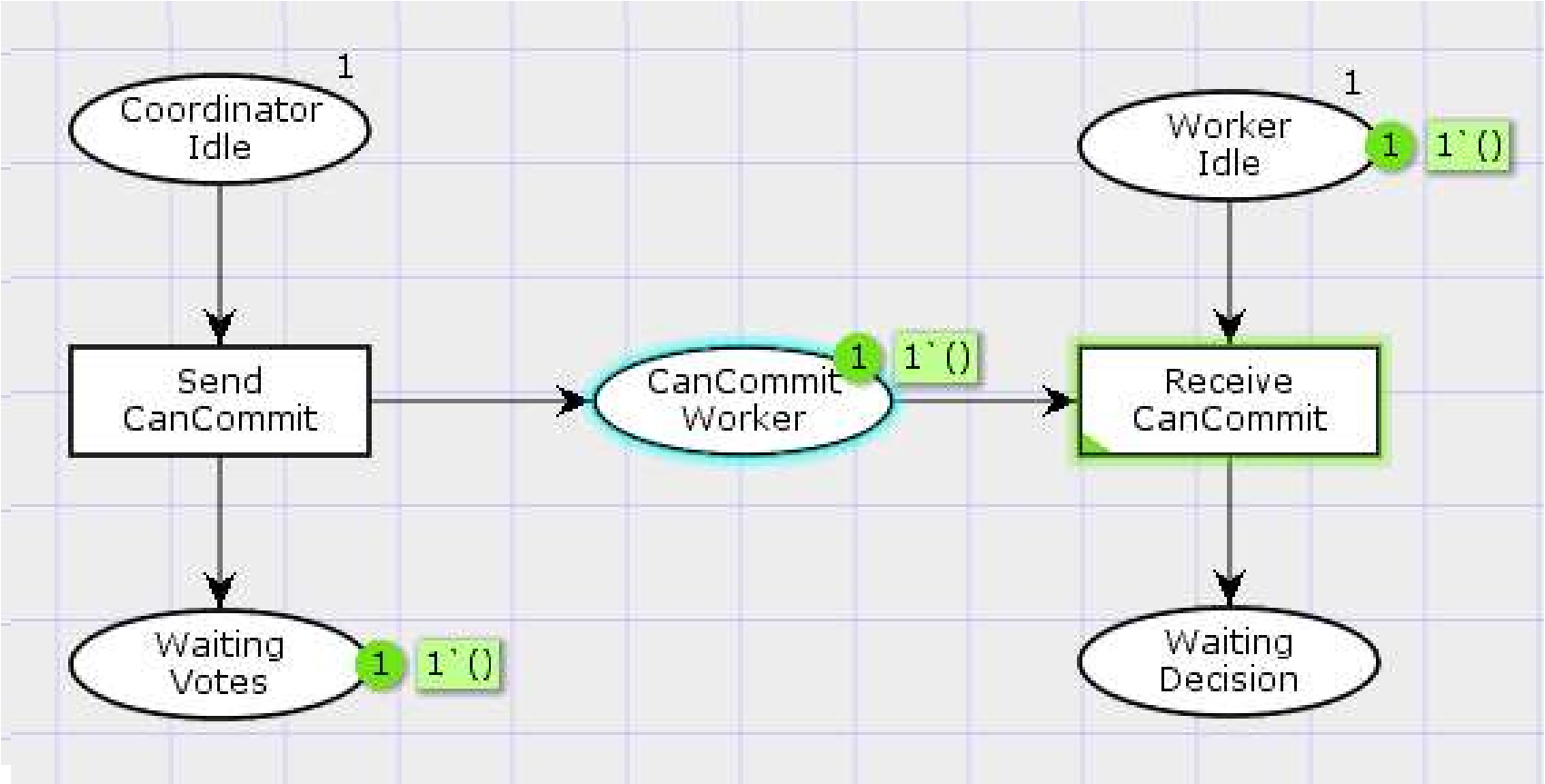
Transition Enabling

- **A transition is enabled if there is at least one token on each of its input places**



