Computer Science Journal of Moldova, vol.5, no.3(15), 1997

On the architecture of problem-oriented simulation systems

L. Burtseva

Abstract

The descriptions of three problem-oriented systems are presented. The general architecture of such systems is determined on the base of comparison analysis of particular problem-oriented simulation systems.

1 Introduction

On turning computers from "holly ground" of computer rooms into subjects of everyday use the universal software has forfeited its global value. All areas of computer science now rival to provide users with more friendly systems, requiring minimum of specific "computer" knowledge. Simulation, being at present one of the most broadly used areas of computer science, is not an exception. Opposite, problem-orientation was actively developed to the prejudice of universal methods even. This is explained by lack of close contacts between theoreticians researching the simulation method, software developers and practicians solving applied problems. Theoretical developments could prompt an user what class of problems can be easy and quickly resolved by the simulation method. Software developers tried to implement computer possibilities in simulation. But practicians, knowing that in principle their problems were resolvable by simulation, use "available" tools.

Group of researchers "Simulation systems", in which the author of this article works, during many years has devoted and researched general-purpose simulation tools, but herewith used this tools for solution of particular practical problems. Due to the cooperation with establishers of these practical problems, software development takes requirements of users into account. On the other hand our group of theoretical researchers tried to implement new theoretical solutions. Problem-oriented simulation system architecture was designed by the author as a result of generalization of this experience. Three simulation systems oriented into different areas were developed by the group according to this architecture [1]. One of them — a benchmark analysis of two schemes of compilation — was submitted for EuroSim95 international conference [3]. The other system — the study of credit-banking operations process — was successfully used in the education of management specialists.

2 Comparison of problem-oriented systems architectures

Let us suppose now that user does not want nor in the least to deal with entrails of computer system. Prepared by corresponding specialists executable module of model in such a case is a "black box". Suitable representation of model's input and output in terms of applied domain, providing dialog in such terms and representation of simulation results in approachable for this domain view create friendly system. Such system is really able to replace the natural experiment by simulation experiment for the researcher.

The brief descriptions of the systems are below.

2.1 System of different transport work simulation

The different transport work simulation system combines three simulation models the one interface. There are the model of working of a quarry, the model of working of a port, the model of the carriage by auto (all problem formulations were taken from A.Pritsker[2]).

2.1.1 Problem descriptions

a In the quarry dump-trucks of different cargo-carrying capacities deliver ore from excavators to a crush-machine. A dump-truck of higher capacity has higher priority.

b There are a bulldozer, loader-machines and finite number of dump-trucks in "carriage by auto" model. Dump-trucks are loaded, leave, unload and return. The whole soil prepared by the bulldozer for working day, must be removed. Changing the number of dump-trucks, it is possible to define their optimum amount to remove the whole soil under maximum utilization of the loader-machines.

c In the port tankers are loaded by oil, which then is delivered on the purpose. There is one tug in the port, that serves tankers when mooring and pushing off. The fact that the port is subjected to storms complicates the process of transportation of oil. During a storm the tug does not work.

2.1.2 Simulation system operation

The system allows to choose a model (supplied by the window with the description), to assign input parameters in the detailed dialogue, to run a series of experiments, to review results and to graphically analyze them. This system configuration and work are represented more detailed in Fig.1. Drawing of the objects by more thin lines with dotted links represent files which are "outside" system. The ovals represent information files, the rectangle represents executable module.

Main menu contains items which represent the stages of 2.1 system work. Model choice (in this case the model name choice) immediately supplies by window with model description. After parameters setting menu item have chosen dialog is provided in terms of chosen model problem. Experiment run item is the essence of a problem-oriented system. Received results user can review or analyze them — so there are two corresponding menu items. After analysis menu item have chosen its submenu is represented in terms of chosen model problem.

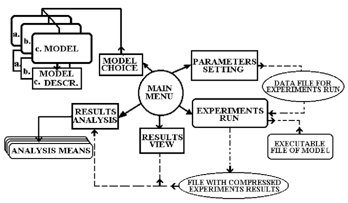

Fig.1. Structure and operation of 2.1.system

2.2 Comparative analysis of two schemes of compiling

2.2.1 Problem description

The system unites two models corresponding two schemes of compilation.

The first scheme corresponds to the traditional process of translation when the work is divided into 3 consequent phases: lexical analysis, syntax analysis, semantic analysis and generation of code. On termination of one phase the following one starts.

