THREE SIMULATION EXPERIMENTATION ENVIRONMENTS: SIMAD, SIMGEST, AND E/SLAM

Tuncer I. Ören
McLeod Institute of Simulation Sciences
Computer Science Department, University of Ottawa
Ottawa, Ontario K1N 6N5 Canada
oren@csi.uottawa.ca

ABSTRACT

Three simulation experimentation environments, each implemented on a Sun workstation, are reported. Each one of them illustrates a different aspect of the computer-aided support necessary in simulation experimentations. SimAd (**Sim**ulation experimentation **Ad**visor) is an environment for statistical design of simulation experiments. It currently supports Modsim II and Voltaire languages and factorial design and fractional factorial design techniques. Its architecture allows multilingual interface with users; a feature which can easily be implemented in any interactive software in general and in interactive simulation environments, in particular. SimGest (**Sim**ulation experimentation environment for **Gest**) allows automatic generation of simulation programs from high-level model specifications as well as computer-aided specification of experimental conditions. E/Slam (Elucidation of **Slam** II programs) is a knowledge-based simulation experimentation environment for Slam II programs. It facilitates understanding of Slam II programs as well as knowledge-based reliable editing of new experimental conditions. E/Slam has a built-in Prolog-based component to assure consistency in the specification of the experiments.

INTRODUCTION

Simulation experimentation environments provide facilities to specify, design, and monitor experiments and to perform analyses of the generated model behavior. They can be invoked from a comprehensive simulation environment. However, their functionalities are distinct from modelling environments. Some of the existing tools and environments for simulation experimentations are developed by Birta and So (1990); Blaisdell and Haddock (1992); Frederking (1991); Hu and Rozenblit (1988); Jávor (1986); O'Keefe (1986); Spiegel and LaVallée (1988); and Taylor and Hurrion (1988).

In this article, three simulation experimentation environments, namely SimAd, SimGest, and E/Slam are presented. They are conceived and implemented at the University of Ottawa as part of the ongoing research to advance the state-of-the-art of the simulation methodology and knowledge-based simulation environments. Each one of them illustrates a different aspect of the computer-aided support necessary in simulation experimentations. Some of their features can be useful in other advanced simulation experimentation environments.

SIMAD

SimAd (**Sim**ulation experimentation **Ad**visor) is an environment for statistical design of simulation experiments (do Rego 1992). A source simulation program has to be manually augmented to include a performance index (to allow comparison of the outcomes of different simulation runs). As seen in Figure 1, SimAd system has access both to an augmented source simulation program as well as its compiled version. SimAd consists of six functional modules:

1. An analysis module identifies the factors (decision variables) of the model. Currently, SimAd supports two simulation languages: Voltaire, a Petri net based language (Parent 1990) and Modsim II (Belanger and Mullarney 1990).

2. An interactive module displays the factors to a user to get their lower and upper levels.

3. A module generates the design matrix according to a specified technique. Currently, two statistical design techniques, i.e., factorial design and fractional factorial design are implemented (Law and Kelton 1991).

4. A module monitors the execution of the simulation runs. It provides each row of the design matrix to the compiled version of the augmented simulation program and takes back the corresponding value of the performance measure (response) to the design matrix.

5. A module analyses the design matrix to generate the main and interaction effects.

6. A module —which is not yet implemented— can have access to the factors, the design matrix, and the effects and interactions to provide advice to the user.

Due to its modular architecture, SimAd can be extended in four independent ways, to support the following functions:

