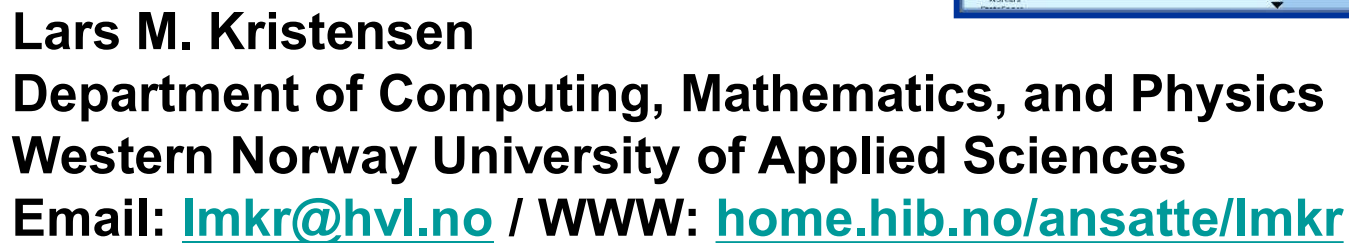
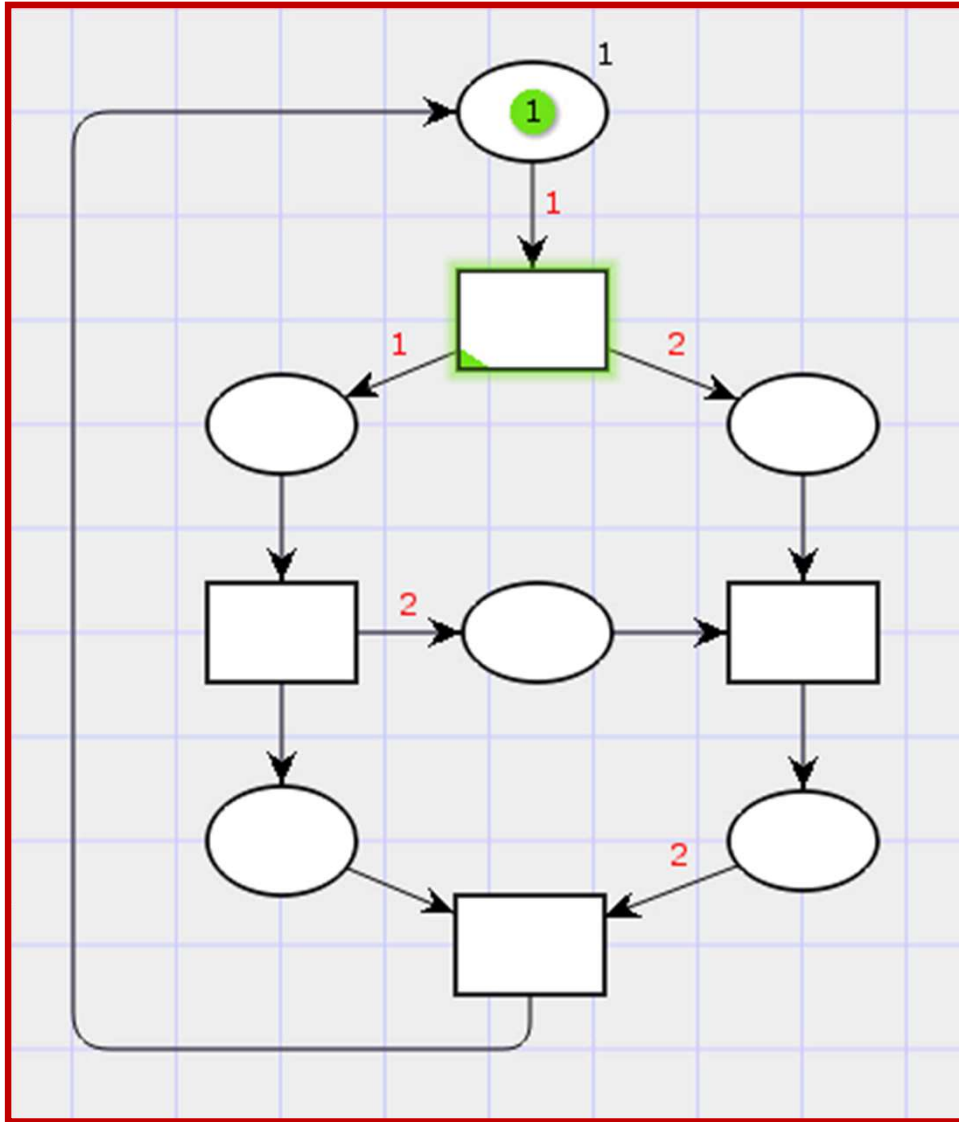


