Comparison 10: Dining Philosophers II

In SNE 3 the Dining Philosophers problem was formulated as (discrete) comparison C4. Dijkstra [1] had been the first to investigate this problem - which is widely known today - from the perspective of computer science, demonstrating the situation of parallel processes in a computer system which have to share resources. The problem is therefore not only sophisticated, it is more than relevant. It has also been frequently discussed in the literature (e.g. [1], [2], [3]).

The Dining Philosophers' problem is relatively easy to describe, but the philosophers' behaviour may cause interesting problems, including especially concurrent access and deadlock situations: Five philosophers are sitting around a large round table, each with a bowl of Chinese food in front of him. Between periods of meditation they may start eating whenever they want to, with their bowls being filled frequently. But there are only five chopsticks available, one each to the left of each bowl - and for eating Chinese food one needs two chopsticks (figure 1).

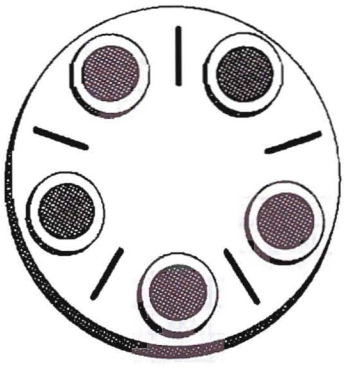

Figure 1: Philosophers' table

When a philosopher wants to start eating, he must pick up the chopstick to the left of his bowl and the chopstick to the right of his bowl. He may find, however, that either one or both of the chopsticks are unavailable as it/they is/are being used by the philosopher(s) sitting on his right and left, so he has to wait. Hopefully, none of the philosophers is starving and, for convenience, they may agree on certain strategies to cope with unforeseen situations.

C4-Definition in SNE 3

The previous definition of this comparison [4] did not postulate fixed modelling techniques or tasks to be performed. It proposed different approaches to modelling, analysis and simulation, e.g. event-oriented modelling or Petri nets. Experiments of any kind were appreciated: application of different strategies, model extensions by introducing "waiters", net and deadlock analyses, etc. To date, seven very interesting solutions have been received, which really demonstrate the expected variety in the approaches employed: Petri net tools, discrete simulators, interesting experiments, etc. [5 -11].

The problem is that these solutions cannot be compared with each other, and this comparison therefore cannot be evaluated in the same way as the other comparisons [12]. However, since there is substantial interest in this problem, it was decided to reformulate the comparison, restricting it to simulation in the time domain and fixing certain tasks.

C10: Definition (Redefined C4-Definition)

The redefined comparison reviews i) modular and / or object-oriented model descriptions, ii) simultaneous access to resources, and iii) detection of deadlocks.

The first assumption is that the philosophers agree that a hungry philosopher will first take the chopstick to the left of his bowl, and, once he has picked up this chopstick (maybe after some waiting), will try to get the chopstick to the right of his bowl (which may be available immediately or after some waiting).

But even this strategy will not prevent the following two situations from happening, in which any further action (meditating or eating) becomes impossible.

1. Simultaneous access to one chopstick: one of the hungry philosophers (having already got hold of the chopstick to the left of his own bowl) wants to take the chopstick to the right of his bowl at exactly the same time instant as his right neighbour (who is also hungry) is about to pick up the chopstick to the left of his bowl (which is the chopstick to the right of his left neighbour's bowl).
2. Deadlock: It may also happen that all philosophers are hungry, and, by chance, each philosopher has picked up the chopstick to the left of his bowl and is waiting for the chopstick to the right of his bowl to become available - which will never happen. This situation is very rare, but it does occur !

Both situations are critical points for simulators. On the one hand, the model descriptions (which are usually higher than event descriptions) are comfortable, yet their translation to the event level may differ considerably from simulator to simulator. In the cases investigated the user usually does not know a) how the simulator handles simultaneous tasks; how the event

scheduler (on which every discrete simulator is based) deals with simultaneous events (some event lists use FIFO techniques for the events, some for the entities); and b) whether a deadlock can really be detected immediately, or only some actions later, or whether the simulation will continue forever ?

In order to explore the simulators' features with respect to simultaneous access and deadlock detection, the following strategies and conditions must be observed:

i) Time for thinking and eating follows a discrete uniform distribution in the interval [1, 10] (the assumption of natural numbers guarantees the occurrence of simultaneous access and deadlock situations)

ii) All five philosophers start with a "thinking period".

iii) In a simultaneous access situation the philosopher sitting on the right gets the chopstick (to the left of his bowl) first and the philosopher to his left must wait (although he has already taken one chopstick and could start eating).

iv) When a deadlock occurs the simulation must terminate. (A deadlock is rarer than simultaneous access, so it is assumed that one or more simultaneous access situations will happen before a deadlock occurs.)