Transition Occurrence

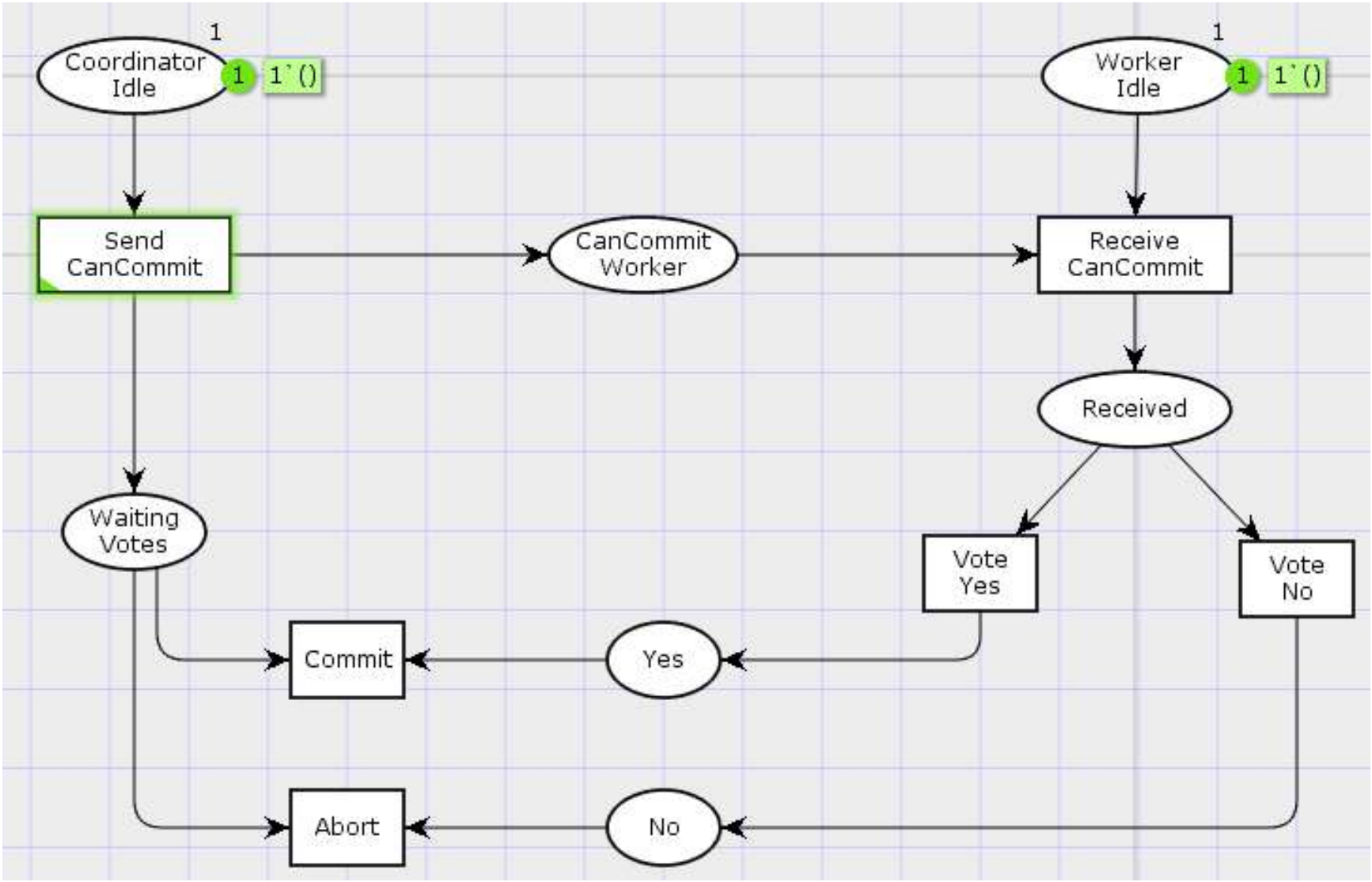
- **An enabled transition may occur (fire):**
  - Removes one token from each input place
  - Adds one token to each output place