In the second scheme while a module, realizing one of the phases of translation, is working, collected information can turn out to be sufficient for working the module, realizing the following phase. In this case these modules work simultaneously. So the time of translation process is reduced.

For operation of translator from 1 to 7 processors are available. The user can fix processors numbers for translation phases himself. If any mistakes have been discovered at any phase, the process of compiling stops.

Input data of the model represent variety of user programs for translation. They simulate programs of different length written with different style and with different probability of presence of mistakes in them. After experiments with models run we get a general time of “programs” processing and of utilization coefficients for processors under different strategies of the use of resources. There is a possibility of comparison of translation strategies and distribution of resources.

2.2.2 Simulation system operation

This system configuration and operation are represented in Fig.2

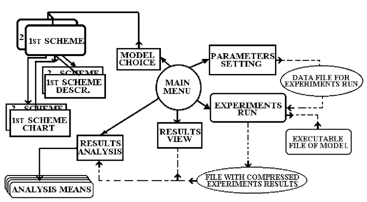

Fig.2. Structure and operation of 2.2.system

Main menu contains items which represent the stages of 2.2 system operation. Model choice (in this case the chart name choice) is supplied by submenu where the user can choose model description or model chart. Parameter setting menu item is succeeded by three sequential dialog screens to set distribution of the processors, user programs lengths and styles. Experiment run menu item is the next one. User can review or analyze obtained results, so there are two corresponding menu items. Analysis menu is added by means of comparison of two charts.

2.3 Simulation system of credit-banking operations

2.3.1 Problem description

In the simulation system of credit-banking operations a process of lending is present as a parallel performing of the following functions: issues of credits and their service (i.e. reception of percent, returning of the credit sum by the client or fixing of his bankruptcy). Bank issues credits, taking in consideration available at given time sum and in accordance with preferences of the system user (preference is the probability that the credit will be issued). At checktime bank ceases issue credits and only collects issued sums with percent.

2.3.2 Simulation system operation

This system configuration and work are represented in Fig.3

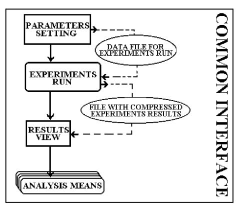

Fig.3. Structure and operation of 2.3. system

According to its business game character this system has not the main menu. Due simulation steps follow one another automatically. Sequential screens represent parameters setting, model run, results review. Some kind of menu is needed for choice of analysis means because of their variety. A number of graphical icons are added to results review screen to make such a choice.

3 Main aspects of problem-oriented simulation systems

The systems described above look very different. But as it was already said it is possible to select components general for any problem-oriented simulation system. So their different appearances are individual "bodies" stretched onto the common "skeleton". Simplified general architecture a of problem-oriented simulation system can be represented by the following scheme:

$parameters\ setting \Rightarrow experiment\ run \Rightarrow results\ review/analysis$

How this architecture has been extracted and how individual features have been formed is described below.

3.1 The setting of model parameters

Information at the input usually named "model parameters" is main information, got just from the user. Naturally, the processing of this information, its filtration and interpretation present the most difficulty. So it is necessary to supply information input by help as far as possible and to filter carefully the entered data to reduce to the minimum probability of technical and semantic mistakes.

For instance, in Fig.4. a snapshot of operation of a comparison analysis system of two schemes of compilation process is represented. "Standard" distribution of the processors for compilation phases will be assigned automatically. User can share or distribute resources by the change of the processors number. Attempt to enter a letter or number more than 7 is ignored.

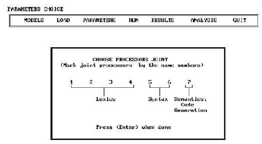

Fig.4. System 2.2 snapshot.

In the same way attempt to enter letters in windows of input of the bank system is ignored. Uncorrectness of input (i.e. excess of number of available tankers or dump-trucks) in the transport system is accompanied by a beep and an error message with the offer to repeat the input appears.

3.2 Model running — an experiment

As it was noted above, the essence of simulation system is replacement of natural experiment by the running of the simulation model. One, who has used simulation long since, knows that, even when a powerful computer is used, simulation model running requires long time (though short in contrast with the natural experiment). Problem-oriented simulation system developer, considering this problem, can choose one of the following ways of implementation:

a. To warn the user about long waiting time and to mark the experiments completion by some signal (this method is usable in the cases, when model run process is already studied and user is interested in results of simulation only). This method is applied in systems 2.1 and 2.2, but since requirements of users could be different, both systems are supported by method b. too.

b. Simultaneously with the computer experiment running information about model running is displayed. This can help to evaluate model during the experiment running and break it if need in correction would arise. As well this method is useful when behaviour of simulated system is reflected not so much in results as in the model running itself. This method (as the most universal) is applied in all three systems. In systems 2.1 and 2.2 choice between methods a. and b. is afforded depending on researcher interests: watching model state changing or getting result statistics only. In the simulation system of credit-banking operations the purpose of study is known, so here method b. is more developed: output information is represented in the graphic primitives. A snapshot of this system operation is represented In Fig.5.

