EUROSIM - Simulation News Europe started a series on comparisons of simulation software. Based on simple, easily comprehensible models special features of modelling and experimentation within simulation languages, also with respect to an application area, shall be compared.

The idea has become quite successful. Here we would like to thank all the authors who took the challenge and the time, solved the problems, documented them and sent in their contributions.

In the future we plan to present a new comparison in the March and in the November issue each year. If you have an idea for a model to be compared in different simulation languages please contact the editors.

We invite all institutes and companies developing or distributing simulation software to participate in this comparison. Solutions of comparisons 1, 2 and 3 described in the previous issues will still be published.

Please, simulate the model(s) and send a report to the editors in the following form (on diskette, any word processing format):

- short description of the language
- model description (source code, diagram, ...)
- results of the tasks with experimentation comments
  1 page A4

Comparison 4: Dining Philosophers Problem

In this issue we introduce a more sophisticated software comparison. It is a discrete one, but there exist different approaches for modelling, analyzing and simulating the process under investigation.

In the old times there lived people, who had time to think over principles of the world, who were wise and who consequently influenced politics and morality and daily affairs: the philosophers. They had a lot of experience, they did not depend on trivial needs like women, money, friends, etc.

Usually they spent day and night together, discussing problems and trying to solve them. Sometimes they thought over problems without any discussion, they were meditating - and they forgot the world around.

But there is one thing they depended on: they had to eat sometimes a little (because thinking is a hard job, and it takes more energy to think over a complicated problem and to solve it than digging a hole through the earth ball from Austria to Australia). The philosophers usually ate simple food, because they were not longing for more, and they gave away all the money they earned in their lives in order not to be dependent on anything - so they were poor.

But eating sometimes caused big problems, and it is said in the old tales that a group of philosophers was starving - although they had full plates.

In order to become aware of this phenomenon, one has to know a little bit about the habits of philosophers. They usually meet in groups of five sitting around a large round table. Supposing they are Chinese philosophers with a bowl of Chinese food in front of them they are able to start eating whenever they want to with their bowl being filled frequently. But being poor they have only five chopsticks lying left to the bowls - and for eating Chinese food one needs two chopsticks (figure 1).

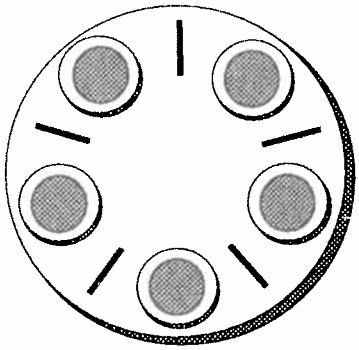

Figure 1: Philosopher's table

If a philosopher is hungry now, he first takes the chopstick to his left, then the chopstick to his right and he starts eating. Then he puts the chopsticks back on the table and starts thinking or meditating again.

It may now happen that every philosopher becomes hungry at the same time, and the five philosophers simultaneously take the left chopstick (when they are meditating, they forget the world around and so they do not become aware of the impending difficult situation), and then they try to take the chopstick at their right - but there is none. Their inflexible behaviour would cause them to starve. And that happened - according to the old tales.

In order to help the philosophers, one could now build up a model of their behaviour and could develop strategies in order to prevent them from starving, etc. The question is whether a philosopher would permit modelling and simulation as an appropriate problem solving tool: is it allowed to use models for gaining new knowledge? Some philosophers affirm this (e.g. Jaques in /1/).

By the way, this dining philosophers problems was first investigated by Dijkstra (/2/) and demonstrates the situation of parallel processes in a computer system which have to share resources - so the problem is not only sophisticated, it is more than relevant.

The problem of the dining philosophers is frequently discussed in the literature (e.g. /2/, /3/, /4/).

One approach to the problem is a description with Petri nets (/3/, figure 2). This approach allows an analytical investigation of the process. The following figure shows one possible Petri net model: Each philosopher may be represented by two places (M_i and E_i) representing the meditating and eating state, resp. Places C_i represent the chopsticks. In order to move from the meditating state to the eating state, both chopsticks (the one at the left and the one at the right) must be available for a philosopher. It depends on the initial marking of the places, whether the problem is solved or causes a deadlock situation. In fig. 2 one philosopher is eating (indicated by a token residing in the place E_1), and the others are meditating (indicated by tokens in the places M_i ($i=2,3,4,5$) so that three chopsticks are free (indicated by tokens in the places C_2, C_3 and C_4). After having finished eating one neighbour may start eating as well as a philosopher who is not neighbouring one of them. Of course exact firing rules have to be defined.

