TEACHING SIMULATION TO UNDERGRADUATES

by

Stephen D. Roberts
School of Industrial Engineering
Purdue University
West Lafayette, Indiana 47906

PANEL MEMBERS

Jerry Banks
School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332

James Kho
Department of Computer Science
California State University
Sacramento, CA 95929

Udo Pooch
Department of Industrial Engineering
Texas A & M University
College Station, TX 77843

John Ramberg
Department of Systems and Industrial Engineering
University of Arizona
Tucson, AZ 85721

The purpose of this panel is to discuss the teaching of simulation to undergraduates. Traditionally, simulation has been taught at the graduate level with emphasis on simulation methods and tactics, particularly the statistical aspects. However, the explosion in computer technology and the growth in simulation languages has brought about widespread application of simulation to a variety of problems. In many cases, simulation is the "tool of first resort" because it places the fewest constraints upon its application and usable models can be obtained quickly. Nevertheless, exactly what should be taught at the undergraduate level is a source of controversy. Can undergraduates be adequately prepared to provide simulation services or should they be exposed only to fundamentals, leaving their development as simulation modelers to another segment of their education? All the members of the panel are actively involved in teaching simulation to undergraduates. They will share their particular perspectives on what they do at their institutions and why. Obviously, opinions will differ but by considering the different arguments you can better judge what should be done to improve the instruction of these impressionable future simulation contributors. The position statements of the panelists follow.

Stephen D. Roberts

Every senior in Industrial Engineering at Purdue University must take IE 431: Systems Analysis and Design. These students will have had previous preparation in the methods of statistics (two courses) and operations research (two courses). They have had individual courses in work methods and measurement, manufacturing processes and machining, engineering economy, production planning and control, and usually are taking facilities design and materials handling. The purpose of the systems analysis and design course is to round out their undergraduate program by the examination of systems -- attempting to integrate the topics of previous courses.

It has been my belief that the best way to study the topic of systems analysis and design is through simulation. Thus, while simulation methods and tactics are an essential ingredient of the course, the primary emphasis is on simulation modeling and its application to industrial engineering problems within a design context. Courses at Purdue are a semester long (15 weeks of class plus 1 week for finals). The course carries three credit hours. We meet twice a week in 75 minute periods.

The classroom discussion is reinforced by homework assignments (usually 8). For the most part, these are modeling assignments involving, at times, extensive computer interaction. Homework has been essential to insure that the students have practical experience in examining a wide variety of simulation tactics and problem environments. Although all students have had some previous computer exposure, their experience is spotty (many have not used it in two years) and the homework provides them with intensive exposure to

Proceedings of the 1982 Winter Simulation Conference
Highland * Chao * Madrigal, Editors

82CH1844-0/82/0000-0706 $00.75 © 1982 IEEE

the computing system including the editors, compilers, and utilities. An important additional benefit of the homework is that students learn to write up their simulation problems in a style that emphasizes insight based on simulation experiments.

All students have had one course in FORTRAN, although their knowledge is superficial and distant (two years prior). Many have some exposure to BASIC and some have had Pascal. Because the course emphasizes modeling, requiring a new general purpose programming language embodying simulation like SIMSCRIPT or SIMULA would be time consuming and detract from modeling. SLAM is inappropriate because it requires extensive FORTRAN preparation (which our students do not have) and using it would duplicate the graduate simulation course. GPSS/H would be a viable candidate but it is unavailable at Purdue. Therefore, I use INSIGHT, a powerful, general purpose simulation modeling language I developed that does not depend on any programming knowledge and has been constructed to appeal to those whose background is similar to our students. The use of this simulation language promotes the modeling viewpoint.

An essential ingredient in the course has been the term project which is conducted by teams of three to four students. The team is required to identify and solve a "systems problem" from a "real world" environment. The students make the contact and prepare a professional (written) report for their client (and the course). The important learning experience from conducting such a project, aside from its execution, is the interaction with the "problem identification and problem definition" phase of systems analysis. Taking an abstract notion of a problem completely through the design stage has been a rewarding and enlightening experience that instills confidence in the students and provides them with a factual appreciation for the entire system analysis and design process.

I believe that upon completion of the course, the students can use simulation to conduct systems analysis and design. They understand how to use the simulation model legitimately to generate insight into the problem and obtain valid inferences. It is the insight that produces results.

Jerry Banks

I. What Is Required

Our students are required to take a three quarter hour course entitled SIMULATION. This course requires elementary statistics and computer programming as prerequisites. We provide the students with an introduction to discrete-event simulation, with an emphasis on methodology. Accompanying the lecture portion of the course is a weekly laboratory in which students are taught to program and solve simple problems using a special-purpose simulation language.

