Computer Science Journal of Moldova, vol.**4**, no.2(11), 1996

Simulation systems: design and applications

L.Burtseva G.Magariu

Abstract

In this paper the history of Simulation System Group investigations is presented. Some important achievements in past and present time are marked. The directions of future investigations are discussed in the fourth section of the paper.

1 Introduction

Modelling — physical and later abstract has been and still remains one of the principal methods of knowledge acquisition about the system we need to investigate. Simulation is a kind of modelling, where a model represents an abstract description of probable system states and processes of transition from one state to an other. Taking into account modern state of computer science it is easy to translate such description into computer program, so just this kind of modelling often is the only appropriate tool for complex systems investigation. It is also applicable when there does not exist an analytic expression of interdependence between input and output parameters of the system being modelled or an analytic model is so complicated, that does not allow to calculate the system characteristics which are interesting for the user.

2 How it has begun

In 70-ties there was the Programming Automatization Laboratory in the Institute of Mathematics of the Academy of Sciences of Moldova. It dealt with the most popular and fruitful direction of computer science at that time, namely, compiler construction (CC) [16] [1]. The original methodology of CC was developed [11] [12] on the base of the formalism of the Context-Controlled Grammar (CCG) [13]. The last one was created under influence of works of Ukrainian scientists [35]. These connections of Moldavian and Ukrainian scientists continue till today. The CCG serves as an instrument of description the syntax of programming languages. On the base of the CCG the universal algorithm for syntactical analysis was elaborated [14]. It is bottom-up grammar analysis without search and deadlocks. Search and deadlocks are avoided owing to correspondingly constructed contexts: any limited number of symbols from the left and only one symbol from the right. The correctly chosen contexts also allow:

- to distinguish the same syntactical units used in different syntactical structures and then to generate the correspondent object code by different semantical modules;
- to find out incorrect symbol just as it is put into syntactical stack.

As the result of syntactical analysis the linear record of syntax parse tree is produced. Then it is used for semantic analysis and code generation control. So the CC methodology based on CCG is not only syntactical oriented, it is syntactical directed one. This methodology is used by SSG for simulation language compilers design in the frame of general purpose simulation systems till now [6]. At first simulation language compiler was elaborated in the Programming Automatization Laboratory as a part of a multilanguage compilation system. The system involved two languages: Algol-like language Algams [2] and simulation oriented language SOL [20]. These languages were chosen as the two ones which differ one from an other quite enough to appreciate the volume of the source language independent part of the system and the volume of the source language oriented part. Besides these researcher's goals the system investor (Moscow architecture-planning managing department) used both system languages for its problems solving. The results obtained during design of this system were published in 2 monographs [11] [12] and in a number of scientific papers which were awarded by diploma of Moldavian Exhibition of achievements in the national economy. The group of authors these results received was the first one in Moldova to be honored by youth Moldova reward.

At the same time the main principles of the discrete simulation language were elaborated [19]. Later they and the language itself were developed [21] [22] and used for simulation language compilers construction.

So the second simulation language compiler was elaborated for computer IBM370 [23]. During its realization the following possibilities were added to SOL language [37]:

- using main statistical characteristics of facilities, stores, tables, processes and labels as initial arithmetic expressions;
- output of current model state at any moment of model time by special statement (the information about each facility, store and transaction is provided);
- output of standard statistical information about model development not only at the end of simulation but at any moment of model time by special statement and control of completeness of this information and of its details;
- output (by special statement) of transaction queue to any facility at any model time moment and of corresponding facility statistical characteristics (the maximal length of queue to the facility during the period of simulation);
- a series of simulation experiments execution with different values of model parameters by special statement (the parameter values are taken from special data file).

This project was supported by the Institute of Applied Geophysics of Government Committee of Hidrometeorology of the USSR. The compiler was used for investigation of information flows in the Informational-computational gelio-geophysical system by elaboration of simulation model of this system and execution of more than 100 experiments with it [7].

Such large number of experiments and processing of their results helped to realize that part of this processing can be performed automatically. So the simplest graphical experiment results representation was implemented in the frame of the same project [3] and more interesting and complicated methods of the experiments results analysis were developed in the later projects.

At the same time the methods of compiler construction for simulation languages were investigated which provide paralleling of simulation experiment. This investigations were partly supported by the Government Committee of Science and Technics of the USSR. Three parallel strategies were investigated. Two of them were based on synchronization of parallel transactions by model time and preserved natural source language parallelism [15]. The third strategy consists in finding out (on the statistical analysis stage) the critical points of possible conflicts while sharing global objects by transactions [24]. This strategy allows to parallelize transactions execution dynamically considering synchronization by model time only when transaction runs through a critical point.