Two-phase Commit Protocol and Place/Transition Nets



Quick Recap: Petri Net Concepts



State modelling

- **Places** (ellipses) that may hold **tokens**
- **Marking (state):** distribution of **tokens** on the places
- **Initial marking:** initial state

Event (action) modelling

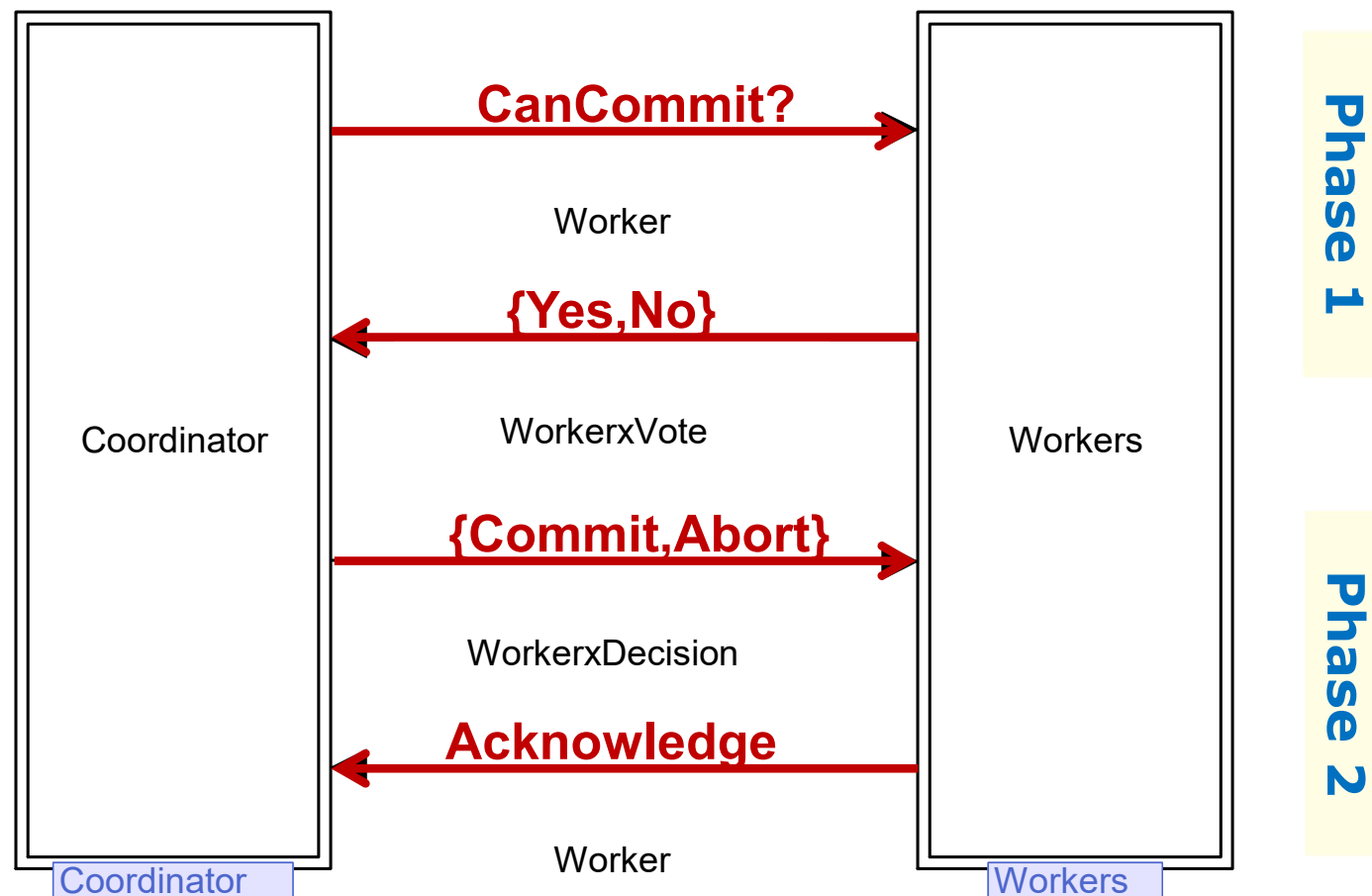
- **Transitions** (rectangles)
- **Directed arcs:** connecting places and transitions
- **Arc weights:** specifying tokens to be added/removed

