DISK INCLUDED

SYSTEMS ANALYSIS USING SIMULATION AND MARKOV MODELS

John R. Clymer

PRENTICE HALL INTERNATIONAL SERIES IN INDUSTRIAL AND SYSTEMS ENGINEERING

Systems Analysis Using Simulation and Markov Models

JOHN R. CLYMER, Ph. D.
Associate Professor
California State University, Fullerton

PRENTICE HALL
Englewood Cliffs, New Jersey 07632

Library of Congress Cataloging-in-Publication Data

Clymer, John R.
Systems analysis using simulation and Markov models / John R. Clymer.
p. cm.—(Prentice Hall international series in industrial and systems engineering)
Includes bibliographies and index.
ISBN 0-13-881335-3
1. System analysis—Simulation methods. 2. Markov processes.
I. Title. II. Series.
QA402.C535 1990
620'.0042—dc20 89-32564
CIP

Editorial/production supervision and
interior design: Mary Espenschied
Cover design: Wanda Lubelska
Manufacturing buyer: Mary Noonan

Published by Prentice-Hall, Inc.
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 0-13-881335-3

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 Asia Pte. Ltd., *Singapore*
Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*

Systems Analysis Using Simulation and Markov Models

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*
ASFAHL *Industrial Safety and Health Management*, 2/E
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*, 3/E
BLANCHARD AND FABRYCKY *Systems Engineering and Analysis*, 2/E
BROWN *Systems Analysis and Design for Safety*
BUSSEY *The Economic Analysis of Industrial Projects*
CANADA/SULLIVAN *Economic and Multi-Attribute Evaluation of Advanced Manufacturing Systems*
CHANG AND WYSK *An Introduction to Automated Process Planning Systems*
CLYMER *Systems Analysis Using Simulation and Markov Models*
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*
GIBSON *Modern Management of the High-Technology Enterprise*
HAMMER *Occupational Safety Management and Engineering*, 4/E
HUTCHINSON *An Integrated Approach to Logistics Management*
IGNIZIO *Linear Programming in Single- and Multiple-Objective Systems*
KUSIAK *Intelligent Manufacturing Systems*
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*, 7/E
TURNER, MIZE, AND CASE *Introduction to Industrial and Systems Engineering*, 2/E
WHITEHOUSE *Systems Analysis and Design Using Network Techniques*
WOLFF *Stochastic Modeling and the Theory of Queues*

LIMITS OF LIABILITY AND DISCLAIMER OF WARRANTY

The author and publisher have used their best efforts in preparing this book and software. These efforts include the development, research, and testing of the theories and programs to determine their effectiveness. The author and publisher make no warranty of any kind, expressed or implied, with regard to these programs or the documentation contained in this book. The author and publisher shall not be liable in any event for incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of these programs.

Any questions regarding this documentation or about OpEM should be addressed to:

JOHN R. CLYMER & ASSOCIATES
P. O. Box 747
Placentia, California, 92670

TO

my wife Dorothea,

who has chosen to continue living with me
throughout preparation of this book
and has provided much constructive criticism
and assistance in its preparation

Preface

Starting out in systems engineering during the 1960s, I encountered two main problems while doing systems design and analysis: Choosing a world view and building and operating models.

The world view problem occurred because I wanted to perform systems design and analysis taking an operational view. The operational view starts with an analysis of **what** the system is supposed to do (its objectives) rather than **how** it accomplishes its task. An analyst is concerned with the functions that must be performed by the system to achieve these objectives. He or she is interested both in the probability that system objectives are achieved properly and in the time required for a system to perform its tasks. Finally, an analyst is interested in isolating problems that reduce system effectiveness.

Taking an operational view allows consideration of many radically different solutions to a problem. My education had taught me to view systems in terms of structure. This caused me to focus immediately on **how** the system should work, rather than on **what** is to be accomplished. I found that taking the structural view from the beginning eliminated much of the creative thinking that could be applied to the problem, often resulting in a nonoptimal solution.

While researching dissertation topics, I encountered several approaches to system design and analysis that took an operational view. The most promising was based on the concept of a parallel process. A parallel process describes the operation of a system as a set of communicating sequential processes, each representing the execution of system functions. In a parallel process, system functions can interact in two ways: (1) resource contention, in which one function needs a resource currently being used by another function, and (2) functional synchronization, in which one function must wait for another parallel function to achieve its objectives before it can continue. Some analysts view these interactions as messages passed from one process to another.