So the simulation researches turned to be interesting and perspective. And soon (1987) The Simulation Systems Group has separated from the Programming Automatization Laboratory.

3 Present investigations

Because of personal computers advancement, at that time the next project was PC/SOL system [25]. The system joins in a single interface the possibilities of model compilation, experiments execution and experiments results analysis.

By the way for the system marketing and testing the simulation models of real systems of different nature were elaborated. They are:

- simple tutorial and demonstration models as:
    - amoeba population model [28],

  - transport events models (autotransportation, port work, quarry work) [36][1]
  - model of TV plant [26][2],
  - muliservice system model[3],
  - two computer architecture models (cluster and multibus type) [5][4],

- real applications:
  - carried out assessment and choice of working regimes of informational-calculational geliogeophysical system [8];
  - investigated the system of data processing of ionosphere radiosounding by satellite [9];
  - simulation of information support of local data bases in a network of satellites of geophysical control system.

Meanwhile it is obvious that simulation as a tool for research of complex system does not now occupy the place it must do. To correct this it is necessary not only to have suitable software, but to train correspondingly the students.

Simulation system SimTeach [6] [4] [30] is designed for discrete simulation of complex systems of different nature and allows to easy study the simulation method. Convenient interface integrates means for models description, editing and analysis, simulation run visualization, results analysis and models base support. So, system SimTeach supports all stages of the work with simulation models. SimTeach source simulation language constructions are provided by context help in built-in editor. Means for the models base support will allow the professor to easy add to the base the necessary models and instructions for the students and successfully use the system for the practices on the simulation method application in any field.

[1]problem formulation with some modifications was taken from [33]
[2]problem formulation with any modifications was taken from [33]
[3]problem formulation with some modifications was taken from [33]
[4]problem formulation with some modifications was taken from [32]

So during design and evaluation testing SOL/PC and SimTeach there were created the methods of design and integration of general purpose simulation systems. The main aspects of these methods are:

- system architecture;
- representation of experiments results;
- a number of analysis means:
  - graphical representation of statistical characteristics depending on chosen parameter;
  - creation of tables with selective results of series of experiments;
  - creation the Hunt diagrams for tracing facility state during the simulation running;
  - creation of Kiviat diagrams for multicrriteria simulated system evaluation;
- the means of simulation experiment running visualization [29]:
  - reflection of states of model objects at the current moment of model time;
  - unwrapping of changes of model objects states according to the model time.

It is obvious that SOL/PC and SimTeach systems are general purpose ones for the work with simulation models of real systems of different nature. These systems imply that the user has some initial knowledge about the simulation technique and the basics of the language used for this purpose. However there exist a lot of potential users who can not allow themselves or simply do not want to spend their time for special simulation and language training. For such persons it is rational to develop problem-oriented simulation systems. These systems include one or a set of models of the system(s) interesting in. The models are "black box(es)" for the user. These models are written according to users requirements by developers who are quite skilled. Problem oriented systems integrate the possibilities of:

- setting of values of these systems models parameters which usually simulate different load of the system and so on;
- execution of simulation experiments with these systems models with values of parameters set by the user;
- statistical and graphical means of analysis of simulation experiments;
- simulation experiments running visualization.

Unlike general purpose systems problem-oriented ones design dialog with the user in the problem domain terms.

During two last years three problem-oriented systems were created by SSG.

1) Credit bank process simulation. By making a number of simplifications, the credit operation process was defined as parallel carrying out of the following functions: credits granting and their flow (we mean percentage receiving, credit sum return or bankrupts registration). Three types of credits were chosen.

For stable enterprises — they demand great sums for a long time period (in our case up to the end of the financial year) under a small percentage; they go to bankruptcy rarely.

For medium stable enterprises — they demand less sums, higher percentage, considerably shorter time period (in our case three months): they go to bankruptcy frequently.

For ventures — they demand small sums, high percentage, credit time is minimal (in our case one month), but the probability of bankruptcy is very high.

The bank grants credits from the current sum which is at its disposal and according to clients priorities. Naturally, the current sum must be greater than the credit requested, besides we take into consideration "personal characteristics" of the system users. Priorities are probabilities of the fact that the credit will be granted. A user preferring risky strategies will likely grant credits to ventures at first (i.e. give higher probability to them). On the contrary, the user, preferring stability will choose stable enterprises. At "control time" (in our case at the month at the end of the financial year) the bank stops credits granting and only collects granted sums and percentages. We developed on the base of this model the computer business game “Credit-bank operations”.