1. Ability to process simulation programs written in languages other than Voltaire and Modsim.

2. Addition of statistical experimental design techniques other than factorial design and fractional factorial design.

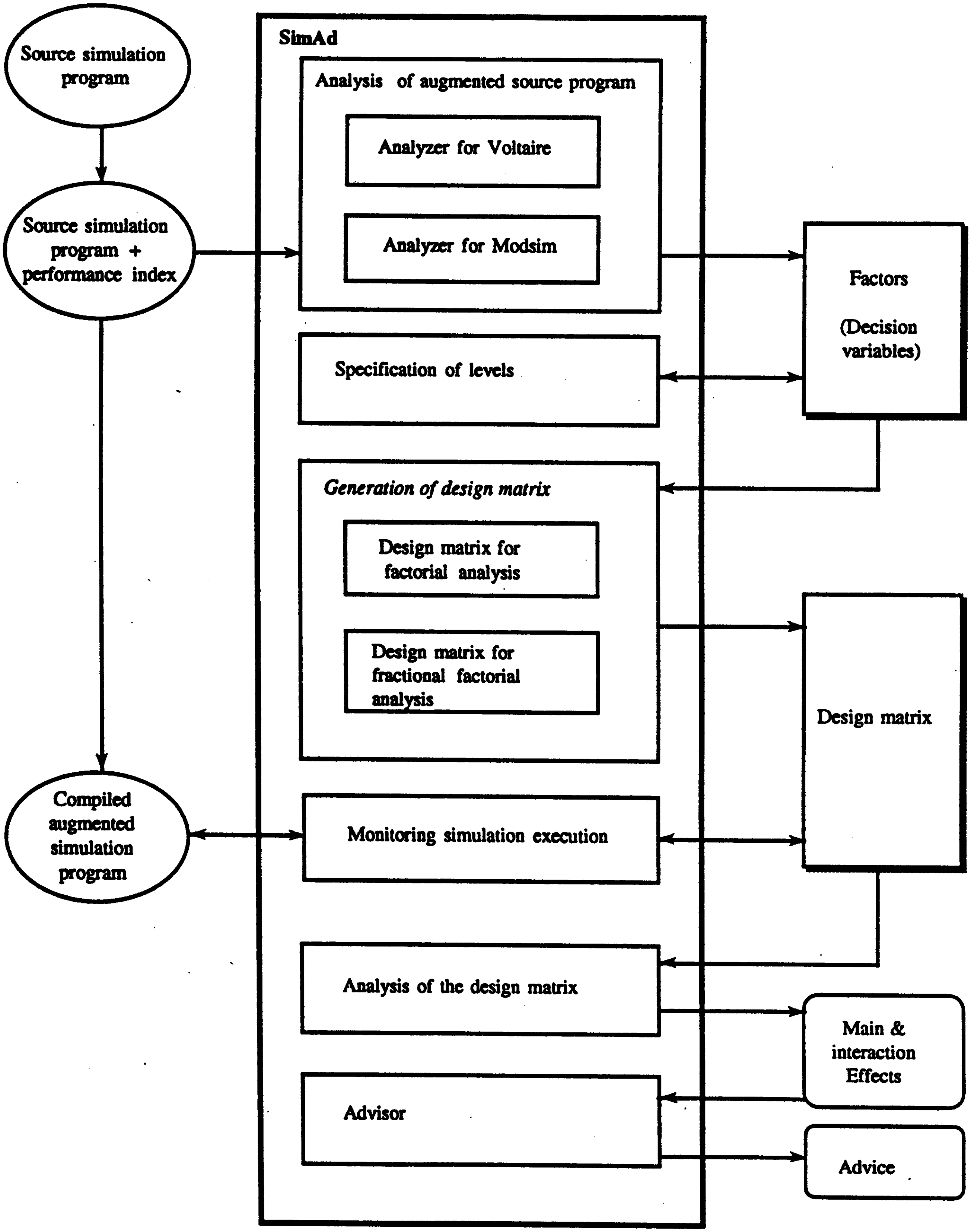

Figure 1. Functional Decomposition of SimAd (Simulation Advisor)

Due to its modular architecture, SimAd can be extended in four independent ways, to support the following functions:

1. Ability to process simulation programs written in languages other than Voltaire and Modsim.

2. Addition of statistical experimental design techniques other than factorial design and fractional factorial design.

3. Addition of a powerful advisor.

4. User / System interface in natural languages other than the currently available English and German. Due to the conception of its implementation, this ability can be added even by a user without recompiling the SimAd software. It is expected that this type of ability will be a common place for other modelling and simulation environments as well as other interactive software systems.

SIMGEST

Gest (General system theory implementor) is the name of a modelling and simulation language (Ören 1984) and an environment to support it. The Gest environment consists of MaGest and SimGest. MaGest (Modelling advisor for Gest) which has already been reported in the literature allows specification of models described by ordinary differential equations (Aytaç and Ören 1986). A new version which is being implemented allows graphical specifications of the coupling of the component models.

SimGest (Simulation experimentation environment for Gest) relies on two types of knowledge: the methodological knowledge of the formalism used in the Gest language to specify models and simulation experiments as well as the knowledge gleaned from a Gest model. As seen in Figure 2, SimGest consists of the following functional modules (Jin 1993):

1. An analyzer accepts a mathematical model expressed in Gest to generate a simulation program expressed in C as well as the templates of the data objects to be read before each simulation run.

