Education and Training in Computer Simulation: a Way to Nested Simulation

Ivan Krivy and Eugene Kindler

Abstract: *This paper deals first with the results of training students in computer simulation at the Computer Science Department of the University of Ostrava during the last 15 years. Education and training in computer simulation are based on an obligatory course of SIMULA taking two semesters. The special attention is paid to implementing nested (especially reflective) simulation models of production and logistic systems for enterprises and institutions outside educational system.*

***Key words**: Object-Oriented Programming, SIMULA language, Nested Simulation, Reflective Simulation, Education and Training.*

INTRODUCTION

Computer simulation of complex systems is a special (and most frequent) case of their computer modeling. There are thousands of such simulation studies. Some of them concern simulation of information systems but a greatest number concern "hard" systems like production ones, logistic ones, environmental ones, medical care ones, service, organisms, organs etc. The hard systems are composed of elements that are realized as "touchable" physical entities, i.e. as something that is viewed as made of solid or liquid substance, or – in some extreme cases – of gas. Such elements are viewed to perform only very limited information processing. The information systems are sometimes viewed as performing a lot of information processing, but into that information processing simulation is never included. In other words, simulation of information processing is never made so that the simulation systems would be viewed as having simulating elements.

The computerization and informatization of the society tends to the state that the hard systems designed by human society will be also computerized. Namely, the epoch to design man-made systems *S* containing one or more computers *C* that will perform simulation when *S* will physically exist, is near. When such a system *S* will be designed it will often be simulated, because the designers will want to get some data about the behavior of the designed variant and according to these data they will be decide to optimize the definitive form of *S*. In [3,6,9] we pointed that in such a case the simulation model of *S* used during the design of *S* should reflect not only the entities of *S* that are viewed as hard, but also the computer(s) *C* with its simulation phases. So we meet the ***nested simulation***, i.e. the simulation of systems that contain simulating elements, or – told in a bit short for – we come to simulation of simulation. If the nested simulation has a form that the computers simulate just *S* we speak on ***reflective simulation***. Evidently, it is a special case of the nested simulation.

PRINCIPLES OF TEACHING PROCESS

In this paper we concentrate on the teaching process concerning the nested (reflective) simulation. Since the beginning of teaching simulation (since the end of the eighties), the principles of that process have been as follows:

a) We are using language SIMULA (the reasons for that choice were presented in [8]).

b) The students were given an obligatory two-semester course of computer simulation based on SIMULA. More details on the course will be given during oral presentation.

c) At the end of the course the principles and programming tricks of nesting simulation models implementation were told the students.

d) The students who wanted to finish their studies by orienting their diploma theses to the simulation have got a certain theme to program a model and to formalize a domain analysis of the model. Such analysis arises almost automatically in parallel with the process of the stepwise formulation of the model in SIMULA and with its stepwise refining. Some of graduate students continue in constructing and applying simulation models (especially nested ones) during their doctoral studies.

QUALIFICATION FOR IMPLEMENTING NESTED SIMULATION MODELS

As one can see, at the beginning phase the students were not forced to implement models applicable in nested simulation. Nevertheless, they were led to simulate rather complex hard systems such as:

- crop works in greater farms: systems composed of combined harvesters, storages and trucks that transport the harvest from the harvesters to the storages,
- sowing works in greater farms: systems composed of storages, sowing machines and trucks that transport the seed corn from the storages to the sowing machines,
- ageing of machines used in greater farms: generation of data supporting the decision making whether a machine should be seen as repairable or replaced by a new one,
- vertical matrix storage served by a conveyor: generating various control algorithms governing the conveyor and their testing relating to the possible flow of requirements,
- pork meat production: systems of halls where pigs of different age are breeded and where a cycle of making sows in-pig – pigging exist, possibly with decisions what sow is to be replaced by a new one; the halls are served by human workers who respect a certain week-based operating rhythm,
- veterinary sanitation in regions: systems of sources of biological waste (slaughterhouses etc.), a center for processing the waste and trucks transporting the waste from the sources to the center with respect to governmental laws concerning the hygiene,
- models of governmental production made by interacting regions.