2) The simulation system for analysis of compilation process [32] may be used in our opinion both by compiler designers and compiler users. For designers system allows to check experimentally some theoretical assumptions. Questions of compilers reuse and their paralleling may be also cleared up. It might help to estimate the time and effort that the user of compiler spends for program debugging. User can find out on one hand what kind of compiler been proposed he needs for his equipment; on the other hand what equipment must he buy for the given compiler.

We choose two approaches for compiling strategies comparison. The first chart corresponds to the traditional sequential compilation process when its work is divided into several parts: lexical analysis, syntactic control, semantic analysis and code generation. On successful completion of one of the phases the next part starts working. If any errors have been detected the compilation process is stopped. Taking into account, that while one of the compilation process parts is working, information collected can be sufficient for the work of the next part (that decreases the time for the whole compilation process), we designed another chart, according to that the compiler works in such a way that the parts communicate with each other and partly act concurrently. If any of the parts detects errors the compiler stops its work.

The system can be used both for theoretical and practical problems solution. It allows to investigate the behavior of a compiler built in a classical way under changing conditions of utilization (the case can take place before compiler creation), the possibilities of ready-for-use product reuse, its quality checking and to make optimization of resources that are at designers (or investigators) disposal by classical sequential compilation strategy modifying. For work with our compiler simulation system the interface is designed that is similar to widely used Turbo-interfaces. In the system the work of compilers that differ by construction method but use the same resources is simulated. Two simulation models are organized by “black box” principle. The first model assumes sequential implementation of different compiling stages, and the second — their partly concurrent (on necessary information collecting) running. Information in the form of data that can be changed (called parameters) is put at the "black box" input. We suggest three ways to change a compiling strategy (by parameters changing):

a) to change a set of technical tools (in our case it is a number of working processors) and as a result to sharer or vise versa to parallel resources needed for different stages of compiling process;

b) to change a class of programs that the compiler translates i.e. to change a probable appearance of different language constructions (such as identifiers, constants, key words and so on) in the user program and the text length;

c) to change a user "class" i.e. a probability of error appearance at different debugging stages.

A user inputs parameters in the form that interface provides. After running the experiments the results can be saved as a file. The information in it represents standard statistics. Moreover the system allows to analyze results by different (generally graphical) methods.

3) Simulation system of planning of different transport types work joins together three models: the quarry work model, the autotransportation system model, the port work model. These problems are joint by the following features: influence of random factors on transportation time, existing of sharing resources, statistical evaluation of queues. But there are some differences between the models. To represent these differences some simulation paradigms are used. For PORT model this is the storms influence: storm prevents the tanker arrival in the port and delays the empty tankers in the port. So this factor influences essentially the port work. For the AUTOTRANSPORTATION model its unique aspect is the "addwork". After working day has been finished the work must be continued, if all ground prepared by bulldozer was not transported. For QUARRY model unique aspect is the priority of big lorries in queue to pounder.

4 How we see the future

So, SSG has the practical experience and provides some theoretical aspects of design of both general purpose simulation systems and problem-oriented ones.

On the base of these experience how we see the future research directions?

First, the general purpose simulation systems require from users some professional knowledge in the domain of computer science. The problem-oriented simulation system is oriented to the professionals in some determinate domain. In conditions when computers enter more and more closely in our life, the number of user of second type would prevail. Because of this it will be usual to create second type systems on the base of advanced first type ones. So we propose the tools for such creation.

Second direction follows from the introduction of artificial intelligence techniques in all branches of computer science. Intellectualization of simulation systems supposes "teaching" them to appreciate simulation experiments results, to choose the "best" variants among these results, to support decisions, to plan further experiments in condition of lack of information. It is necessary to create new formalisms or adopt existing ones for application of artificial intelligence techniques to simulation systems.

5 Conclusions

The investigation, being provided by SSG, concerns one of the most perspective directions of computer science.

The questionings have appeared in several serveys [34] [39] [10] [17] [18] [31] cleared up that the simulation technique takes the second place among computer methods.

The analysis of **simulation** development history shows that this development has moved from practical problems resolving necessity through particular problem resolving software to the design of special simulation languages, simulation systems on the base of these languages and further to increase intellectual level of these systems.

During this period mutual influence of researchers and practitioners have being taken place with mutual benefit receiving.

These tendencies common for simulation development are traced in the history of investigations fulfilled by SSG. Undoubtedly, participation in automated informational-calculational geliogeophysical system investigation by simulation advanced the SSG work in designing the tools of execution of series of experiments and analysis of their results.