II. What Objectives Are We Attempting to Accomplish?

1. To introduce the art of simulation.
2. To illustrate classes of problems that are amenable to discrete simulation.
3. To describe when and how analytic solutions can be obtained to waiting line models.
4. To provide an understanding of some statistical methods which are useful to discrete event simulation.
5. To introduce proper data collection and analysis procedures.
6. To describe the most basic tenets of output analysis.
7. To introduce students to GPSS.

III. What Topics Are Covered?

1. Concept of system, model, and related terminology.
2. Examples of discrete simulation (queuing, inventory, reliability, and maintenance).
3. Statistical models often used in discrete simulation.
4. Queuing models (transient and steady state).
5. Random numbers (generation and testing).
6. Random variate generation (inverse transform, convolution, acceptance-rejection).
7. Input data analysis (parameter estimation and hypothesis testing).
8. Output data analysis (one system and several alternative designs).
9. GPSS (currently V, switching to H).

IV. What Texts Are Used?

Lecture

Banks, Jerry and J.S. Carson, II, Discrete Event System Simulation, Prentice-Hall, Englewood Cliffs, N.J., to appear, September, 1983.

Laboratory

Gordon, Geoffrey, The Application of GPSS V to Discrete System Simulation, Prentice-Hall, Englewood Cliffs, N.J., 1975.

V. What Else Can Students Take?

We offer ADVANCED SIMULATION, a course for undergraduates, once per year. This course is a continuation of the required course. The emphasis is on discrete-event simulation methodology within the computer (time keeping, file keeping, arrival events, departure events, statistical reporting, etc.), validation techniques, inference, comparison of several system designs and optimization techniques. The refinements of a special purpose simulation language are taught in a weekly laboratory. The students are assigned a term project which consists of performing a simulation study of a real system.

VI. Comments

We are providing the students with an overview and appreciation of simulation in the required course. We have nearly 300 students completing this course each year. As many as five different people will teach the required course in a year, all of them following the same course documentation, but varying the emphasis within the documentation.

We have shied away from mathematical sophistication in our two undergraduate courses. For example, we don't teach spectral analysis, regenerative method and time series analysis.

We feel that teaching a special purpose simulation language accompanying the lecture is a must. We want the students to have an appreciation for solving problems larger than the paper and pencil variety. Also, we like a special purpose language (over FORTRAN, for example) because the student can have successful experiences throughout the course.

James Kho

Computer Science majors at California State University, Sacramento are required to take C Sc 148 System Simulation. This course is also often chosen as an elective by engineering students. The prerequisites for the course include 6 semester units in statistics and probability and proficiency in at least one higher-level programming language. The objective of the course is to familiarize students with simulation methodology, with a heavy emphasis on the simulation of discrete stochastic systems. A brief portion of the course covers continuous system simulation. It should be noted that students may elect to take additional courses related to simulation. These additional courses include Systems Analysis and Design I and II, Performance Modeling and Measurement, Simulation for Managerial Decision Making, and a graduate course in Simulation Methodology. The discussion here centers around the first basic course.

The course presents simulation and model building as tools for understanding the expected actions of real systems. The distinction is made between continuous and discrete systems through the discussion and use of two distinct modeling approaches and their associated simulation languages. Course assignments involve the students in the use of GPSS and DYNAMO as they formulate and implement computer simulations. The course carries three semester credit hours.

Classroom time is spent strictly on lecture and discussions and students are expected to write, debug, and test programs outside the classroom. Since students are expected to be familiar with the computer system used (a CDC CYBER 170/730), a minimum amount of time is spent on system utilities and the editor.

Proficiency in higher-level languages such as PASCAL or FORTRAN has not always been a plus since the design of GPSS departs radically from that of statement-oriented languages. The flow of transactions from block to block in GPSS presents a new and different approach to most computer science students. This is, in fact, one of the main reasons why GPSS is chosen over other discrete simulation languages. Before being exposed to simulation languages, students are asked to implement the simulation of a simple system using the language with which they are familiar (usually requiring some 50 or more statements). An appreciation of simulation languages is gained quickly when they realize that the same program can be implemented in GPSS in about 10 blocks. A survey of simulation languages is included in the course.

A strong emphasis on the use of statistical tools, testing and validation is made. A danger of computer modeling, as opposed to mathematical modeling, is that errors are not as easily detected. All too often, conclusions are drawn from simulation results that may be inaccurate or incorrect due to faulty design.

The course concludes by bringing the techniques, tools, and knowledge developed earlier to bear on a specific simulation application. A team effort takes a problem completely through analysis and design to implementation and validation. The team is expected to submit a report including the conclusions reached supported by the simulation results.

