SIMULATION MODELING AND SIMNET

Prentice-Hall International Editions

HAMDY A. TAHA

SIMULATION MODELING AND SIMNET

HAMDY A. TAHA

University of Arkansas/Fayetteville

PRENTICE-HALL INTERNATIONAL, INC.

0-13-811621-0

This edition may be sold only in those countries to which it is consigned by Prentice-Hall International. It is not to be re-exported and it is not for sale in the U.S.A., Mexico, or Canada.

To MAISA
for the joy of fatherhood

A Division of Simon & Schuster
Englewood Cliffs, New Jersey 07632

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-811621-0

PRENTICE-HALL INTERNATIONAL (UK) LIMITED, *London*
PRENTICE-HALL OF AUSTRALIA PTY. LIMITED, *Sydney*
PRENTICE-HALL CANADA, INC., *Toronto*
PRENTICE-HALL HISPANOAMERICANA, S.A., *Mexico*
PRENTICE-HALL OF INDIA PRIVATE LIMITED, *New Delhi*
PRENTICE-HALL OF JAPAN, INC., *Tokyo*
SIMON & SCHUSTER SOUTHEAST ASIA PTE. LTD., *Singapore*
EDITORA PRENTICE-HALL DO BRASIL, LTDA., *Rio de Janeiro*
PRENTICE-HALL, INC., *Englewood Cliffs, New Jersey*

SIMULATION MODELING AND SIMNET

PRENTICE HALL INTERNATIONAL SERIES
IN INDUSTRIAL AND SYSTEMS ENGINEERING

W. J. Fabrycky and J. H. Mize, Editors

ALEXANDER *The Practice and Management of Industrial Ergonomics*
AMOS AND SARCHET *Management for Engineers*
BANKS AND CARSON *Discrete-Event System Simulation*
BANKS AND FABRYCKY, *Procurement and Inventory Systems Analysis*
BEIGHTLER, PHILLIPS, AND WILDE *Foundations of Optimization*, 2/E
BLANCHARD *Logistics Engineering and Management*, 2/E
BLANCHARD AND FABRYCKY *Systems Engineering and Analysis*
BROWN *Systems Analysis and Design for Safety*
BUSSEY *The Economic Analysis of Industrial Projects*
CHANG AND WYSK *An Introduction to Automated Process Planning Systems*
ELSAYED AND BOUCHER *Analysis and Control of Production Systems*
FABRYCKY, GHARE, AND TORGERSEN *Applied Operations Research and Management Science*
FRANCES AND WHITE *Facility Layout and Location: An Analytical Approach*
GOTTFRIED AND WEISMAN *Introduction to Optimization Theory*
HAMMER *Occupational Safety Management and Engineering*, 3/E
HAMMER *Product Safety Management and Engineering*
IGNIZIO *Linear Programming in Single- and Multiple-Objective Systems*
MIZE, WHITE, AND BROOKS *Operations Planning and Control*
MUNDEL *Improving Productivity and Effectiveness*
MUNDEL *Motion and Time Study: Improving Productivity*, 6/E
OSTWALD *Cost Estimating*, 2/E
PHILLIPS AND GARCIA-DIAZ *Fundamentals of Network Analysis*
SANDQUIST *Introduction to System Science*
SMALLEY *Hospital Management Engineering*
TAHA *Simulation Modeling and SIMNET*
THUESEN AND FABRYCKY *Engineering Economy*, 6/E
TURNER, MIZE, AND CASE *Introduction to Industrial and Systems Engineering*, 2/E
WHITEHOUSE *Systems Analysis and Design Using Network Techniques*

PREFACE

This book deals with discrete simulation. The first part (Chapters 1 to 4) focuses on the fundamentals of simulation, and the second part (Chapters 5 to 9) introduces the simulation language SIMNET.*

In Part I, the subject of simulation is treated in a generic sense to provide the reader with an understanding of the capabilities and limitations of this important tool. Chapter 1 provides an overview of simulation modeling with emphasis on the statistical nature of simulation. Chapter 2 gives a summary of the role of statistics in the simulation experiment. Chapter 3 is devoted to introducing the elements of discrete simulation including types of simulation, methods for sampling from distributions, and methods for collecting data in simulation runs. The material in Chapter 4 deals with the statistical peculiarities of the simulation experiment and ways to circumvent the difficulties arising from these peculiarities.