I decided to use the parallel process concept on a system design problem. To visualize system operation, I represented a parallel process as a formal language model using computation theory. After developing a modified, context-free grammar to represent parallel processes, I used it to assist in the design of a computer-to-computer data communication channel. Results of this project are described in Chap. 5. Using this parallel process operational model gave us considerable insight into better use of the band of frequencies available on the telephone network. My research and the subsequent development of the idea of basing system design and analysis on the concept of a parallel process, expressed as a formal language, are documented in my dissertation (referenced in Chap. 1).

Building and operating models for systems analysis can be considered in terms of five tasks: (1) modeling, (2) programming, (3) verification, (4) validation, and (5) operation.

Of these tasks, modeling was my biggest problem at first. To model, one must analyze a scenario to define system effectiveness and performance measures, state the objectives of the analysis, and develop an explicit definition of system operation. Programming languages such as SIMSCRIPT had been developed, but only an advanced SIMSCRIPT programmer could use it to visualize system operation. Thus, a more general modeling approach was needed. The formal language model, discussed above, helped me to visualize system operation, but it was difficult to use it to explain system operation to others. As the number of people involved in a system analysis project increases, communication becomes more important. The directed-graph modeling approach, described in this book, addresses this difficulty of communicating system operation to all people working on an analysis project.

Another aspect of this problem is verification and validation of model results. The directed-graph model describing system operation helps the analyst visualize parallel process outcomes. An analyst compares his concept of system operation with event-state traces generated by the model to verify model results. Validation is then done by comparing model outcomes with real system operation or specified operation.

Model operation can be difficult. The analyst must relate model results, given by event-state traces or sensitivity curves, to problems in system operation. The directed-graph model and event-state traces help the analyst visualize where problems exist in system operation.

The methodology we developed to solve these problems, called Operational Evaluation Modeling (OpEM) in this book, consists of (1) a graphical language for explicitly defining, understanding, and analyzing parallel processes that can represent a hierarchy of modeling/system complexities; and (2) FORTRAN and Pascal simulation programming systems and mathematical modeling methods to implement the graphical language primitives. This methodology has been applied to the analysis of very large systems such as (1) carrier air defense with destroyer screen and outer-air battle aircraft; (2) strategic defense initiative with sensors, weapons, and BM/C^3 (battle management, command, control, and communications); and (3) air defense initiative with sensors, aircraft, satellites, weapons, and BM/C^3. These models all require complex global control and regulation of system resources. Verifying and validating such models is a problem, because it is difficult to imagine all cases that the logic must handle. This problem is discussed in Chap. 9, where I propose using cybernetic/fuzzy systems theory in building large models.

My first opportunity to apply OpEM to large systems occurred after I joined the United States Navy Fleet Analysis Center (FLTAC) in Corona, California. I led a team that applied the parallel-process concept to evaluation of fleet readiness and surface-ship effectiveness and survivability. Jim Sangamang and Tom Olson of FLTAC contributed heavily to the development of the early OpEM FORTRAN Programming system. We found that the operational view of systems also applied to the evaluation of systems already in place. An operational evaluation of the Corona Airport, an example of such a system, is discussed in Chap. 3.

In 1982 I joined the electrical engineering faculty at California State University Fullerton (CSUF) to teach systems engineering. Sections of this book are based on student projects. The oil-drilling model of Chap. 1 was developed with Timothy Yeh. The multiprocessor and multitasking study in Chap. 3 was originally a project of Margaret Oshiro. The electric motor model in Chap. 3 resulted from a consulting project with Charles Jogwe. Finally, the astronomical observatory study in Chap. 5 is based on a project by Victor Carfi. Other student projects are referenced at the end of Chap. 1.

This book originated as notes for a Master's level graduate course in systems engineering at CSUF. Later, an undergraduate course in systems engineering was created, and introductory material was added to the book to assist beginning students. Chapters 8 and 9 were added more recently to provide advanced topics for practicing systems engineers.

The objective of this book is to teach beginning systems analysis students how to do systems design and analysis. Students would typically be seniors or first-year graduates with management science, engineering, or computer science backgrounds. Model implementation methods using FORTRAN and Pascal are used so that class time is not occupied teaching programming languages. Students should already know how to program in at least one of these languages. Thus, the essential aspects of systems design and analysis can be emphasized, rather than learning a new computer language. Simulation languages can be covered in a subsequent course when students can better appreciate their benefits.

Model implementation based on FORTRAN or Pascal also allows students portable, flexible, and inexpensive access to modeling methods they can readily understand and experiment with. These methods allow students hands-on experience with the internal operation of simulation routines, with building simple models, and with conducting system design-and-analysis experiments using models. A software disk is available in the back of this book to assist students with some experiments. It contains FORTRAN, Pascal, and Markov modeling experiments.