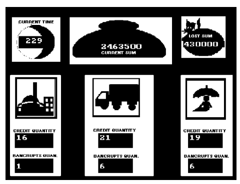

Fig.5. System 2.3 snapshot.

On the background an "moon" icon time of modeling (CURRENT TIME) is changed. A current value of sum (CURRENT SUM) is showed at the icon "purse". In the right upper corner of screen in "Basilio-cat" icon data (LOST SUM) appear only if a bankruptcy of some enterprise occurs — financial losses are fixed in this window. The general performance of the process of lending is displayed in "credit fields". Each field is supplied by the picture marking credit type: for stable, average and venture enterprises. These fields are placed at the lower part of the screen from left to right. In two corresponding windows of each "field" the following information is output: how many credits of given type were issued and how many of them have gone bankrupt.

3.3 Simulation results review/analysis

Results of simulation modeling usually represent by themselves a statistics like that really collected during natural experiment. Along the collection the major part of the results is primary processed, so statistics is represented in certain "probabilistic" form already. There are utilization coefficients, averages, hits on tables, etc. Let us propose that user of simulation model either is a professional only in his own area, or requires from models well foreseeable results in terms of application domain. Really friendly simulation system thereby shall contain the means of translation of general statistics in notions of application domain. For instance, it is better immediately attach the analysis tools applied only to characteristics of certain type to the corresponding terms of application domain. This is possible to do either by offering user the choice only of possible characteristics (that expects one more filter) or by beforehand addition to notions possible analysis means as object properties. The first variant is applied in systems 2.1, 2.2 and partly in 2.3. In 2.3 the corresponding characteristics are automatically attached to some analysis means. For example on Fig.6. there is the snapshot where the Kiviatty diagram for experiment series is attached to the two simulated system characteristics: final sum as a positive criterion and lost sum as a negative one.

4 Improvements and additions

Consider now some possible complications of standard architecture used in systems developed by our group.

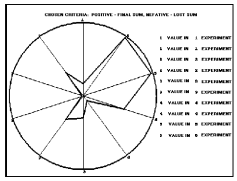

Fig.6. System 2.3 snapshot: Kiviatty Diagram

First this concerns changes of model. Not all possible changes, required by researcher for the benchmark analysis, are possible to represent by changing the parameters. In particular for our system researching the process of compilation technology of paralleling can be changed. Since it is impossible to simulate different technologies by one model, in this case we add to the interface items of model choice and comparative analysis of results obtained with different models. It is also possible to associate to a single simulation system several absolutely different models, if they refer to the same application domain. The described above simulation system of different transport can serve as an example of the such system.

Secondly, it is possible to associate the components of architecture "as a whole", that is implemented in the system 2.3.

Return now to the user, which himself develops a model. For this he usually uses either some universal programming language, or general-purpose simulation system. However, if long-time unique model processing goes, it is also more preferable to use a problem-oriented system. In this case the problem-oriented system inherits from its universal parent means of edit and translation of models.

5 Conclusions

Extracting of common components from the architectures of different problem-oriented simulation systems allows us to propose the basic architecture for any system of such type. On the basis of this architecture the author develops technique for construction of problem-oriented simulation system. Now using the developed technique we design the software tools that automatically creates problem-oriented systems — SimW (Simulation Wizard).

References

[1] Simulation systems integration and problem orientation. Science-technical report of "Simulation languages group", Institute of Mathematics, Academy of Sciences, Moldova Republic, 1995.

[2] Pritsker A. Alan B., Introduction to simulation and SLAM II, John Willey&Sons NY, 1984.

[3] Nadvodniuc N.A., Burtseva L.V. and Kukotina N.Yu., A simulation system for analysis of compiling process, In: Proceedings European simulation meeting on simulation tools and applications, Gyor, Hungary, 1995, pp.150–156.

L.Burtseva,
Institute of Mathematics,
Academy of Science of Moldova,
2028, Kishinev, Moldova.
e–mail: 24*luda@math.moldova.su*

Received October 7, 1997