Although the material in Part I treats simulation in a general sense, it ties to SIMNET by introducing and defining the terminology and the special concepts used in the development of the language. The reader is thus encouraged to review at least Chapters 3 and 4 before embarking on SIMNET.

SIMNET, a network-based simulation language, is detailed in Chapters 5 through 9. Chapter 5 provides an overview of the language and its design philosophy. Details of the language are completely covered in Chapter 6. Chapter 7 introduces

---

* SIMNET is a trademark of SimTec, Inc.

special modeling effects designed to handle "peculiar" situations. In Chapter 8, a wide range of SIMNET applications are introduced, including design of service installations, inventory control, materials handling, job shop scheduling, flexible manufacturing, project scheduling, manpower allocation, and reliability and maintenance. A number of these examples have been simulated in other languages, and their use in this chapter serves to compare SIMNET with these languages. The last chapter (Chapter 9) treats the important topic of analyzing simulation results within the framework of a statistical experiment. The material in this chapter is geared toward showing how the special facilities of SIMNET can be used to carry out statistical analysis of simulation experiments.

My interest in the design of simulation languages dates back to 1962 at Arizona State University when I developed and coded the timekeeping routines that were later integrated into the design of GASP II. Over the past 25 years, I have used simulation extensively in all areas of teaching, research, and consulting. My experience has always favored the use of process-simulation languages (e.g., GPSS) over those of the event-scheduling type (e.g., SIMSCRIPT). This bias stems from the fact that, even though event-scheduling languages offer high modeling flexibility, the compactness and ease of learning that characterize process oriented languages presents a definite advantage.

In spite of this advantage, I have always felt that the basis for the design of presently available process languages is somewhat convoluted. Specifically, the design approach of these languages is based on the use of blocks (or nodes) that perform their functions in serial fashion. For example, suppose that a transaction leaving a queue must dispose of a previously acquired resource before it enters one of several available facilities for processing. The approach followed by these languages requires the leaving transaction to pass through a special block/node to dispose of the resource and then through another special block/node to select a desired facility.

The disadvantage of this approach is that a special block/node must be designed to respond to each distinct modeling need. In addition, the use of such blocks/nodes usually results in a higher degree of abstractness in model representation. The end result is that the language will become less user friendly mainly because of the high level of model abstractness and the large number of blocks/nodes with which the user must deal (for example, GPSS utilizes over 60 blocks). Moreover, the fact that transactions must pass through these blocks/nodes serially does indeed reduce the flexibility of the language and hence gives rise to the need for using external (FORTRAN) inserts. A third disadvantage is that each special block/node is designed to respond to specific modeling needs, an approach that will inevitably lead to a degree of redundancy, and hence inefficiency, in the design of the language.

With these considerations in mind, I embarked in 1980 on the design of a process simulation language. The following objectives were set as the basis for the design of the new system:

1. Achieving flexibility without using (FORTRAN) inserts.
2. Maintaining ease of language implementation.
3. Eliminating altogether the use of special purpose blocks/nodes.
4. Integrating the statistical aspects of the simulation experiment directly into the language.

In essence, my goal was to design a process language that is easy to implement, yet flexible enough to tackle complex problems.

After studying the pioneering ideas of Q-GERT's network simulation (nodes connected by branches), I came to realize that the approach had great potential for meeting my design objectives. Observing that the majority of discrete simulations could be viewed in some form or another as queueing systems, I decided to restrict the new language to three basic types of nodes: a source from which transactions arrive, a queue where transactions may wait if necessary, and a facility where service is offered. My ultimate goal in this regard was to eliminate altogether the use of special purpose nodes.

To abide by the three-nodes restriction, each node was designed to be completely self-contained, in the sense that sufficient information is provided to describe the exact behavior of a transaction as it enters, resides in, and leaves a given node. In essence, such a definition necessitated that a transaction acquire all its "needs" (e.g., resources, next node selection) simultaneously or in parallel, as compared with the serial acquisition approach used by all presently available process languages. The parallel acquisition approach represents a fresh design idea that has proven effective in handling very complex simulations rather conveniently without the need to use external (FORTRAN) inserts.