CPN Tools Demo

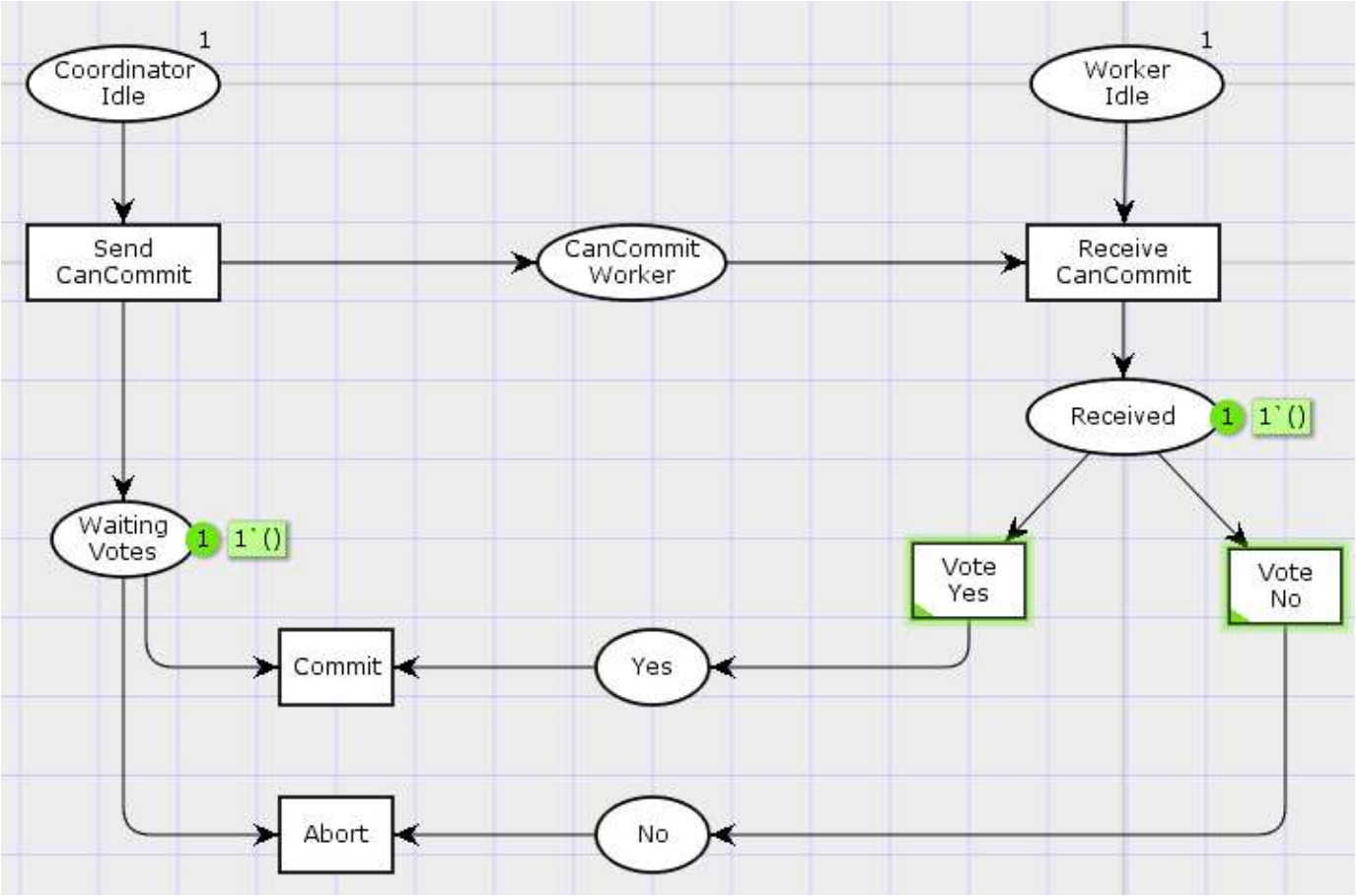
- **Simulation of CPN models**
- **Extensions to the Place/Transition-net model**
  - Modelling votes – conflict and non-determinism
  - Multiple workers - concurrency
  - Collecting votes - arc weights
  - Modelling the protocol as a reactive system