These methods have been extended to allow students to explore advanced modeling topics such as object-oriented, rule-based simulation. Thus, the book allows students to experiment, starting with simple models and continuing with more complex ones, to gradually gain an appreciation of state-of-the-art system design and analysis concepts.

This book is intended to teach students system design and analysis, but it is not a survey of Operations Research (OR). A course using this book might come after a survey course in OR. Several fine texts are available for a beginning OR course, such as those by Hillier and Lieberman, Phillips, and Taha.

Our book focuses on simulation and Markov modeling methods, showing the student how to use OR tools to do systems design and analysis. It is applications-oriented and has many simple- to intermediate-level examples. It does not provide a survey of simulation methods, as the text by Banks and Carson does. Our book discusses one proven method, and we apply it to many examples.

Chapter 1 provides a discussion of the basic ideas and concepts of operational evaluation modeling. Chapter 2 defines the concept of a parallel process using a directed-graph notation and provides basic simulation methodology. Directed graphs are two-dimensional representations of parallel-process concepts. Chapter 3 provides three example applications of discrete-event simulation. Chapter 4 presents Markov chain models, which are one way of expressing parallel processes mathematically. Chapter 5 documents two operational evaluation studies based on Markov models. Chapter 6 presents reliability theory expressed in parallel process concepts, taking the operational view of systems. Chapter 7 develops queuing theory models as parallel processes and presents results for single-queue, single-server and single-queue, two-server systems using discrete-event simulation, Markov process, and queuing theory models. Comparison of these three techniques demonstrates the fundamental assumptions of each. Chapter 8 presents the Pascal simulation programming system for discrete-event simulation. This chapter is based on a project done at CSUF by Nafise Nili. Chapter 9 discusses cybernetic system modeling. The operational view is shown to implement Ross Ashby's concept of cybernetics. An expert system, implemented using "fuzzy" logic, allows global control of a complex parallel process. The last third of this chapter is based on a Master's thesis written by Michelle Knowlden.

Courses that might follow from this book are simulation languages; knowledge-based simulation; object-oriented, rule-based simulation; distributed simulation; and advanced statistics.

I wish to thank my typist, Kathleen A. Crum, and everyone who has contributed to this work, directly or indirectly. I especially wish to thank the major contributors:

Philip D. Corey, Ph.D.
Rockwell International

Steven L. Goodfriend
Rockwell International

Michelle L. Knowlden
Hughes Aircraft Company

Nafise Nili
Rockwell International

John R. Clymer

Contents

1

Basic Concepts in Operational Evaluation Modeling

Modeling is done to experiment with systems. A system can be a factory, a business organization, a biological organism, or an electronic apparatus, among other things. A model is a mathematical representation of the operation or behavior of a system.

A model is particularly useful for experimentation when it is not feasible or cost-effective to use the real thing. For example, an electronic system in the conceptual stage of design does not exist and so cannot be observed in operation. Likewise, a factory owner would not want to purchase all the equipment that might improve productivity just to evaluate it.

A model is also used to gain insight into system dynamics when such insight cannot be obtained by observing the real thing. Sometimes such observation disrupts normal system operation. Examples are studies of biological systems or traces of computer system operation.

This book describes a form of modeling that we call Operational Evaluation Modeling (OpEM). In this chapter we discuss the basic concepts of OpEM. A general discussion of modeling presents a comparison of the structural versus operational view of systems. OpEM is described as a system design and analysis methodology. The relationship of OpEM to systems analysis, operations research, and systems engineering is explained. Three major application areas are discussed. Next, the utility of OpEM for people with different roles in a system-design-and-analysis project is discussed. Objectives of OpEM application are presented, along with basic OpEM concepts used for achieving them. The chapter concludes with three example models to illustrate these basic concepts. The cybernetics view of OpEM is presented as a preview of advanced topics discussed in Chap. 9.

STRUCTURAL VERSUS OPERATIONAL VIEW OF SYSTEMS

Some models involve solving differential equations. These models use a collection of state variables in vector form, $\mathbf{X}(\mathbf{t})$, to represent the state of various elements of the system. The state variables are interrelated to model system dynamics using vector differential equations of the following form:

$$\mathbf{X}'(\mathbf{t}) = F[\mathbf{X}(\mathbf{t}), \mathbf{U}(\mathbf{t}), t]$$

where $\mathbf{U}(\mathbf{t})$ is an input vector representing environmental factors, $\mathbf{X}(0)$ are initial conditions at time zero, and t is the independent variable.

Early models of this type were solved mathematically without sophisticated machines using analytical methods such as transform techniques. Analog computers allowed more complex systems to be studied. With the digital computer, numerical techniques could solve large systems of differential equations. Later, digital filters were developed to numerically implement transfer functions that result from transform solutions. These have evolved into the continuous system simulation languages now available that are used to implement the structural view of systems.