2. A module guides the user in providing the knowledge necessary to specify sets of parameter values and the experimental conditions.

3. Another module is used for displaying the results.

SimGest is an example of a table-driven simulation environment. An important implication of SimGest is the facilities it provides to work with high level model specifications. Therefore, it supports the concept of model bases and allows a user to perform any update of a model on its high level specification rather than on a computer code.

E/SLAM

E/Slam (Elucidation of **Slam** II programs) is a knowledge-based simulation experimentation environment for Slam II programs on a Sun Workstation (Ören et al. 1992, Wendt 1993). It accepts as input a Slam II network program (Pritsker 1986). The following functions are supported by E/Slam (Figure 3):

1. In the analysis phase, an analyzer gleans knowledge from the source Slam II program. An internal model builder stores the knowledge in an internal model which is the essential part of this program understanding system.

2. In the synthesis phase, two types of documentation templates, i.e., statement and program templates can be generated. Contrary to the templates used in the SLAMSYSTEM (O'Reilly and Ryan 1992) E/Slam's templates have a vertical format which allows knowledge about several statements of the same type to be displayed in the same statement template (Ören et al. 1992). All 43 statement templates are implemented. Program templates provide vertical slicing of a program according to an interest area. Several program templates are designed and some are implemented. A program generator synthesizes an updated Slam program based on the source Slam program and user's specifications of new simulation run(s). A network graph generator is also included to display the model's graph representation. Templates can also be generated by selecting graph elements.

3. A knowledge-based editing module, named QA/Slam, allows a user to edit the source Slam program through the documentation templates (as opposed to a source Slam program), to specify new simulation runs. Some acceptability checks are also performed by this module. Hence with the help of QA/Slam, the editor used to specify new experiments become a knowledge-based reliable editor.

4. A knowledge-based quality assurance module checks the acceptability and the consistency of the user's specifications against a rule base expressed in Prolog. It either certifies the acceptability of the specifications or advises the user on the sources of the inconsistencies and recommends possible remedies.

CONCLUSIONS

Computer aided environments for simulation studies are maturing. Already a good number of modelling and simulation environments exists. In this article, three simulation experimentation environments which provide different categories of computerized assistance for simulation experimentation are reported.

