# Petri Net Modelling of Different Strategies for ARGESIM Comparison C4 'Dining Philosophers' with MATLAB / PetriSim

**Martin Kirner, Felix Breitenecker, Vienna Univ. of Technology; martin.kirner@gmx.at**

**Simulator:** This comparison solution was performed with a MATLAB toolbox "PetriSim", which is freely available. This toolbox offers a GUI for modelling classical and timed state / transition Petri nets and three operation modes: net analysis (P/T invariants, coverability tree, etc.) for S/T nets, simulation with conflict resolution strategies for S/T nets, and time simulation with conflict resolution, prioritisation and control of firing sequences for timed S/T nets.

**Model:** The basic model is realized using two nodes for each philosopher. The first node represents the thinking philosopher and the second the eating one. Between two neighbouring philosophers is also a node, indicating a free chopstick. The transition between thinking and eating can fire if the node [thinking] and the left and right [free chopstick]-nodes are marked. If the transition at the node [eating] is firing, the three nodes described before, will be marked again. Analysis shows the expected results, and the Simulation Mode gives the expected firing sequences.

**Refinement of the model.** The first step of the refinement was to add a new node *hungry*. Now if a philosopher is [*thinking*] he will change to the node [*hungry*], before he is able to eat.

The second step was to clean the chopsticks before they are available again. This has been resolved by adding two extra nodes between eating and the left and right chopstick.

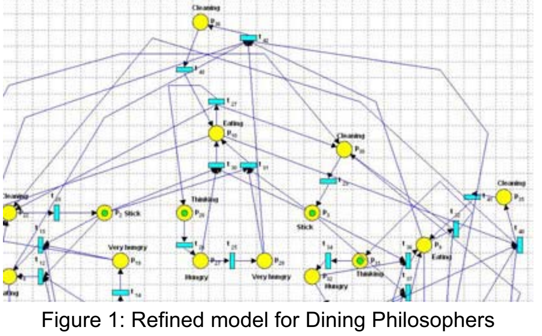

Figure 1: Refined model for Dining Philosophers

And the last task was to enable the philosophers communicate with their neighbours in the case he is very hungry (request token). Therefore, an extra node [*very hungry*] was introduced and a philosopher changes from [*hungry*] to this node if he is getting very hungry.

Than there are two options to start eating. The first is the same as at the node [*hungry*] and the second is contacting his neighbours if they are eating.

Every neighbour has the possibility to continue eating or to fire a transition to the very hungry philosopher and release the chopsticks in favour of him. The refined model is given in Figure 1.

**Different strategies:** In Figure 2 a screenshot of a conspiracy of two philosophers against the poor between them is shown. To simulate this situation a "synchronisation" node is introduced. A mark is added to the synchronisation node if one the conspirators start eating; is removed if one of the conspirators decides for thinking again. At the initialisation state the synchronisation node is marked and one conspirator is in state [*eating*]. The result is that the conspirators are eating alternately and the poor man in the middle never gets two chopsticks and has to starve.

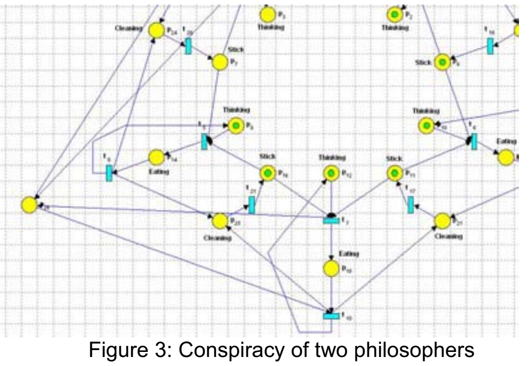

Figure 3: Conspiracy of two philosophers

Let's now consider a more fair approach: For this we assume a fairness pact is decided between all philosophers. To implement this, two synchronisation nodes are added to each philosopher. A philosopher could only start eating if these two nodes are marked. If a philosopher starts eating one node of each neighbour will be marked. This guarantees that all philosophers starts eating for equal times (Figure 3).

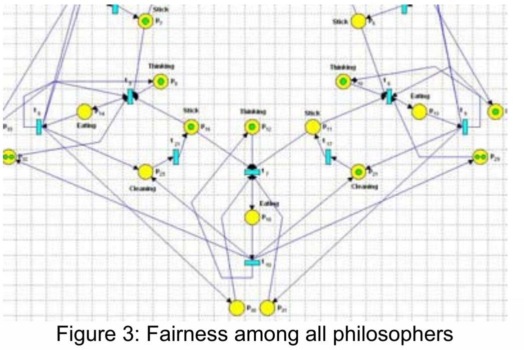

Figure 3: Fairness among all philosophers

**C4 Classification: Petri Net Approach**

**Simulator: MATLAB Rel.14, PetriSim Toolbox**