Modelling Votes



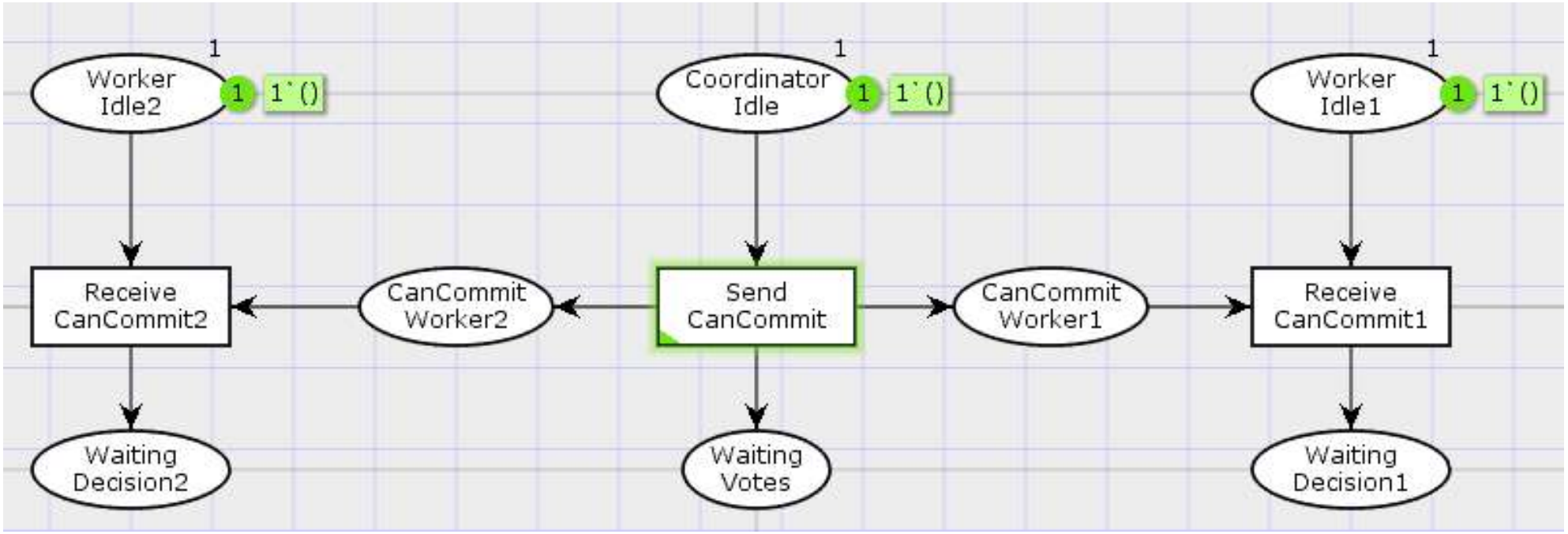
Conflict

- **Transitions are in conflict if they compete for tokens with other enabled transitions**