John S. Ramberg

Digital systems simulation (SIE 405) is a required senior course in both the system engineering and the industrial engineering curricula at the University of Arizona. Prerequisites are (i) a two semester sequence in probability and statistics for engineers, typically taught at the level of Hogg and Tanis [3] (SIE 320) and Hines and Montgomery [2] (SIE 420), and (ii) a two semester sequence in operations research, typically taught at the level of Taha [4] (SIE 340) and Hillier and Lieberman [1] (SIE 440). All of the students have also taken Introduction to Systems and Industrial Engineering (SIE 150), Engineering Problem Solving with FORTRAN (SIE 170), and Engineering Numerical Analysis (SIE 270), all of which have strong programming components. These students also have had five semesters of calculus level mathematics and the usual engineering science core. Thus they should be well grounded in the programming/systems/statistics/operations research methodologies. The purpose of this course, in addition to teaching simulation, is to integrate these previously learned materials in preparation for industrial job situations. Students in this simulation course, as well as in our other computer courses, use a VAX 11-780 interactive computing system.

Having stated the background of the students, I will summarize my own background which influences my choice of materials for this course. My graduate studies were principally in the stochastic and statistical aspects of operations research and my training in simulation was provided by Professors William Maxwell and Richard Conway at Cornell University. My ideas toward course content are heavily influenced by their philosophy as well as by my teaching and research experience at the University of Iowa and later at the University of Arizona. I have taught undergraduate engineering courses in simulation, continuing education courses in simulation, and graduate research courses in simulation. My research in the statistical aspects of simulation includes random variate generation, statistical modeling of inputs, the design of computer simulation experiments and the analysis of simulation output. Note the absence of any research in simulation languages, which may explain some statements that follow.

Over the fifteen years that I have taught simulation, the statistical content has not changed substantially. The topics that I teach are nicely presented in Chapters 5, 7, 8, 9, and 12 of the text by Law and Kelton. During the past five years, as random variate generation software has become more readily available, I have deemphasized this topic somewhat. More time has been devoted to the selection of statistical models for input (a topic that does not seem to receive adequate attention in engineering statistics courses), the design of simulation experiments and the analysis of simulation output.

The choice of a simulation language has not been as simple for me and my decisions have often been influenced by the availability and usability of teaching materials and the programming language. Since our students have a reasonable background in FORTRAN, we desire a simulation language that can be learned easily and quickly. This allows the teacher to concentrate on the simulation modeling methodology and its application to systems problems. I have used FORTRAN, GASP, GPSS, SIMLIB, SIMSCRIPT, and SLAM with varying degrees of success. At this point, I prefer to use a language such as SIMLIB or GASP, since they are FORTRAN based and facilitate the understanding of the time advance system and filing system necessary for efficient simulation languages. We are currently using SIMLIB, a decision that was coincident with the arrival of Averill Law on our campus. I recognize that the student may need to learn a new language upon employment, and that there are many possible ways to do this if he has a fundamental background in simulation.

The selection of course content and particularly the selection of a computer language and the decision as to the balance between language, modeling methodology and statistical methodology derives from our department's two martini rule (not to be confused with the three martini lunch), as stated by Wymore [5]. "If we are comparing two possible courses which could be offered based on ideas from our basket, we would prefer one of the courses to the other if the one was more fundamental -- if by taking one of the courses in school, our engineering student, now on the job, after eight hours at the plant, commuting home, two martinis, putting the kids to bed and attending the PTA, could sit down with a textbook for the other course and have some hope of conquering the material."

SIE 405: Credit = 3. GC I Simulation modeling of systems using digital computer languages, including random variate generation, Monte Carlo, timekeeping structures, input modeling, systems modeling, statistical design and analysis of simulation experiments.

Textbook:

Simulation Modeling and Analysis, A.M.Law and W.D. Kelton, McGraw Hill, New York, 1981.

References:

System Simulation, Geoffrey Gordon, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1978.

Simulation Using GPSS, Thomas Schriber, Wiley, New York, N.Y., 1974.

Introduction to Simulation and SLAM, A.A.B. Pritsker and C.D. Pegden, Ed. Halstad Press, West Lafayette, Indiana, 1978.

The GASP-IV Simulation Language, A.A.B. Pritsker, Wiley-Interscience, New York, N.Y., 1974.

Concepts and Methods in Discrete Event Simulation, G.S. Fishman, Wiley-Interscience, New York, N.Y., 1973.

Goals:

To introduce students to the advantages and limitations of digital computer simulation methods in the design and analysis of complex systems. By the end of the semester, students successfully completing the course should be able to:

a) judge when, if, and at what level simulation should be used to help solve a particular problem;
b) select appropriate techniques for a variety of problem types;
c) select a simulation language for solving a specific problem;
d) select appropriate statistical models for simulation inputs;
e) design a simulation model and implement it in at least two programming languages;
f) design a simulation experiment and analyze the output;
g) organize and prepare a written statement of a simulation proposal and study.