The achievements in system programming using language integrated environments were the basis for design by SSG of general purpose integrated simulation environments such as SOL/PC and SimTeach.

During last 5 years SSG researchers have being taken successful participation in a number of international scientific conferences: the Ukraine 1993 [27], the USA, China, Hungary, Russia, Austria 1995 [4] and so on). This fact confirms the quite qualitative level of investigations.

References

[1] A.Aho, J.Ulman. The Theory of Parsing, Translation and Compiling. Prentice-Hall, Engewood Cliffs, N.J., 1972.

[2] Algorithms and algorithmic languages, N3, The Calculation Center of the USSR Academy of Sciences, 1967.(Russian)

[3] L.Burtseva. Graphical representation of objects statistical characteristics changes based on the results of series of experiments with model in SOL/ES system, Mat. Issled., No.81 (1985), pp.3–7. (Russian)

[4] L.V.Burtseva, N.Yu.Kukotina, G.A.Magariu, N.A.Nadvodniuc, T.L.Tofan. Application of Simulation System SOL/PC for Education. Poster Book (ARGESIM Report) of EUROSIM Simulation Congress, Vienna, Austria, 1995.

[5] Computer system architecture. Parallel programming. Investigations of parallel system and computing by simulation. Scientific-technical report, The Institute of mathematics, Academy of Sciences of Moldova, Kishinev, 1992. (Romanian)

[6] Development of software tools for simulation method teaching in higher educational institutions, Scientific-technical report, The Institute of mathematics, Academy of Sciences of Moldova, Kishinev, 1995.

[7] F.L.Dlikman, A.Z.Zelikovskii, G.A.Magariu, T.L.Tofan, F.M.Frishberg. Interactive system of graphical representation of geliogeophysical information investigation by means of translation system SOL/ES, Mat. Issled., No.81 (1985), pp.87–90. (Russian)

[8] F.L.Dlikman, F.M.Frishberg, V.P.Sedyakin, G.A.Magariu, T.L.Tofan, L.V.Burtseva. Evaluation of characteristics and choice of working regimes of geliogeophysical informational-calculational system according to simulation results, In: Advances in Modeling &Analysis, B, AMSE Press, Vol.31, 4 (1994), pp.23–29.

[9] F.L.Dlikman, F.M. Frishberg, M.Yu.Borozdina, G.A.Magariu, T.L.Tofan, L.V.Burtseva. Investigation of functioning of system of processing of data of ionosphere radiosounding by satellite, Summaries of International Conference on " Systems, Control, Information" methodologies & applications, China, 1994.

[10] G.F.Forgionne. Corporate Management Science Activities: An Update, Interfaces 13, 20–3.

[11] M.G.Gontsa, M.N.Marichiuc, G.A.Bostan, G.A.Magariu. Parametric translation systems, Part I: Basic principles and design methods. Shtiintsa, Kishinev, 1974. (Russian)

[12] M.G.Gontsa, M.N.Marichiuc, G.A.Bostan, G.A.Magariu. Parametric translation systems, Part II: System START source languages and guide. Shtiintsa, Kishinev, 1974. (Russian)

[13] M.G.Gontsa. Some problems of parametric programming systems algorithms construction and optimization, Summary of PhD Thesis, Institute of Cybernetics AS UR Press, Kiev, 1969. (Russian)

[14] M.G.Gontsa, M.N.Marichiuc. Parametrically controlled syntactical control and analysis of programming languages, Proceedings of the second USSR Conference on Programming , series K, Novosibirsk, (1970), pp.65–72. (Russian)

[15] G.F.Grecu, G.A.Magariu, M.N.Marichiuc. Parallel processes synchronization in SOL language implementation. Abstracts of YII USSR school-workshop on Parallel programming and high productivity systems, Institute of Cybernetics AS UR, Kiev, (1986), pp.96–97. (Russian)

[16] D.Gries. Compiler Construction for Digital Computers, Wiley, New York, 1971.

[17] J.L.Harpell, M.S.Lane, A.H.Mansour. Operation Research in Practice: A Logitudinal Study, Interfaces 19, pp.65–74.

[18] J.L.Harpell, M.S.Lane, A.H.Mansour. Operation Research Techniques: A Logitudinal Update, Interfaces 23, pp.63–68.

[19] G.A.Juravleva-Magariu. Simulation language SOL and some principles of its implementation. Mat. Issled., Vol.8, Shtiintsa, Kishinev, (1973), pp.62–72. (Russian)

[20] D.E.Knuth, J.L.McNelley. The formal definition of SOL//IEEE Transaction. Vol.EC-13 No.4, (1964), pp.409–416.

