Petri Net Modelling of Different Strategies in ARGESIM Comparison C4 'Dining Philosophers' with HPSim

Michael Szell, Florian Judex, Vienna University of Technology, Austria; *michael@szell.net*

Simulator: *HPSim* is a simple Petri Net Simulator developed by Henryk Anschuetz in the context of a study. It has a graphical editor which provides basic editing and simulation of Petri Nets, and is useful for beginners - such as students - in order to get familiar with Petri Nets. The program is a win32 application developed in C++; it can be downloaded for non-commercial purposes at WWW.WINPESIM.DE free of charge.

HPSim provides an editor, a simulator offering token game animation and CSV file outputs, and is capable to simulate non-coloured place-transition networks with different kinds of edges and transitions (including means to stochastic simulation). Time units are given in milliseconds.

Modelling. The classic *Dining Philosophers Problem* describes a group of five philosophers sitting around a dining table, competing for food. There are five chopsticks on the table, each one between two philosophers.

When a philosopher gets ready to eat after some time of meditation, he first takes the chopstick on his right side, then the left chopstick, and begins to eat. Once finished, the chopsticks are laid back to their original places and the philosopher falls back into meditation. Since there exist only five chopsticks for five philosophers, conflicts are to be expected. This model serves as an analogy for competing computer processes and as the standard example for deadlock.

Figure 1 shows this basic model implemented in HPSim's model editor.

Simulation of Classic Model. For simulation, the classic model (Figure 1) has to be parametrised.

Meditation spans are implemented via stochastic transitions (hollow rectangles), all uniformly distributed with a range from 0 to 4 ms. Every philosopher takes 1 ms to eat; all other transitions are of type *Immediate* (black rectangles).

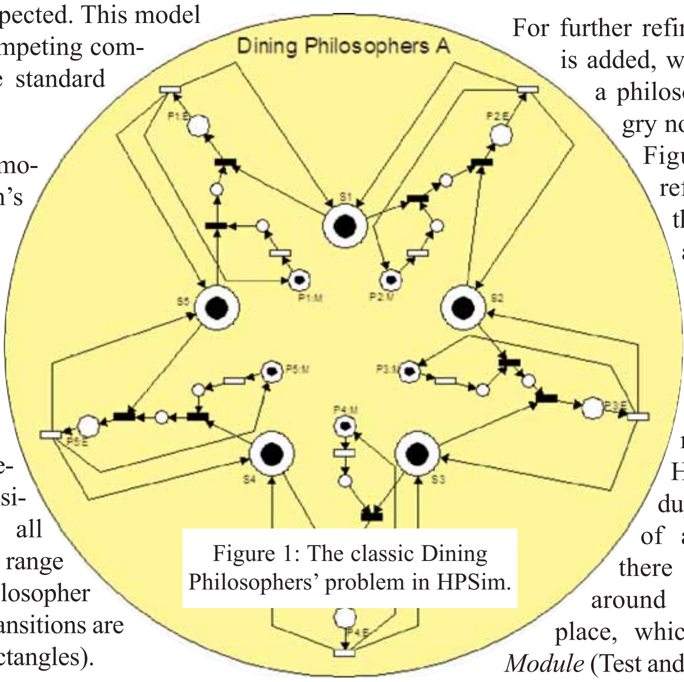

Figure 1: The classic Dining Philosophers' problem in HPSim.

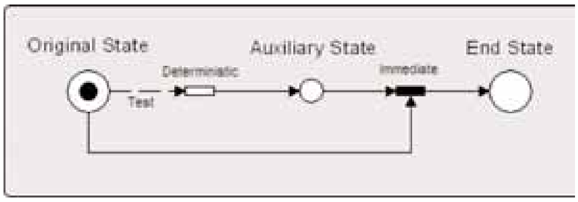

Figure 2: TF Module (for Test and Firing).

As is well known, sooner or later a deadlock situation takes place, where all five philosophers grab the sticks to their rights at the same time and wait for the left ones to become available - forever. At this time, HPSim stops the simulation with the exception 'Error: Deadlock at Time:'. The deadlock is detected when it occurs, but HPSim does not provide means to preemptive deadlock prevention or alge-braic analysis - it is a rather simple tool, well suited for relatively simple problems and great for visualisa-tion through its flow-animation eye candy.

Hunger and Death. For a first refinement, the state *hungry* is introduced. Whenever a philosopher stops meditating, an additional token is fired onto a *hungry* place. Once he has snatched both sticks, the hungry state releases the token. So now, a philosopher is always in one of these three main states: *meditating*, *hungry*, or *eating*.

For further refinement, a *dead* state is added, which gets filled once a philosopher has been hungry non-stop for 150 ms. Figure 3 shows these refinements displaying the *dead* states inside a red circle.

Implementing the test for 150 milliseconds together with a firing in one place is not possible in HPSim, since *Test* arcs duplicate tokens in case of a match. However, there is a simple workaround using an auxiliary place, which we will call *TF Module* (Test and Firing), see Figure 2.

SNE 17/1, April 2007

Cleaning the Chopsticks. In this variation, chopsticks get dirty after use. The dirty sticks are laid back on the table, and have to be cleaned by the next person who grabs them. This process is realised by changing the *Immediate* transitions to *Deterministic*. To gather more interesting results, three philosophers need 6 ms for the cleaning process, while the other two only require 2 ms. This inequality leads to starvation and three dead philosophers at the end of the simulation.