Execution (token game)

- **Current marking**
- **Transition enabling**
- **Transition occurrence**

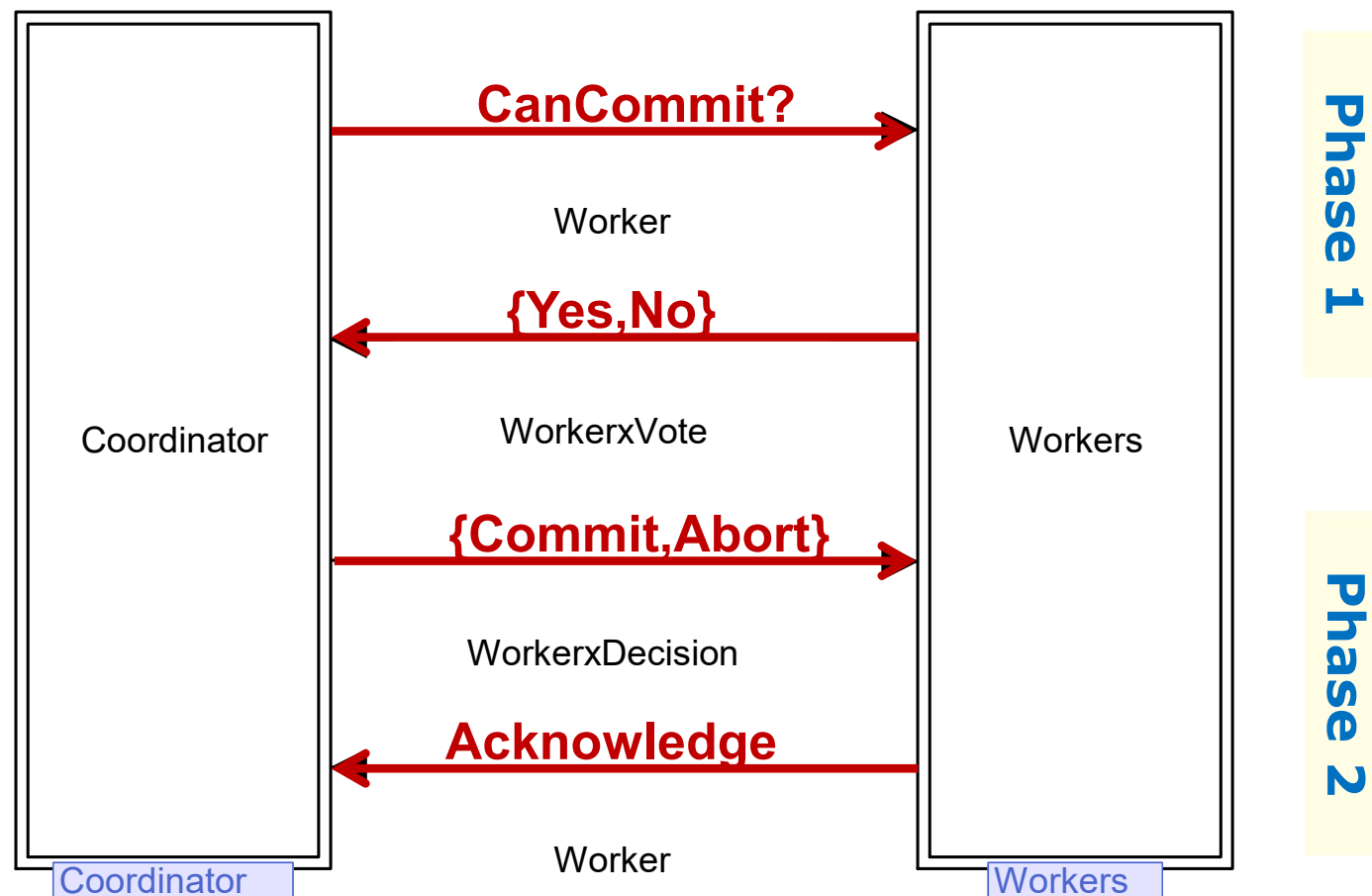
Two-phase Commit Transaction Protocol

- A **concurrent system** consisting of a **coordinator process** and a number of **worker processes**



Two-phase Commit Transaction Protocol

- How to model the first part with PT-nets?