[21] G.A.Magariu. SOL language extension by means of model analysis. Mat. Issled., Shtiintsa, Kishinev, Vol.81, (1985), pp.87–90. (Russian)

[22] G.A.Magariu, M.N.Marichiuc. Transactions parallel development simulation in translation system SOL/ES. Mat. Issled., Shtiintsa, Kishinev, (1987), pp.90–93. (Russian)

[23] G.A.Magariu, M.N.Marichiuc, T.L.Tofan. Translation system SOL/EC oriented to simulation of complex systems. Moldavian Academy of sciences News, Series of physical-technical and mathematical sciences, Shtiintsa, Kishinev, No.3, (1986), pp.54–55. (Russian)

[24] G.A.Magariu, M.N.Marichiuc, G.P.Moshneaga. Parallelism in implementation of the discrete simulation language SOL. Control systems and machineries, (1988), No.5, pp.20–22. (Russian)

[25] G.Magariu, V.Madan, L.Burtseva. What does the simulation system SOL/PC do? Computer science journal of Moldova, Vol.1, No.2, (1993), pp.51–62.

[26] G.Magariu, L.Burtseva. Graphical representation of experiments results in simulation system SOL/PC. 9th International conference on Control systems and Computer Science, CSCS 9. Conference Preprints. Bucharest, 25–28 May 1993.

[27] G.Magariu, L.Burtseva. Analysis of experiment results in simulation system SOL/PC. Proceedings International AMSE Conference on Applied Modelling and Simulation, Lviv (the Ukraine), (1993), pp.51–62.

[28] G.Magariu, L.Burtseva. Biological processes modelling by simulation system SOL/PC. 9th International conference on Control systems and Computer Science, CSCS 9. Conference Preprints. Bucharest, 25–28 May 1993, pp.280–289.

[29] G.A.Magariu. Model running visualization in simulation system SOL/PC, Proceedings European simulation meeting on simulation tools and applications, Gyor, Hungary, (1995), pp.217–223.

[30] G.Magariu, L.Burtseva, N.A.Nadvodniuc. Software for simulation teaching support, The 5-th St.Peterburg International Conference “Regional Informatics-96”, Abstracts of reports, St.Peterburg, May 13–16, Part 2, (1996), pp.252–253.

[31] C.L.Morgan. A Survey of MS/OR Surveys, Interfaces 19, pp.95–103.

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

[33] Alan B.Pritsker. Introduction to simulation and SLAM II, John Willey & Sons, New York–Chichester–Brisbane, 1984.

[34] J.J.Rasmussen and T.George. After 25 years: a survey of Operation Research Alumni, Case Western Reserve University, Interfaces 8, pp.48–52.

[35] V.N.Redko, E.L.Yushchenco. Algorithmic languages and translating systems, Cybernetics, (1967), No.5, pp.87–91. (Russian)

[36] Simulation Systems Integration and Problem Orientation, Scientific- technical report, The Institute of mathematics, Academy of Sciences of Moldova, Kishinev, 1995. (Russian)

[37] Software development for simulation models of automated informational-calculational geliogeophysical subsystems investigation. Scientific-technical report, The Institute of mathematics, Academy of Sciences of Moldova, Kishinev, 1990. (Russian)

[38] Susumu Horiguchi, Tahashi Suzuki and Yoshiharu Shigei. Throughput of multiprocessor system with Common Memories, The transaction of the Institute of electronic and Communication Engineers of Japan, Section E (English), Vol.E69, No.6 June (1986), pp.726–729.

[39] G.Thomas and J.DaCosta. A sample survey of corporate operation research, Interfaces 9, pp.102–111.

Liudmila Burtseva is a junior scientific researcher in the Institute of Mathematics of Academy of Sciences of Moldova Republic. She is a Kishinev State University graduate. Her scientific interests include simulation, simulation experiments results analysis, graphical analysis means, general purpose and problem-oriented simulation systems architecture. She has published 7 papers.

Galina Magariu is a leader scientific researcher in the Institute of Mathematics of Academy of Sciences of Moldova Republic. She is a Kishinev State University graduate. In 1980 she received her Candidate's Degree (Ph.D) in physico-mathematical sciences from the the Ukrainian Academy of Sciences Institute of Cybernetics. Her scientific interests include simulation systems design and applications. She has published 3 monographs and over 40 papers.

phone: (373–2) 738110; fax: (373–2) 738027
e-mail: 24gal@math.moldova.su