I spent about one year going through the simulation literature (textbooks and journals) with the objective of ascertaining the viability of the proposed framework in representing any discrete simulation system. The investigation was carried out without any consideration of the programming effort that might be entailed in converting the proposed ideas into a usable computer code, mainly because I did not want to impose any external restrictions on the exploration of new ideas.

The one-year exercise of testing the feasibility of the new design approach convinced me that a viable simulation language could, in fact, be designed without the use of special purpose nodes. The investigation also revealed that the effectiveness of the new language would be further enhanced by adding a fourth node (the "auxiliary" node), which was conceived as an infinite capacity generalization of a facility node.

I started on the computer code for SIMNET early in 1981. Initially, the language was coded in BASIC. However, I soon came to realize the shortcomings of using the then available versions of BASIC to develop a system with this level of sophistication. Subsequently, I abandoned BASIC in favor of standard FORTRAN77. Although the programming complexity reached formidable levels at times, I was able to complete the code without deviating seriously from the basic design framework I had developed for SIMNET. My background in operations research was most helpful in developing the code efficiently and effectively. To cite one example, the intricate chain-effect blocking/unblocking operation of a network of facilities and finite-size queues was resolved in a general sense using recursive computations in dynamic programming.

The initial version of SIMNET was completed in mid-1983. From 1983 to 1986, SIMNET was tested on every possible simulation problem that I could find. Simultaneously, I used SIMNET in the classroom, which was the best way to get the bugs out of the system. It is gratifying to report that I was able to handle conveniently a great variety of complex simulations, all within the coding facilities of SIMNET and without the need to use external (FORTRAN) inserts. (Indeed, SIMNET does not allow the use of FORTRAN inserts.) It is also gratifying to report that my students, both undergraduates and graduates, were able to use SIMNET effectively after two to three weeks of instruction. This result was evident by the complexity of the term projects they submitted at the end of each course. Indeed, many of these projects have been included as problems in this book.

The students indicated that the main attraction of the language was that it required the use of four nodes only. This, together with the powerful file manipulation assignments and the PROCs (used for representing repetitive segments) significantly enhanced the modeling flexibility of the language. Additionally, the explicit error messages and the interactive debugger together with its highly detailed "like-English" trace report were indispensible life savers when it came to debugging SIMNET models. The fact that many runs with different initial data can be executed in a single simulation session was also cited as a great convenience in the use of the language. Finally, the integration of statistical methods (including observations gathering, estimation of transient period, and computation of confidence intervals) were regarded as important features of the language.

SIMNET is available for both the mainframe and microcomputers. Both versions are totally compatible. The language is maintained by SimTec, Inc., P.O. Box 3492, Fayetteville, Arkansas 72702, from whom copies of the system may be purchased. A Teaching Manual consisting of specially designed examples has also been prepared for the language.

Acknowledgments

My sincere appreciation and gratitude go to my students, whose feedback was instrumental in putting the final touches on SIMNET. I would like to cite their names individually in this acknowledgment, but space limitations unhappily preclude the fulfillment of this desire. I wish to give special thanks, however, to my graduate students Jose Pablo Nuno de la Parra, William Pazdera, and Kingsley Ogwu for their untiring efforts in testing the language and editing the manuscript.

Versions of SIMNET were used at the University of Dayton, Oklahoma State University, and the University of Southern California. I wish to thank my friends and colleagues Professors Jay Ghosh, Behrokh Khoshnevis, and Joe Mize for their helpful comments. I should also like to thank my friend Professor C. Robert Emerson for the encouragement and support he gave me in his previous capacity as Head of Industrial Engineering at the University of Arkansas.

A portion of the manuscript was typed while I was on a sabbatical leave with Kuwait Institute for Scientific Research (KISR). I wish also to acknowledge that SIMNET was partly developed on the computing facilities of KISR.

HAMDY A. TAHA

CONTENTS

2 PROBABILITY AND STATISTICS IN SIMULATION 13

3 ELEMENTS OF DISCRETE SIMULATION 35

SIMNET APPLICATIONS