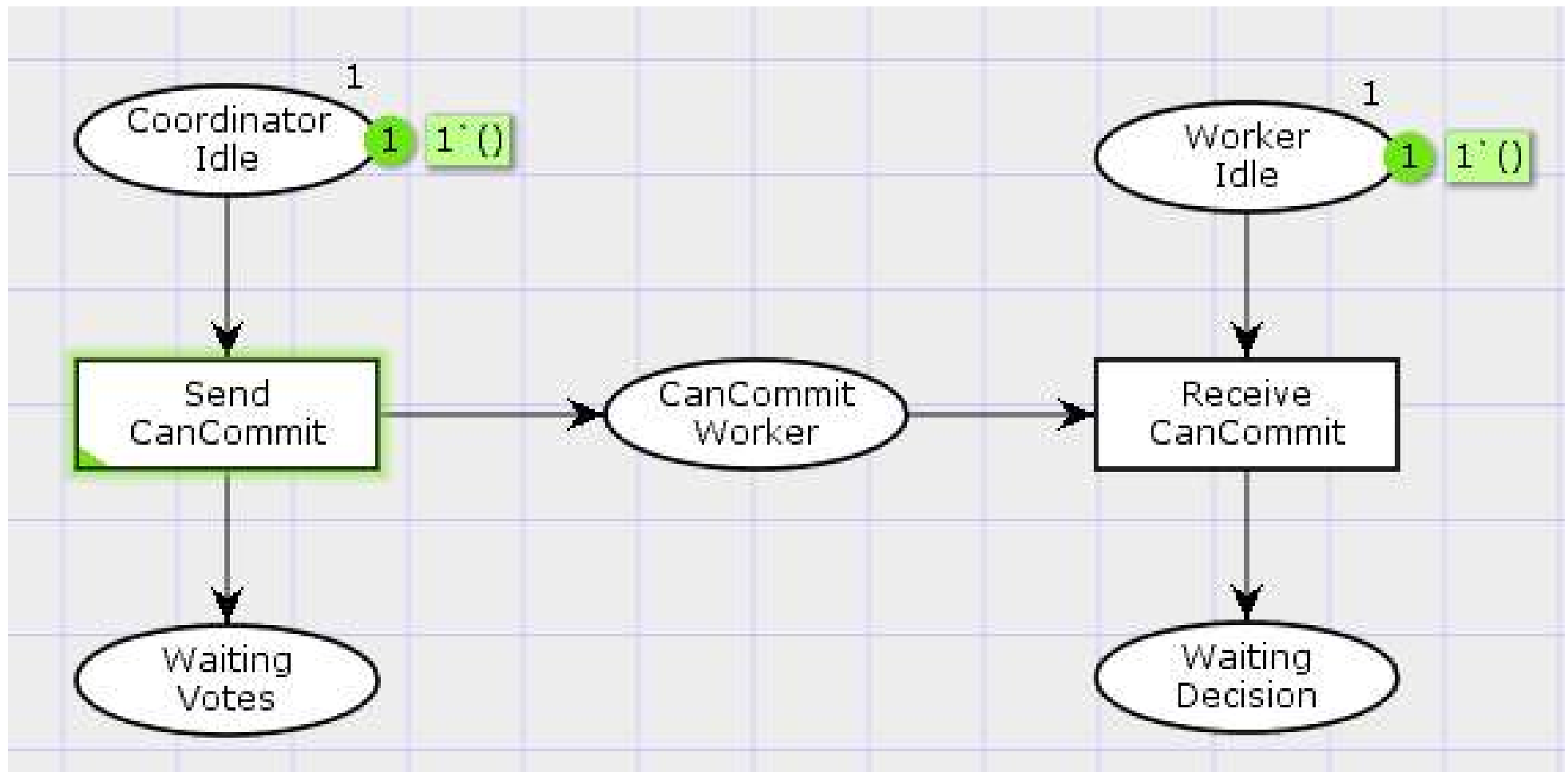


CPN Tools Demo

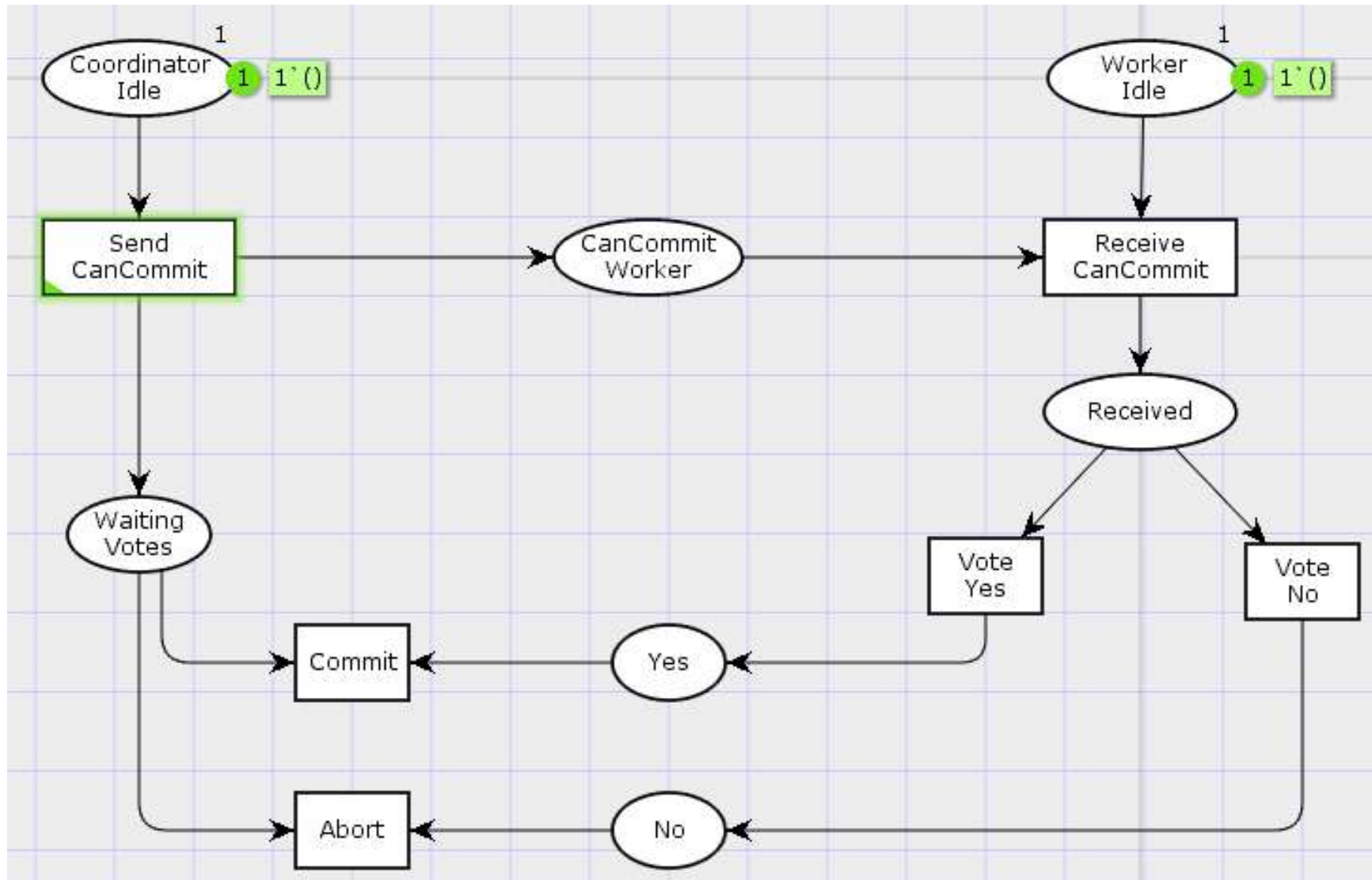
- **Construction, editing and simulation of basic Petri Net models**
- **First part of the two-phase commit protocol using Place/Transition Nets**