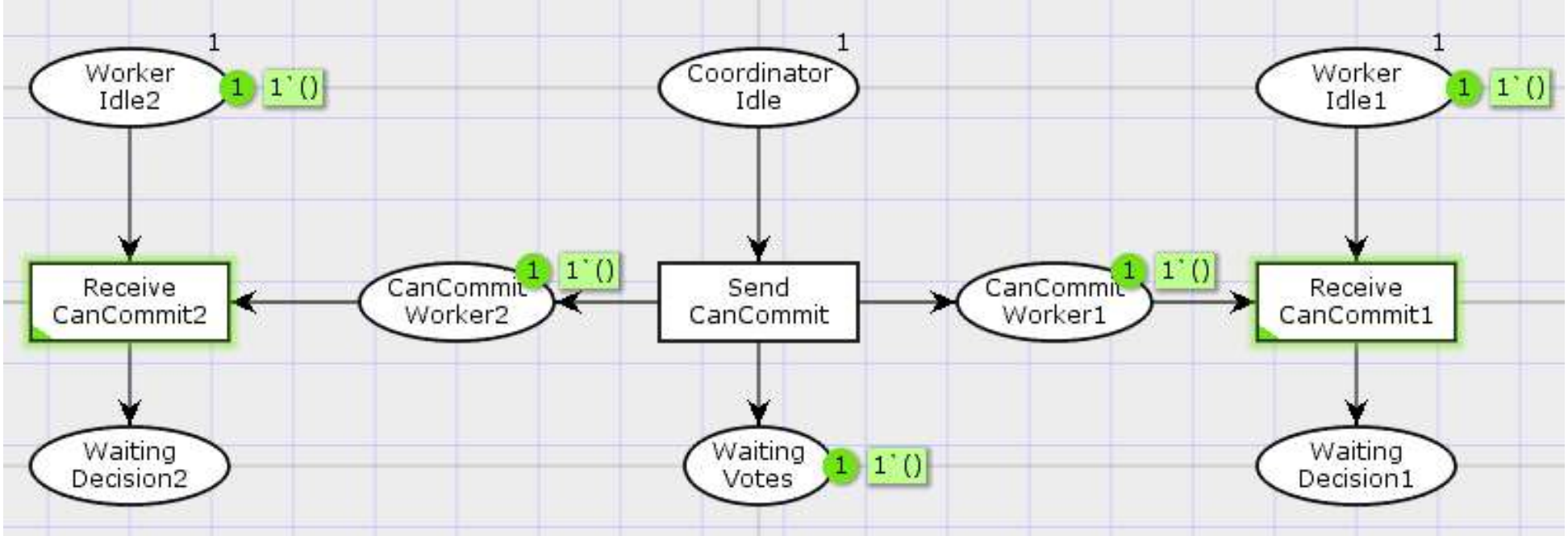
Multiple Workers

- **Extending the model to multiple workers**



Concurrency

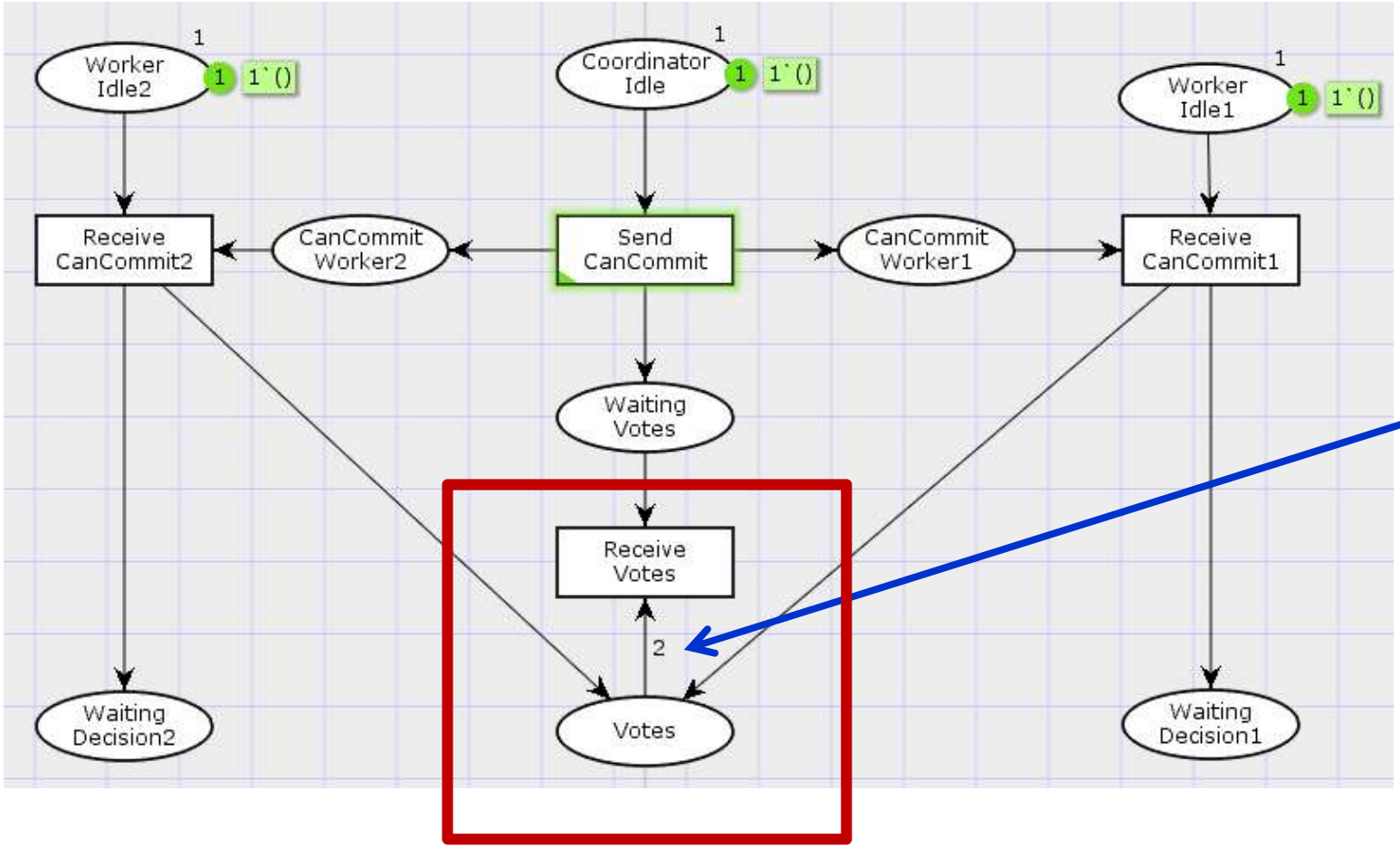
- **Transitions may be concurrently enabled in the same simulation step**