Some models were stepwise prepared by us (i.e. by the authors of this paper) and used as demonstration ones informing the students on the large spectrum of simulation:

- machine production: various systems reflecting occasional demands coming from Czech enterprises,
- washing of contaminated textiles in nuclear power stations,
- transport of radioactive waste from nuclear power plants to subterranean disposal site: testing of various algorithms for configuration of transport frames carrying containers,
- solidification of radioactive waste: testing of various methods of the solidification with respect to accidents and to irradiation of the human staff,
- local epidemics,
- dissemination of insect pests in forest,
- forest fires,
- potassium ion dynamics in the environment of neural cells,
- use of beds by patients in hospitals,
- iodine transport in human organisms,
- airplane wing assembly: optimization of the order of assembly steps,
- fault systems of large parts of nuclear power stations.

All the models mentioned above were applied in enterprises outside the universities. Beside them, we also prepared some demonstration non-simulation methods:

- two-dimensional Cartesian geometry – concepts of point, circle, incise, rectangle, *etc*.; relations „a point is inside (or at the boundary of) a circle (rectangle, …)" *etc*.,
- vector and matrix algebra – concepts of n-dimensional (numerical) vector and matrix together with their operations,
- complex number arithmetic,
- foundations of chess play – concepts of component of the chessboard, chessman, its specialization to particular sorts of chessmen, pane, *etc*.,
- geometry of freely placed objects – concepts of circle, rectangle, cylinder, cone, rectangular parallelepiped, etc., three-dimensional concepts being declared as specializations of two-dimensional ones.

Beside these models, we implemented also so called ***fictitious simulation*** ones, i.e. models of fictitious systems, which serve to fast implementation or modification of complex non-simulation algorithms:

- configuration of chess-queens,
- sheep-wolf play,
- escape from a maze,
- shortest path in a transport network,
- quasi-parallel processing of several files into one file, depending on special properties of the data occurring in the files,
- processing of arithmetic and logical expressions into systems of computing and interacting blocks,
- synthesis of free rhythm structures: experiments with generating first millennium European melodies according to certain hypotheses on their rhythm.

The fictitious systems are supposed to produce the same data as the intended algorithms.

OVERVIEW OF IMPLEMENTED NESTED SIMULATION MODELS

In the nineties students were acquainted with certain cases of nested simulation, up to this time only in a passive mode (they were not forced to engage in implementing such models). The principle was as follows: using a simulation model of a realistic system, fictitious simulation was applied, i.e. a model applied in a fictitious simulation was included into a model of a realistic system. The examples of such nested simulation models are as follows:

- simulation of a machine production system that applies induction carriages for operation transport (the fictitious simulation was used for the shortest path computing),
- simulation of interregional product exchange in Czech republic (the fictitious simulation was used for inserting geometrical figures to the tables of the simulated results),
- container terminal with ground-moving operational transport tools (e.g. forklifts) that move at the places free of containers (the fictitious simulation was used for the shortest path computing).

All three studies mentioned above were made for institutions belonging to other fields than education. The last of them was implemented under the project COPERNICUS-312 AMCAI managed under European Commission and coordinated by Center of Fraunhofer Institutes in Munich (Germany).

Since 1998 students were acquainted with reflective simulation, namely with the following models:

- queuing system (a bank) with one or more dispatchers who decide on the number of employees serving the customers at the tellers [4]: when a dispatcher decides to change the number he simulates what could happen in the near future in order to test whether his decision is not wrong;
- container terminal with ground-moving operational transport tools [1,5]: the nested simulation was used for testing whether the computed shortest path will not lead to a conflict with an object that was not place at its place when the path was computed but was transferred there during the initial phase of the moving according to the computed path. Also this model was applied under the project COPERNICUS DAMAC-HP;
- cyclical conveyor with free and motorized rollers [7]: simulation solution of various problems concerning its behavior after faults and in case of too high frequency of demands; the implementation was made for a French enterprise;
- simulation of urban traffic in the town Havířov [2]: the human planning made by the passengers and using their imagining of the future moving along bus traces and their potential sequencing was modeled by simulation.