Any proposed solution should include, as an introduction, a brief summary of the simulator's features and the advantages of the modelling technique (object-oriented, modular etc.), followed by a model description of the Dining Philosophers' problem in the simulator's notation or syntax (textual and/or graphical). As a full description may be too long, a rough outline may do for some of the aspects whereas the elements (blocks, tasks, functions etc.) proposed to solve the conflict of simultaneous access and the elements required for detection of a deadlock should be defined in detail.

Two tasks are to be performed:

i. Single simulation run until a deadlock is reached

a) giving the average times (with standard deviation) of thinking, waiting and eating periods for each philosopher and all of them together, and rate of chopstick utilisation (individually and all together);

b) demonstrating the correct management of simultaneous access e.g. by documenting the status of the event queue when such a situation occurs (debugging of current events and entity movements at time instant of simultaneous access, including resolution of simultaneous access).

ii. Performing at least 50 simulation runs (ending in a deadlock !), indicating the maximum and minimum termination time. Indicate, how the deadlock is detected by the system and/or how the model has to be extended in order to terminate with a deadlock.

Please keep in mind that solutions must fit onto one page. A template solution is provided in this issue of SNE [13], another template solution may also be found on the ARGESIM WWW server (http://argesim. tuwien.ac.at/comparisons/). Further solutions that might serve as examples can be found in [14 - 16] (in German). We hope, that many simulationists (and philosophers) will accept the challenge of solving this comparison.

References

[1] Dijkstra E.: Co-operating sequential processes in F. Genuys (ed.), Programming Languages, New York: Academic Press, New York (1968), p. 43-112

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

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

[4] SNE Editors: Comparison 4: Dining Philosophers Problem. EUROSIM - Simulation News Europe, Number 3 (1991), p. 28

[5] Henriksen J.O.: Comparison 4 - GPSS/H. EUROSIM - Simulation News Europe, Number 4 (1992), p. 43

[6] Page B., Martinssen D.: Comparison 4 - DESMO. EUROSIM - Simulation News Europe, Number 4 (1992), p. 44

[7] Ruzicka R.: Comparison 4 - SIMUL_R. EUROSIM - Simulation News Europe, Number 5 (1992), p. 37

[8] Starke H.P.: Comparison 4 - PAN. EUROSIM - Simulation News Europe, Number 6 (1992), p. 33

[9] Grossenbacher Elektronik: Comparison 4 - PACE. EUROSIM - Simulation News Europe, Number 6 (1992), p. 34

[10] Behrens A.: Comparison 4 - POSES. EUROSIM - Simulation News Europe, Number 7 (1993), p. 34

[11] Wintz S.: Comparison 4 - NETLAB. EUROSIM - Simulation News Europe, Number 7 (1993), p. 35

[12] Breitenecker F.: EUROSIM Comparisons - Dokumentation und Ergebnisse. 10. Symposium Simulationstechnik, Dresden 1996, Hrsg.: W. Krug. Fortschritte in der Simulationstechnik, Vieweg (1996), p.537

[13] Klug M., Breitenecker F.: Comparison 10 - GPSSH. EUROSIM - Simulation News Europe, Number 18 (1996), p. 34

[14] Schmidt B.: Das 5-Philosophen-Problem und die Modellspezifikation mit Simplex II. 10. Symposium Simulationstechnik, Dresden 1996, Hrsg.: W. Krug. Fortschritte in der Simulationstechnik, Vieweg (1996), p.537

[15] Henriksen J.O., Preusz F., Schulze T.: Simulation des 5-Philosophen-Problems in SLX. 10. Symposium Simulationstechnik, Dresden 1996, Hrsg.: W. Krug. Fortschritte in der Simulationstechnik, Vieweg (1996), p.551

[16] Rüger M.: Dining Philosophers - Create!. 10. Symposium Simulationstechnik, Dresden 1996, Hrsg.: W. Krug. Fortschritte in der Simulationstechnik, Vieweg (1996), p.557

F. Breitenecker, ARGESIM, Dept. Simulation Techniques, TU Vienna, Wiedner Hauptstr. 8-10, A-1040 Vienna, Email: Felix.Breitenecker@.tuwien.ac.at
B. Schmidt, Universität Passau, Inst. f. Operations Research & Systemtheory, Innstraße 33, D-94032 Passau, Tel: +49-851-509 3080, Fax: +49-851-509 3082