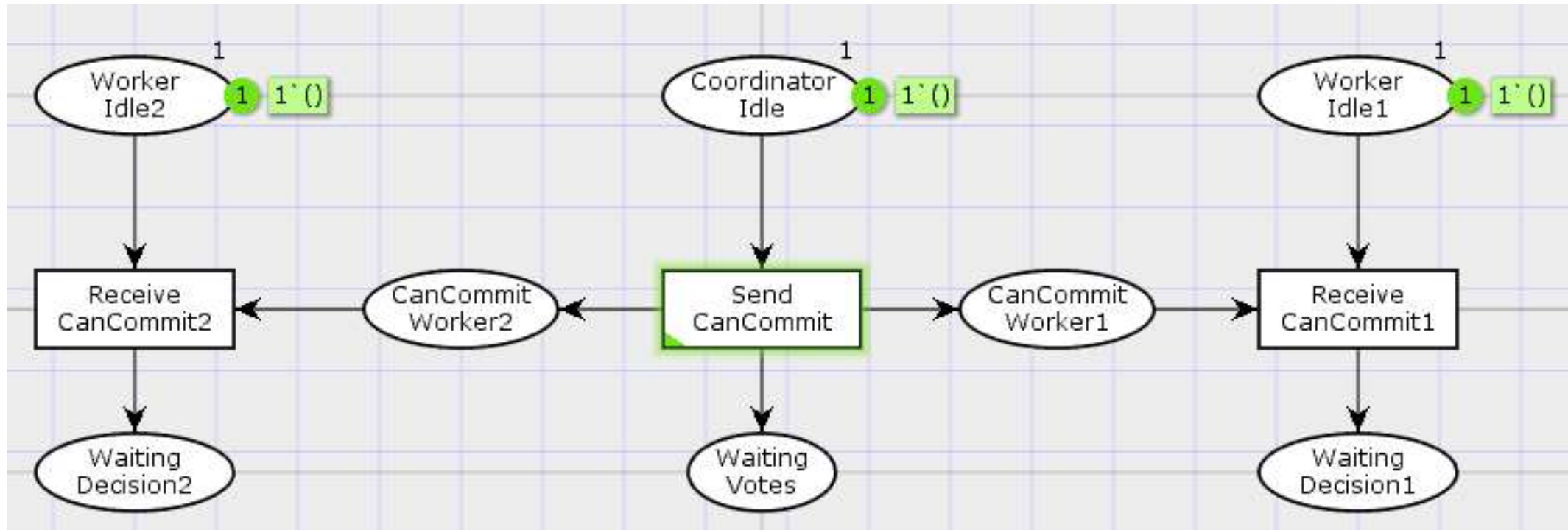
Sending/Receiving CanCommit