Prerequisites by topic:

1. FORTRAN Programming
2. Probability topics including conditional probability, transformation of variates, discrete and continuous distributions, and expectation.
3. Statistics topics including estimation, hypothesis testing, confidence intervals, and intro to experimental design.
4. Operations research topics including queuing and production and inventory models.

Topics:	Class Sessions
1. Course Organization, Introduction, and computing details	3
2. Monte Carlo	3
3. Distribution Sampling and	5
Random Variate Generation	2
4. Systems Modeling	1
5. FORTRAN Simulation Models of Queuing and Production/Inventory Models	4
6. Simulation Language Survey and Study of a Language Selected from GPSS, GASP, SLAM, SIMLIB	10
7. Variance Reduction	2
8. Tactical Planning and Output Analysis	5
9. Experimental Design	2
10. Validation and Verification	1
11. Exams and Solutions	3
	41

Laboratory Project:

No lab associated with class. However, nearly all of the class assignments require designing, programming, and running computer programs in FORTRAN and in selected simulation languages.

Estimated Content:

Engineering Science: 1 1/2 credits,
Engineering Design: 1 1/2 credits.

References

1. Hillier, Frederick S., and Gerald J. Lieberman. Introduction to Operations Research, Holden-Day, Inc., San Francisco.

2. Hines, William W. and Douglas C. Montgomery. Probability and Statistics in Engineering and Management Science, 2nd Ed., John Wiley and Sons, New York.

3. Hogg, Robert V. and Eliott A. Tanis. Probability and Statistical Inference, MacMillan Publishing Co., Inc., New York.

4. Taha, Hamdy A. Operations Research An Introduction, 2nd Ed., MacMillan Publishing Co., Inc., New York.

5. Wymore, A.W. "A Model Curriculum in Systems Engineering", The Journal of Systems Engineering, Vol. 1, No. 2, April, 1969.

Udo W. Pooch

I EN 489 and CS 489 (Special Topics Course Simulation) represent the undergraduate versions of respective graduate courses in both I EN and CS.

It is appropriate to point out that the department of Industrial Engineering at Texas A & M University is composed of several academic divisions, of which Industrial Engineering (with 1000 majors) and Computing Science (with 1500 majors) are the two major divisions. Each of the academic disciplines have their own degrees, ranging from BS, MS, to Ph.D. The two courses, because they originate from within their two academic disciplines, have some similarities but considerable differences, as well as different prerequisites.

The Computing Science course is more theoretically oriented with emphasis on computer systems simulation, while the Industrial Engineering course emphasizes the applications of simulation technology. Students in the department have the option and the opportunity to take either of the two courses depending upon the emphasis of their plan of studies. Both courses are 3 credit semester courses for 16 weeks, meeting either MWF 3 times a week, or TTH 2 times a week.

With respect to the Computer Science program, the students will have had preparation in statistics, calculus and operations research. Within their major, they will have had individual courses on computer architecture and organization, discrete structures, data bases, and data structures, surveys of high level languages, operating systems, information systems and system management.

Students have also been exposed to the fundamental methods of operations research which include linear programming, dynamic programming, deterministic models for inventory and production control, and applications of queuing theory.

Once in the course, the students are introduced to simulation and comparison with other problem solving techniques, discrete simulation models and review of queuing theory and stochastic processes, comparison of discrete change simulation languages, simulation methodology including generation of random numbers and variates, validation of simulation models and results, and related applications of simulation.

All students have had several courses in programming languages including FORTRAN, assembly language, PL/I, COBOL, APL PASCAL, ADA and overviews of data processing, character string processing and list processing languages.

Programming projects, given to the students to expose them to such languages as GPSS, GASP, SIMSCRIPT, and CSMP, include individual projects as well as team projects.

With respect to the Industrial Engineering program, the students have had preparation in engineering costs, economic analysis, engineering and systems analysis, production engineering, analysis and design, statistical control of quality and work measurement. They have, of course, also been exposed to calculus, statistics, design of systems and experiments.

The course principally deals with simulation methods and applications, and thus exposes students to fundamental methods and methodology of systems simulation, random number generation, random deviate generation, clock routines and statistical analysis of simulation models. It also includes discrete simulation, continuous simulations, and combined simulations. GASP IV is stressed as the primary simulation language in all of the programming projects, although students are also exposed to the fundamentals of GPSS, SIMSCRIPT, and Industrial Dynamics.

The programming projects present little difficulty to the students, since they have already had courses in FORTRAN and PL/I. In either of the two courses, we see that we have students who have maturity in programming languages and the corresponding interaction with both mini and large scale computers, a significant background in mathematics and operations research, and thus provide to us, the teachers, an unusual opportunity to discuss systems analysis and design at a very high level.