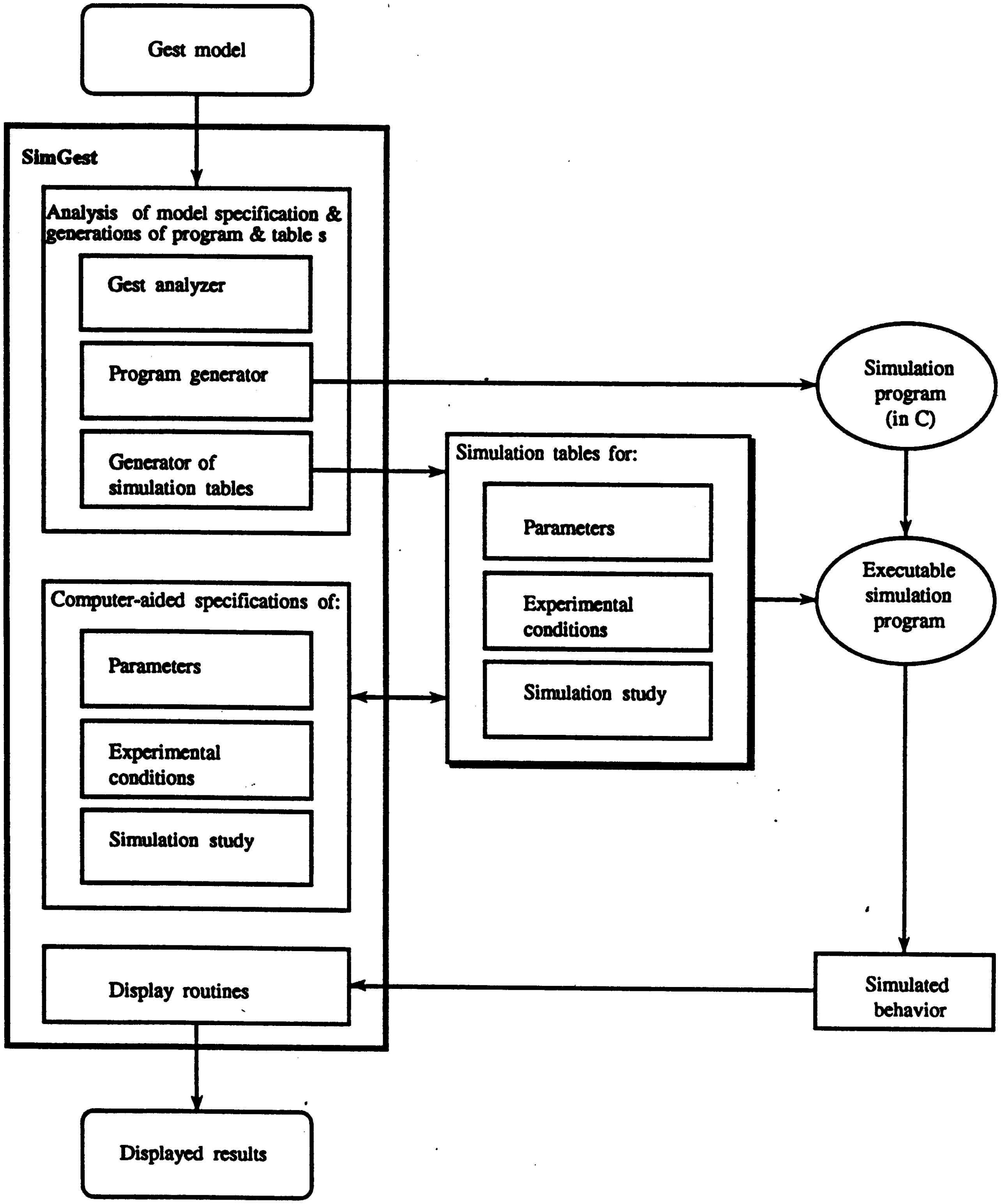

Figure 2. Functional Decomposition of SimGest
(Simulation Experimentation Environment for Gest)