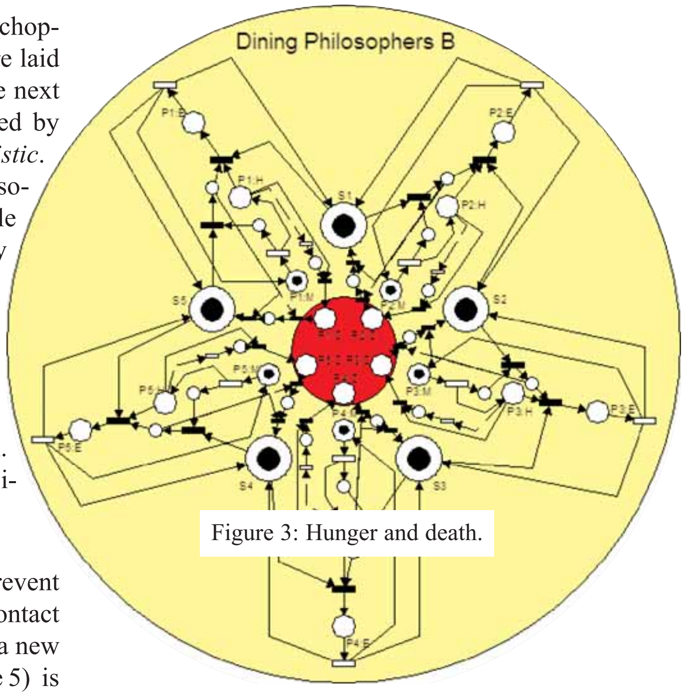

Figure 3: Hunger and death.

Conspiracy. Here, all philosophers need 2 ms for cleaning, but two decide to conspire against the one sitting between them. HPSim provides *Inhibito*r arcs, enabling inhibition of transitions.

Big Hunger and Communication. To prevent starvation, philosophers are now able to contact their neighbours in case of big hunger. For this, a new state - *very hungry* - (orange circles in Figure 5) is introduced, which is getting filled after 30 ms of hunger.

A very hungry philosopher takes the right chopstick immediately after his right neighbour has dined; the left chopstick is given to him by his left neighbour right after cleaning. This strategy prevents starvation and death.

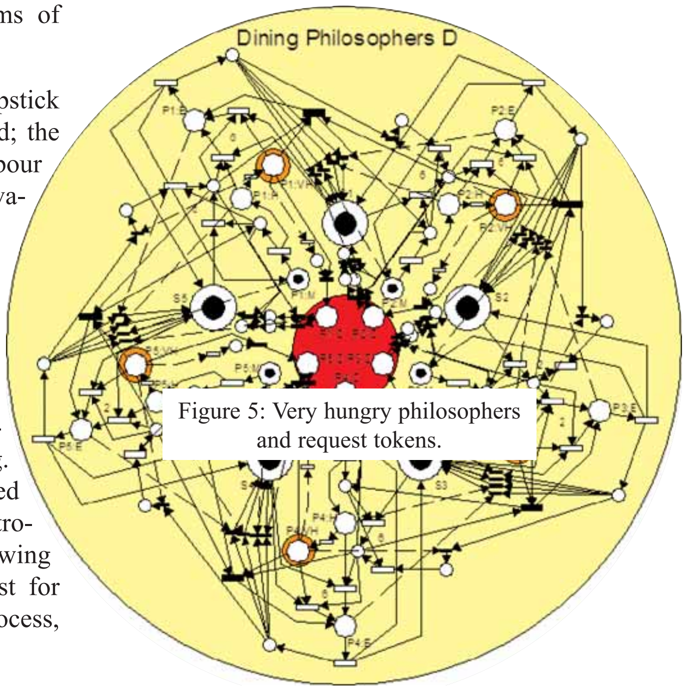

Figure 5: Very hungry philosophers and request tokens.

Resume: The Dining Philiospers problem is analysed by a classical Petri modelling approach using HPSIM, a simulator for simple Petri net modelling and simulation a introductory level. Time is introduced by means of holding transitions which consume time until firing. Simulation of the basic model shows the expected deadlock occurence. The model is refined by introducing additional states *(very) hungry*, allowing deadlock prevention and strategies like request for chopsticks, and by implementing a cleaning process, allowing control for starvation and conspiracy.

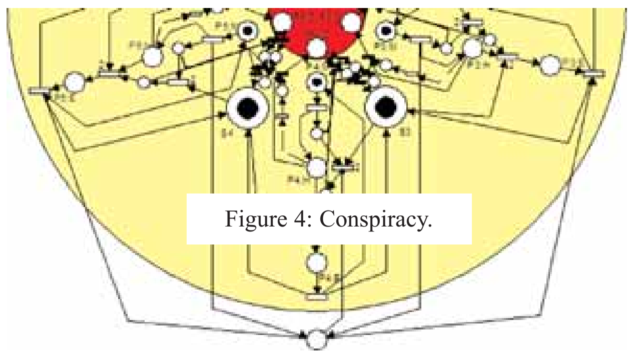
Figure 4: Conspiracy.

Corresponding Author: Michael Szell
Michael Szell, Florian Judex
Vienna University of Technology, ARGESIM -
Inst. f. Analysis u. Scientific Computing,
Wiedner Hauptstrasse 8-10, 1040 Vienna, Austria;
michael@szell.net

Reveived: November 3, 2006
Revised: November 25, 2006
Accepted: December 10, 2006