- **ReceiveCanCommit transitions can get the tokens required without sharing.**

Arc Weights

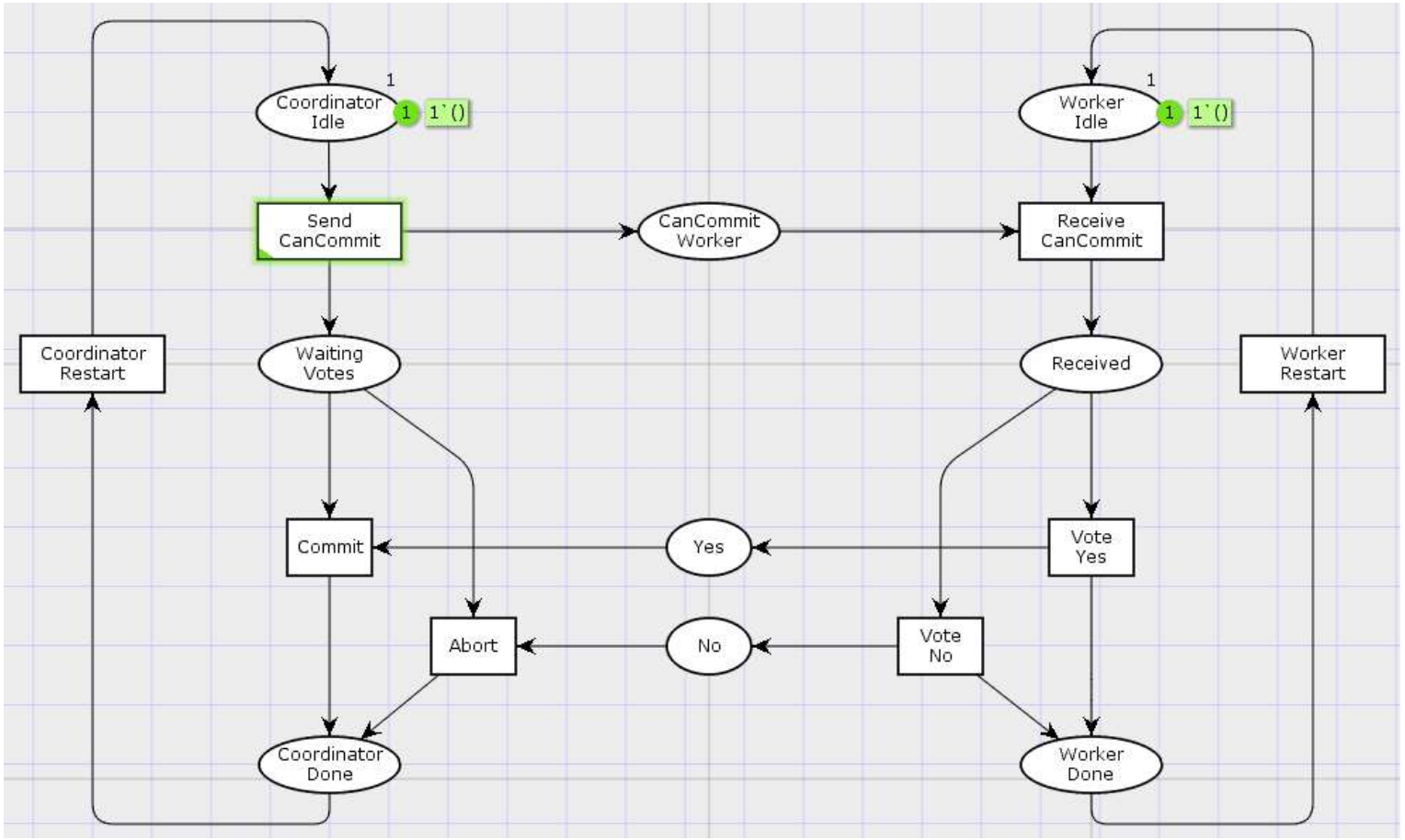
- **Number of tokens required for enabling, consumed and produced (occurrence)**