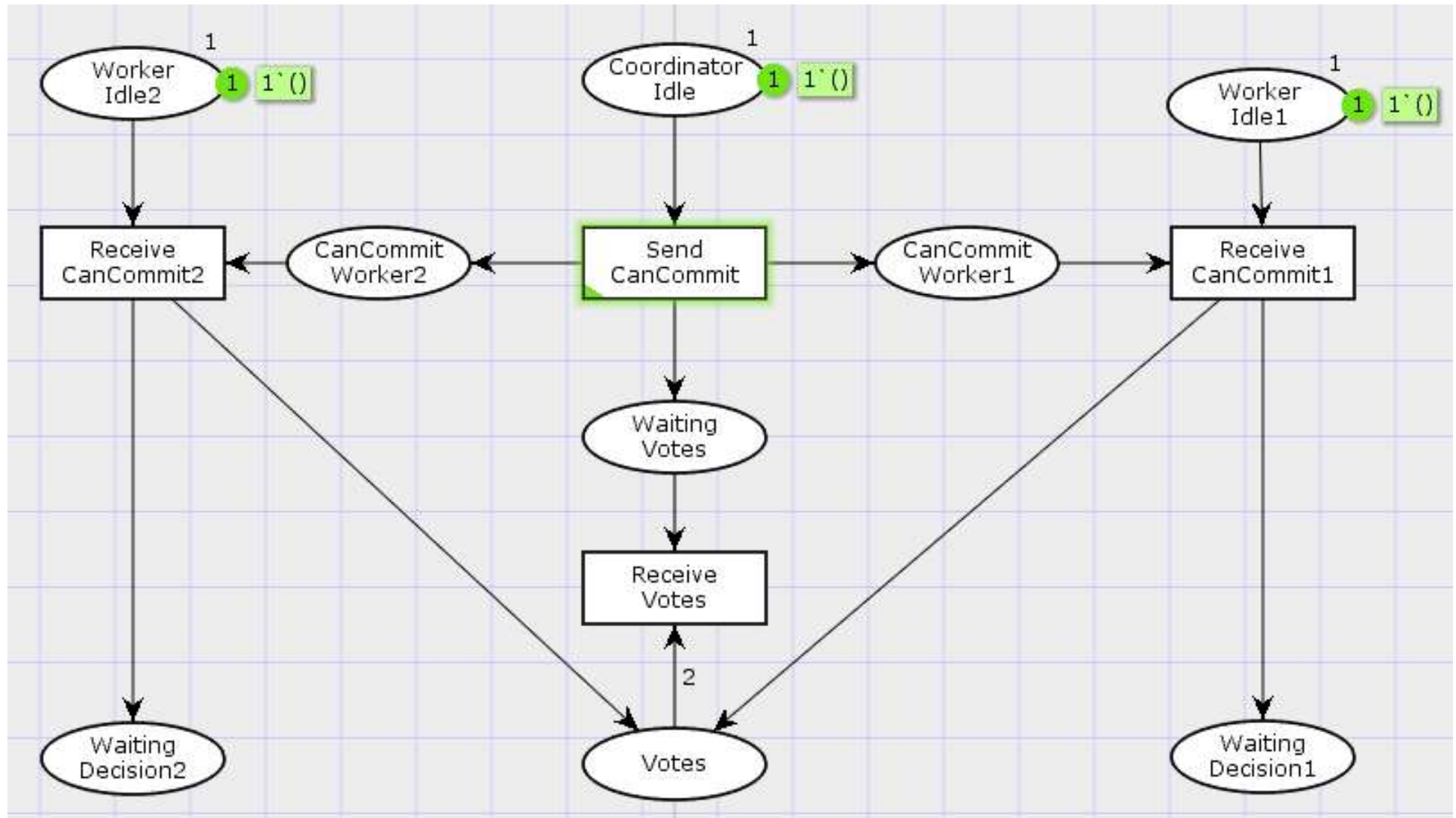
Modelling Votes ?