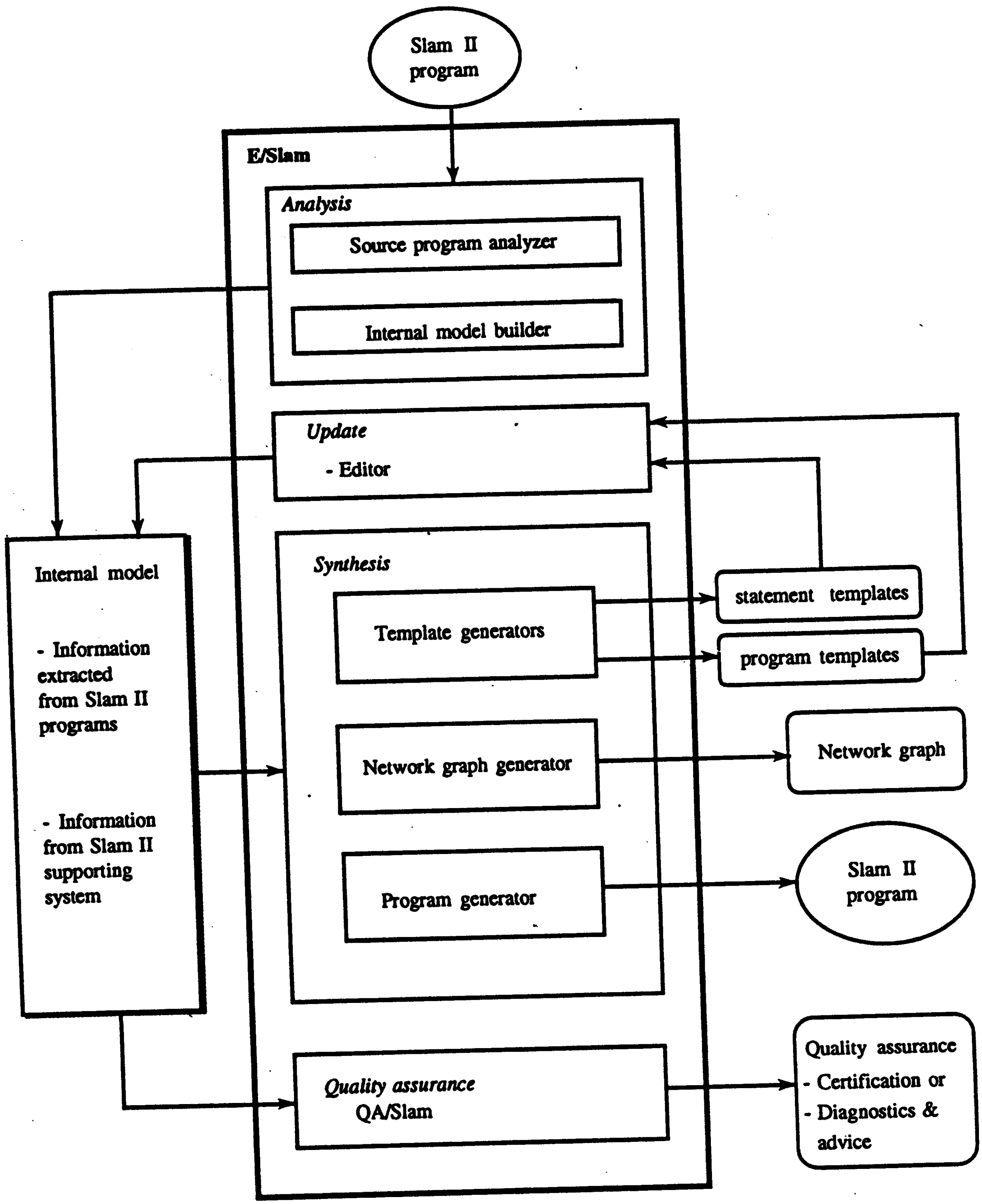

Figure 3. Functional Decomposition of E/Slam (Elucidation of Slam Programs)

ACKNOWLEDGEMENT

SimAd and E/Slam have been carried out as part of a broad project funded by Bell Canada. I wish to express my appreciation both for this financial support and the encouragement and cooperation of Mr. F. Coallier and Mr. O. Tanır of Bell Canada. The support of the Austrian Research Council (Fonds zur Förderung der wissenschaftlichen Forschung) under contract number J0521-PHY made the contributions of Dr. Martin Hitz to E/Slam and other parts of the research possible. As members of the research group, Dr. L.G. Birta and Dr. D. King did also significantly contribute to the research. Mr. do Rego wrote his thesis on SimAd to support the Voltaire language. Ms. K. Stein, from the University of Nuremberg, Germany, during her six months stay at the University of Ottawa, modified SimAd to support Modsim and to add multilingual abilities. Mr. Q. Jin developed SimGest based on a previous implementation (Ye 1989). Mr. N.R. Wendt designed the templates, developed QA/Slam and integrated it to E/Slam.

REFERENCES

Aytaç K.Z. and T.I. Ören. 1986. "MaGest: A Model-Based Advisor and Certifier for Gest Programs." In: Modelling and Simulation Methodology in the Artificial Intelligence Era, M.S. Elzas, T.I. Ören, B.P. Zeigler eds., North-Holland, Amsterdam, 299-307.

Belanger, R. and A. Mullarney. 1991. "MODSIM II Tutorial." CACI Products Company, La Jolla, CA.

Birta, L.G. and M. So. 1990. "NEMS: A Database Support Environment for Numerical Experimentation." Simulation, 54:4 (April), 189-200.

Blaisdell, W.E. and J. Haddock. 1992. "Simstat: A Tool for Simulation Analysis." In: Proc. of the 1992 Winter Simulation Conference, J.J. Swain et al. eds, Dec. 13-16, 1992, Arlington, VA. SCSI, San Diego, CA, pp. 421-425.

do Rego, Y. 1992. "Outil logiciel pour le design et analyse d'expérience par les méthodes d'analyse factorielle et fractionnelle factorielle." Master's Thesis, Ottawa-Carleton Institute for Computer Science, University of Ottawa, Ottawa, On. Canada.

Frederking, R.E. 1991. "Interactive Experiment Planning to Control Knowledge-Based Simulation." In: Proc. of the 1991 Winter Simulation Conference, B.L. Nelson, W.D. Kelton, and G.M. Clark eds., Dec. 8-11, 1991, Phoenix, AZ. SCSI, San Diego, CA, pp. 1218-1227.

Hu, J., and J.W. Rozenblit. 1988. "Towards Automatic Generation of Experimental Frame in Simulation-Based System Design." In: AI Papers, 1988, Proc. of the Conference on AI and Simulation Simulation Series 20:1, 1-6, R.J. Uttamsingh ed., SCSI, San Diego, CA.