Both workers must have sent a reply

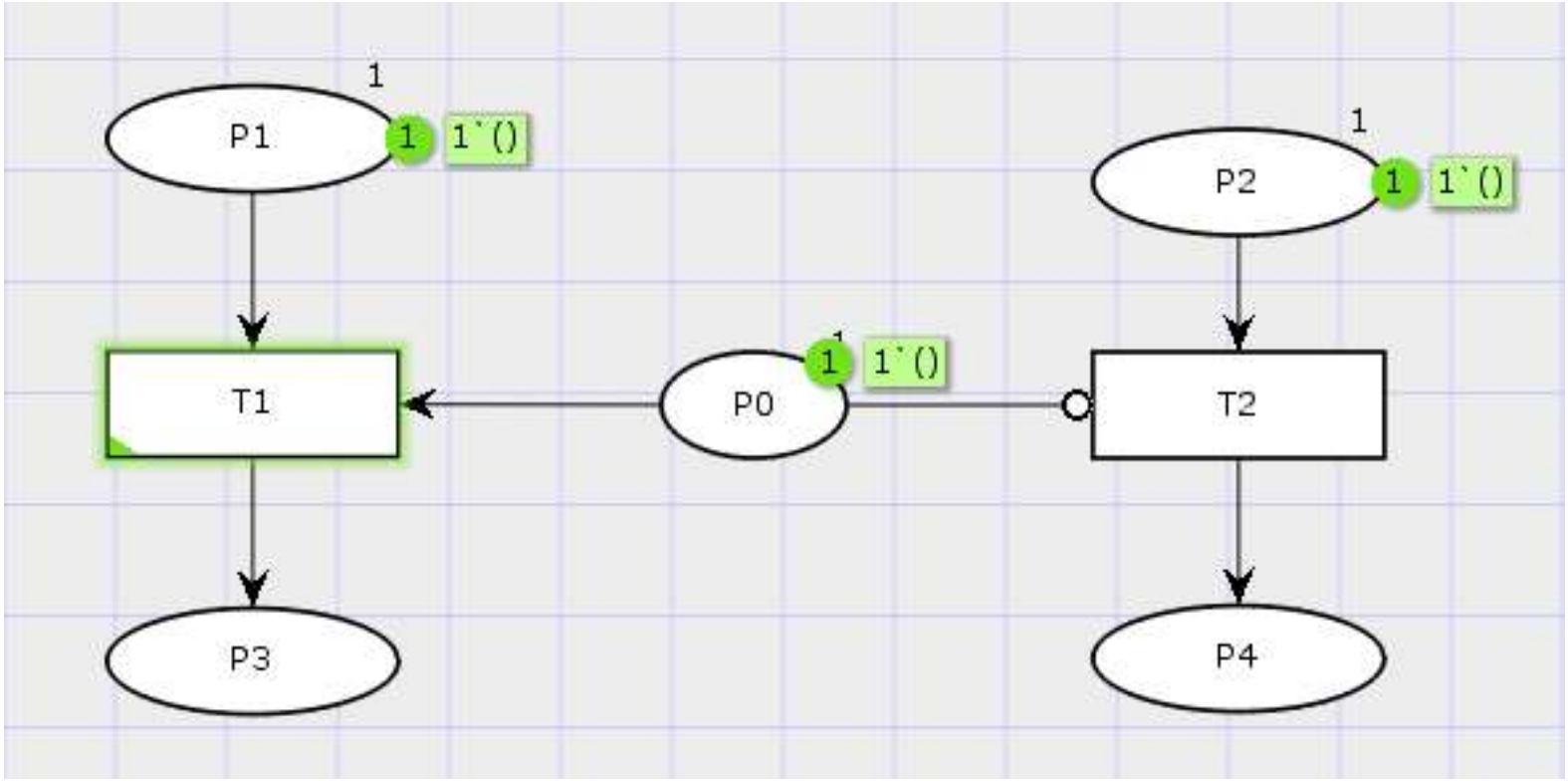
Reactive Systems

- **Many concurrency systems are intended to continuously operating**



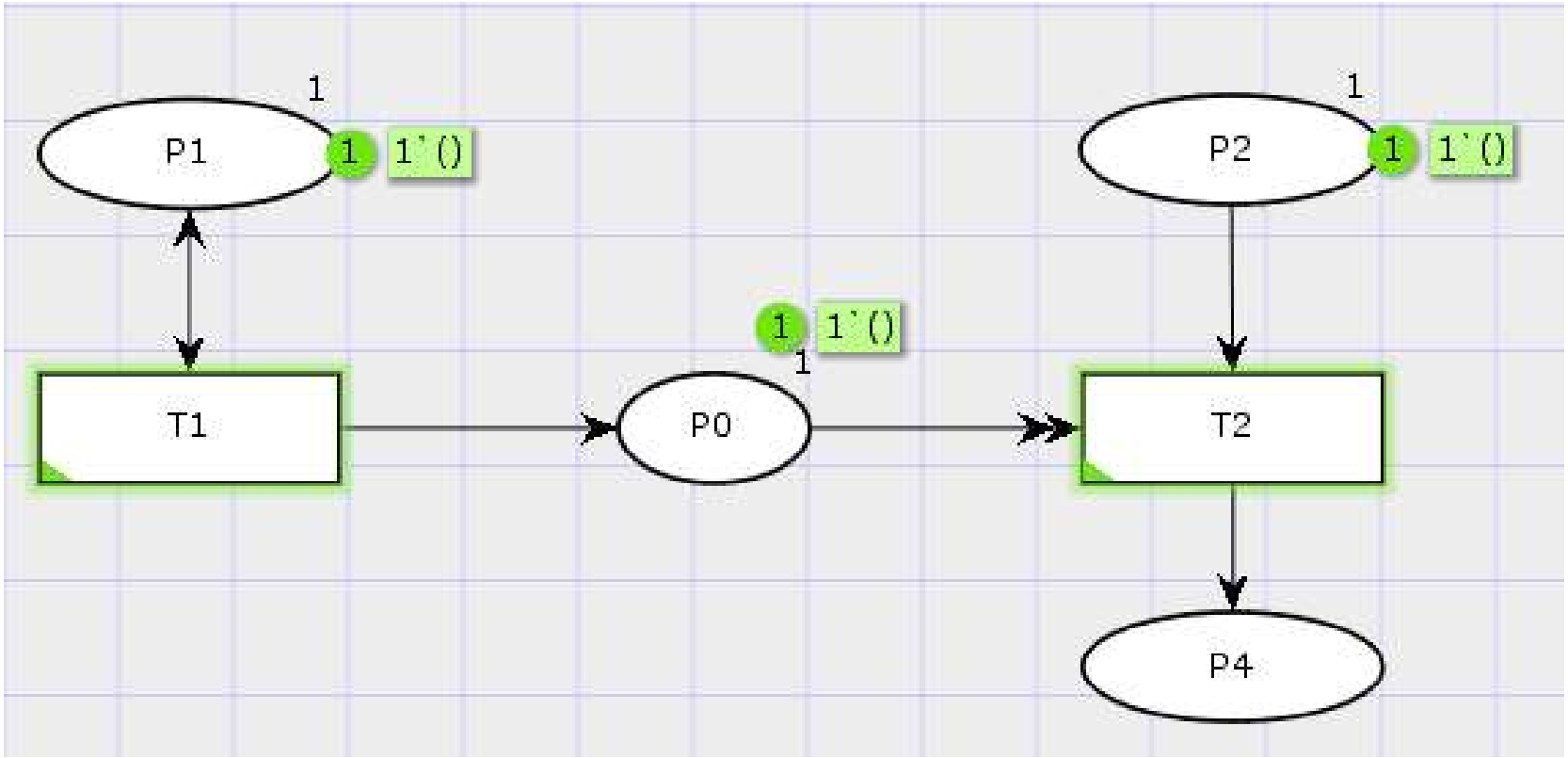
Inhibitor Arc

- **Can be used to test for the absence of tokens on a place**



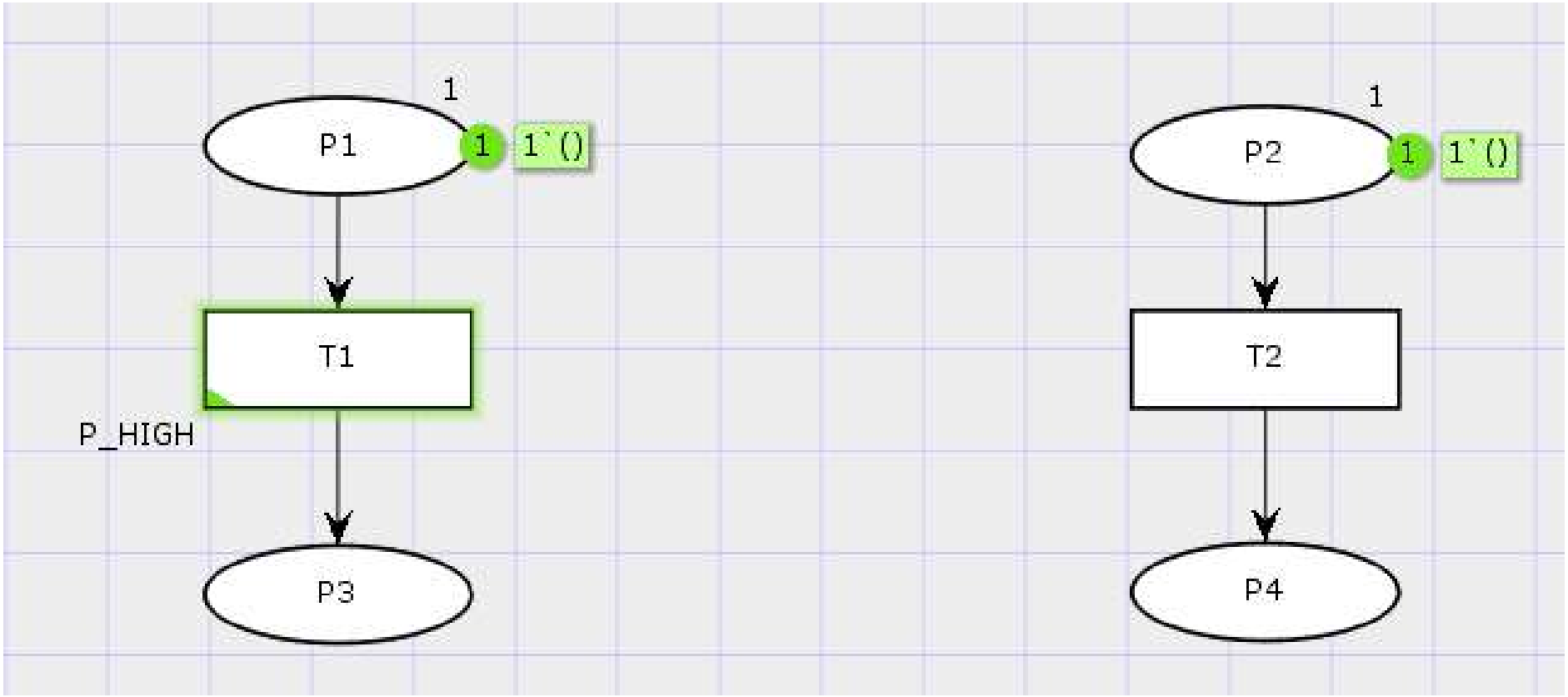
Reset Arc

- **Removes all tokens that are currently present on a place**



Transition Priorities

- **Transitions can be given a priority level**



Summary

- **Basic syntactical and semantical concepts of Place/Transition Nets introduced.**
- **Additional language constructs**
  - Inhibitor arcs and reset arcs
  - Transition priorities
- **A main limitation of Place/Transitions Nets is scalability to large (real) software systems**
  - Modelling of data is inconvenient.
  - Does not allow models to be split into modules
  - Does not support parametric systems in an elegant way