Multiple Workers ?

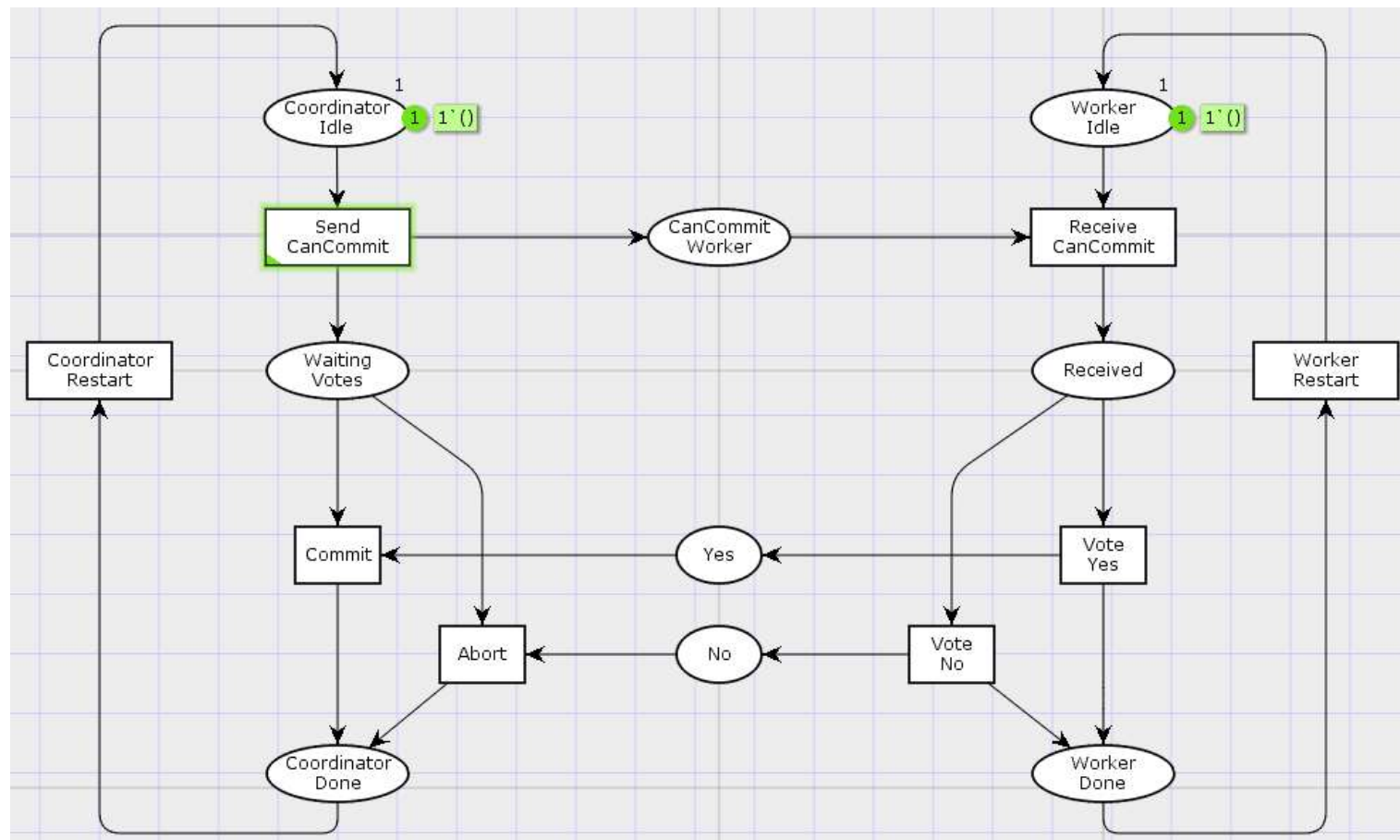


Receiving Votes



Reactive Systems

- Many concurrency systems are intended to be continuously operating



Why do we need CPNs ?

- **CPNs include the basic syntactical and semantical concepts of Place/Transition Nets.**
- **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