CONCLUSIONS AND FUTURE WORK

Since the end of the fifties, simulation has been a very efficient technique for applied computer science, stimulating both mathematical modeling and programming paradigms. Since the fifties, simulation has passed trough a rich development, the most important result of which is the reflective simulation, i.e. simulation of systems that apply simulation during their existence in order to anticipate their possible future states. Constructing simulation models for reflective simulation is far from the everyday simulation routine. The authors received interesting experiences in teaching simulation, so that nowadays certain university (undergraduate) students are able to construct and use reflective simulation models. The students were stimulated to use the object-oriented programming technique for constructing simulation models of more and more complex systems, so that in the last years the interested students were able to use sophisticated object-oriented programming tools (namely SIMULA) for implementing reflective simulation.

The next progress of reflective simulation is ensured by two aspects. The first one is a larger reflection of the reflective simulation in the students seminary works and theses, the second one consists in the automation of certain steps in constructing reflective simulation models.

Therefore, our future work will be oriented first to:

- enlarging the spectrum of simulation applications from the industrial systems to other ones, namely to the system related more to life sciences (health care systems, ecological ones *etc*.),
- designing, implementing and testing a translator that converts conventional (non-reflective) simulation models into those with reflective abilities.

REFERENCES

[1] Blümel, E. et al. *Managing and Controlling Growing Harbour Terminals*. San Diego, Erlangen, Ghent, Budapest: Society for Computer Simulation International 1997, 308 p. (Kindler, E. Chapter 4, Classes for object-oriented Simulation of Container Terminals – pp. 175-278) ISBN 1-56555-113-3.

[2] Bulava, P. Transport system in Havirov. In: Proceedings of 28th ASU Conference. University of Technology, Brno, 2002, pp. 57-62.

[3] Kindler, E. SIMULA and concurrent engineering. ASU Newsletter 21 (3), 1993, pp. 1-16. ISSN 1102-593.

[4] Kindler, E. Reflective Simulation (First Experiences). In: Simulation und Animation für Planung, Bildung und Präsentation '96. Magdeburg – Wien: ASIM 1996, pp. 39-50.

[5] Kindler, E. Object-Oriented Simulation of Container Yards. In: Simulation und Animation '97. Erlangen, Ghent, Budapest, San Diego: Society for Computer Simulation International 1997, pp. 111-122. ISBN 1-56555-111-7.

[6] Kindler, E. Simulation of systems with imagining components. In: Ch. Urban (Ed.): Workshop 2000 – Agent-Based Simulation. Ghent: SCS European Publishing House 2000, pp. 239-244. ISBN 1-56555-185-0.

[7] Kindler, E., Berruet, P., Coudert, T. Conveyors with Rollers and Their Reflective Simulation. In: Modelling & Applied Simulation 2003. Genoa University & Liophant Simulation Club & McLeod Institute of Simulation Science – Genoa Center: Genoa 2003, pp. 147-152. ISBN 88-900732-3-3.

[8] Kindler, E., Krivy, I., Tanguy, A. Reflective simulation of logistic and production systems. In: Modelling & Applied Simulation 2002, Genoa University & Liophant Simulation Club & McLeod Institute of Simulation Science – Genoa Center: Genoa, 2002, pp. 97-102. ISBN 88-900732-9-2.

[9] Křivý, I., Kindler, E., Tanguy, A. Towards Simulation of Simulating Enterprises. In: CompSysTech'03. Sofia: Bulgarian Acad. of Sci., 2003. pp. 151-156. ISBN 954-9641-33-3.

ABOUT THE AUTHORS

Assoc. Prof. Dr. Eugene Kindler, PhD., Department of Mathematics, University of Ostrava, Phone: +420 596 160 288, E-mail: evzen.kindler@mff.cuni.cz.

Prof. Dr. Ivan Křivý, PhD., Department of Computer Science, University of Ostrava, Phone: +420 596 160 273, E-mail: ivan.krivy@osu.cz.