Jávor, A. 1986. "Applications of Expert Systems Concepts to Adaptive Experimentation with Models." In: Modelling and Simulation Methodology in the Artificial Intelligence Era, M.S. Elzas, T.I. Ören, B.P. Zeigler eds., North-Holland, Amsterdam, 153-163.

Jin, Q. 1993 - In Progress. "SimGest: A Simulation Experimentation Environment and a Program Generator." Masters Thesis, Systems Science Programme, University of Ottawa, Ottawa, Ont., Canada.

Law, A.M. and W.D. Kelton. 1991. Simulation Modeling and Analysis. McGraw-Hill, New York.

O'Keefe, R.M. 1986. "Advisory Systems in Simulation." In: AI Applied to Simulation, E.J.H. Kerckhoffs, G.C. Vansteenkiste, B.P. Zeigler, eds.. Simulation Series 18:1, 73-78. SCSI, San Diego, CA.

O'Reilly, J.J. and N.K. Ryan. 1992. "Introduction to Slam II ® and SLAMSYSTEM ®." In: Proc. of the 1992 Winter Simulation Conference, J.J. Swain et al. eds, Dec. 13-16, 1992, Arlington, VA. SCSI, San Diego, CA, pp. 352-356.

Ören, T.I. 1984. "GEST - A Modelling and Simulation Language Based on System Theoretic Concepts." In: Simulation and Model-Based Methodologies: An Integrative View, T.I. Ören, B.P. Zeigler, M.S. Elzas eds.. Springer-Verlag, Heidelberg, Germany, pp. 281-335.

Ören, T.I., Wendt, R.N., King, D.G., Abou-Rabia, O., Birta, L.G. 1992. "A Template-Oriented Approach for Simulation Program Understanding." Mathematics and Computers in Simulation, 34: 219-229.

Parent, P. 1990. "Voltaire Netlist Language." Tutorial Manual. VLSI Design Laboratory, Department of Electrical Engineering, McGill University, Montreal, Québec, Canada.

Pritsker, A.A.B. 1986. Introduction to Simulation and SLAM II. Halstead Press, New York, NY.

Spiegel, J.R. and D.B. LaVallée. 1988. "Using an Expert System to Drive a Simulation Experiment." In: AI Papers, 1988, Proc. of the Conference on AI and Simulation (Simulation Series 20:1, 108-112), R.J. Uttamsingh (ed.). SCSI, San Diego, CA.

Taylor, R.P. and R.D. Hurrion. 1988. "An Expert Advisor for Simulation Experimental Design and Analysis." In: Artificial Intelligence and Simulation: The Diversity of Applications, February 3-5, 1988, San Diego, CA, T. Henson (ed.). SCSI, San Diego, CA, 238-244.

Wendt, N.R. 1993. "Application of Program Understanding and Rule-Based Quality Assurance to Slam II Simulation Programs." Masters Thesis, Ottawa-Carleton Institute for Computer Science, University of Ottawa, Ottawa, Ont., Canada.

Wittman, J. 1992. "Model and Experiment — A New Approach to Definitions." In: Proc. of the 1992 Summer Computer Simulation Conference P. Luker, ed., July 27-30, 1992, Reno, Nevada, 115-119.

Ye, Y. 1989. "A M ethod for Automatic Program Generation from a Specification." In: Proc. of the Canadian Conference on Electrical and Computer Engineering, Montréal, Québec, Canada, Sept. 17-20, 1989. pp. 1078-1081.

EUROPEAN SIMULATION SYMPOSIUM 1993

OCTOBER 25-28, 1993
DELFT UNIVERSITY OF TECHNOLOGY
DELFT THE NETHERLANDS

EDITED BY
ALEXANDER VERBRAECK
EUGENE J.H.KERCKHOFFS

Organized by:
The Society for Computer Simulation International
Hosted by DELFT UNIVERSITY OF TECHNOLOGY
in cooperation with EuSC, CASS, JSST and EUROSIM

Printed in the USA

Ören, T.I. (1993). Three Simulation Experimentation Environments: SimAd, SimGest, and E/Slam. In: Proceedings of the European Simulation Symposium, A. Verbraeck and E.J. H. Kerkhoffs (eds.), Delft, The Netherlands, SCS, San Diego, CA, pp. 627-632.