As the digital computer was developed, another view of systems emerged—the operational view. Models based on this view are concerned primarily with *what* a system does rather than *how* it is done. The system is viewed only at certain discrete times, called events, and is assumed to be composed of a set of processes operating concurrently. Some of the state variables in the state space are discrete states, each representing the time required to execute a system function, rather than all continuous states as in early models. The concept of a process outcome as a sequence of system states (combined discrete and continuous) and events (changes of system state) resulted. Such models based on purely discrete events are sometimes called discrete-event simulations. Operational Evaluation Modeling (OpEM) evolved from these concepts. References Ahmed,[1] Clymer,[3–12] Crabbs,[13] Holzinger,[19–20] Kasahara,[21] Quandt,[26] Ramirez,[27] and Yeh[30] document part of this evolution.

An alternative to the operational view is the structural view, which is popular among some analysts. Again, a model based on this view may consist of discrete or continuous states and events, but this view is more closely related to *how* a system accomplishes objectives, rather than *what* it does. Figure 1-1 compares these two views of systems.

In the structural view, a system is an interconnected set of structural elements. Interfaces among elements define how they communicate with each other. Operation of each element is determined by transformations that relate inputs to outputs.

In the operational view, system operation is represented by a parallel process, defined as the set of all possible sequences of states and events that occur. The sequences of states and events may relate to one or many structural elements or, alternatively, structural elements may not be identified at all. Interfaces take place between processes and are called process interactions. One such process interaction is called resource contention. Operation of each function of the process is described by what are called mission attributes, for example, reaction time or probabilities of correct operation, which are parameters that characterize process behavior.

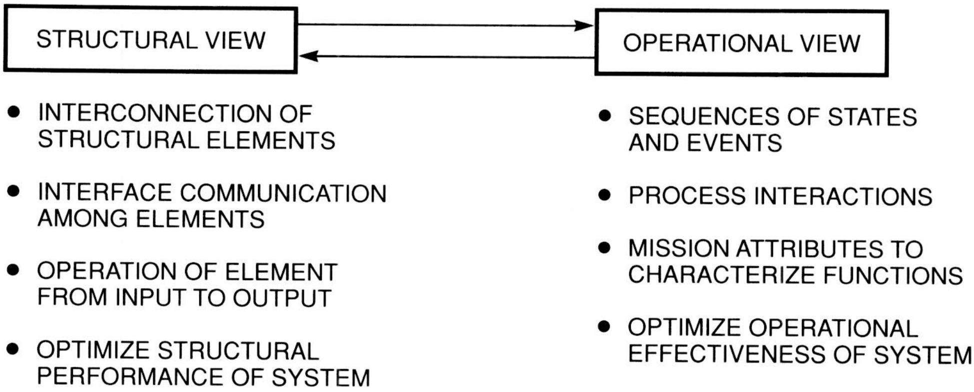

Figure 1-1 Structural versus operational view of systems.

The structural and operational views are two distinctly different ways of thinking about a system. Both views are required to describe a system, but something else is required to complete the description. A simplified definition of system will clarify this point: A system consists of a set of resources, a set of operations, and a set of goals. An example is a small fish (a system) whose environment includes a large fish. The set of operations could be to run, to hide, or to attack. The resources could be fins, teeth, and a hiding place. The goals could be to survive and to protect territory.

A system contends with its environment to achieve its goals. A system must decide which operations to perform to achieve its goals under constraint of limited resources. The fish has resources either to run and hide or to attack. What he decides to do depends on his conflicting goals.

The organization of the resources relates to system performance and is the concern of the structural view. The organization of the operations relates to system effectiveness and is the concern of the operational view. The organization of the goals relates to both structure and operation; it is the concern of the cybernetic system concept. The structural and operational views plus the cybernetic system concept combine to define a system.

The objective of this book is to present a system-design-and-analysis methodology (OpEM) that integrates the structural and operational views with the cybernetic system concept to create one holistic paradigm. We begin by visualizing system operation as a parallel process. Then we use timeline analysis of several operational scenarios to learn the basic concepts. We describe a two-dimensional graphical language that further assists visualization and also defines the structure of operations for the operational view. This structure is then shown to be the basis of the system-design-and-analysis methodology (OpEM). Many real-world applications of OpEM are provided. The stuctural and operational views are discussed in Chaps. 1–8.

Much of the complexity in the analysis and design of systems results from the organization of goals. System goals often conflict, with many interacting rules for making decisions. Routines to make these decisions in simulation programs are sim-