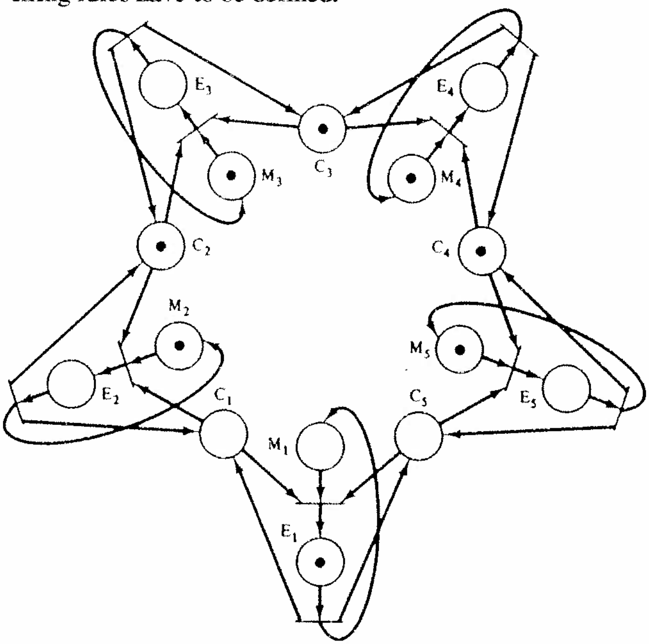

Figure 2: Petri net approach

Another approach is the simulation of the model in the time domain, for instance on the basis of process description or on the basis of delayed Petri nets. Once again the processes may be eating, meditating, etc.

Simulationist (especially 'discrete' ones) are invited to model and analyze and/or simulate the dining philosophers problem with the software tool of their choice - either on the basis of network analysis or on the basis of process (or event-) orientated simulation or on the basis of a mixed software tool. In contrary to the previous comparisons no fixed taks are asked to be solved or investigated. In order to assure comparable results only certain rules should be obeyed in refining the model within four steps (/4/):

a) First basis of the investigations is the Petri net model in fig.2., with the forementioned assumptions. In case of simulation suitable random time delays for the processes and initial conditions should be fixed.

b) For a first refinement a third philosopher's status "hungry" is to be introduced.

c) If a philosopher gives back the chopsticks, they are available, but they are dirty. So a cleaning process is to be introduced.

d) The philosophers should be able to communicate with their neighbours, if they are very hungry (request token).

For experimentation at least two situations should be considered:

1) Two basic situations (experimental conditions) should be investigated: 1.1) normal conditions (no conflicts, no deadlocks), 1.2) deadlock (starving)

2) Although the philosophers are wise, it is a situation involving human behaviour. One or two (or three) philosophers could make a conspiracy against the others (against one, two or three) to let them starve. Different strategies should be implemented.

Please feel free to analyze and/or simulate other interesting situations or to invent tricky strategies for whatever purpose (e.g. 'friendly' strategies in order to prevent from starving) or to use different software tools (analysis and simulation). We would also be glad to receive demonstration disks (with animation), which will be offerd to interested readers on request.

Hoping not to interfere with your Christmas holidays we ask you to take the challenge of this sophisticated comparison. It seems to be an appropriate problem suiting the season.

The EUROSIM - Simulation News Europe Editors.

References

/1/ Shakespeare W. The Globe Illustrated Shakespeare. Ed.H.Staunton (1810-1874). Edition 1983, Greenwich House, Inc., New York; p.894, 2nd column, lines 39-40.

/2/ Dijkstra E. 'Cooperating sequential processes' in F. Genuys (ed.), 'Programming Languages', New York: Academic Press, New York (1968), p. 43-112.

/3/ Peterson J.L. Petri Net Theory and the Modelling of Systems. Englewood Cliffs: Prentice-Hall (1981), ch.3.4 Computer Software.

/4/ Chandy K.M., Misra J.: Parallel Program Design - A Foundation. Edison Wesley (1988), chapter 12.