COMPUTERS IN EDUCATION
R. Lewis & D. Tagg (editors)
North-Holland Publishing Company

TEACHING THE ART OF MODELING AND SIMULATION AT A TECHNICAL UNIVERSITY

Francois E. Cellier
Institute for Automatic Control
ETH - Zentrum
CH-8092 Zurich
Switzerland

Jakob S. Vogel
IBM Switzerland
General Guisan Quai 26
CH-8002 Zurich
Switzerland

Simulation techniques have become recognized analysis and synthesis tools in many different disciplines of engineering and natural sciences. Therefore, it is recommendable to offer an introductory course of modeling and simulation at a technical university already to undergraduate students. They may profitably attend such a course as soon as they have acquired sufficient knowledge of programming to understand, and of their particular area of interest to appreciate a course of simulation.

Experience with teaching the art of modeling and simulation at ETH Zurich led over many years to a concept where the students (in approximately equal parts) get acquainted with the theoretical basis of modeling and simulation, are taught how to use two of the more frequently used simulation languages (CSMP for continuous system simulation, and GASP for discrete event and combined continuous/discrete system simulation), and acquire practical experience by solving small scale simulation problems on the computer. The success of a simulation course depends substantially on the available computer facilities and on the documentation placed at the students' disposal. It is of utmost importance to carefully choose the theoretical topics to be presented in such a course, to balance between the diversity of topics and the depth of elaboration.

The aim of this paper is to describe both form and contents of the lecture on simulation techniques being offered at ETH Zurich by the authors.

1. INTRODUCTION

Lectures in which the art of modeling continuous and/or discrete systems is taught have belonged for years to the standard courses in applied data processing. In the early days of simulation, the mapping of the behavioural attributes of technical systems into a computerized model was mostly done by control engineers who found their way from pure analog through hybrid to pure digital computation. Meanwhile, the users of simulation techniques are spread over the entire range of engineering and natural sciences. Even medical doctors and economists (the latter primarily thanks to Industrial Dynamics, a modeling technique introduced by Forrester and his colleagues in the early sixties [6,14]) have learnt to use and appreciate the advantages of computerized models.

At ETH Zurich, simulation lectures have been offered for twelve years by now. Experience has shown that those attending these stem from various departments, and, consequently, dispose of a quite diverse background [8]. Furthermore, they come to this course with quite distinct expectations. Both the selection of the presented topics, and the organization of the -- for such type of lectures indispensable -- computer exercises must take this diversity into account. This seems to be of primary importance as we are convinced that knowledge of the principles of simulation should belong to the general education of anyone completing his studies at a technical university.

The "ideal modeler" can be characterized by the following attributes:

- His ability to analyse a "real" system, to separate relevant from irrelevant attributes, and to represent the former by mathematical relations.
- His knowledge of computer techniques which enable him to code those mathematical relations into a computerized model, possibly by allowing tolerable simplifications. For this task, he uses an adequate tool, preferably a piece of software being equipped with special purpose simulation operators, that is: a simulation language.
- His ability to collect an appropriate amount of data relevant to the modeling task, and to arrange those data in a form suitable for simulation.
- His experience in experimentation with models [9], his capability to recognize which mathematical and statistical techniques and tests are appropriate for the task, and his capability to determine type, number, and length of the required simulation runs.
- His knowledge of techniques to evaluate the

validity of a model for its intended use, and his care in verifying the obtained simulation results with respect to the underlying model [15].

- His aptitude to convince the management of his enterprise of appropriateness, relevance, and range of applicability of the obtained simulation results, as well as to prove their economy.

It is certainly impossible to turn a student into this "ideal modeler" by means of a single one-term lecture. Nevertheless, he may advance quite a bit along this line if he brings with him an appropriate background. This includes:

- Experience in using one of the available higher level programming languages, preferably FORTRAN due to the fact that:
  a) most of the currently available simulation languages are either FORTRAN programs by themselves (like GASP [1,12]) or translate into FORTRAN (like CSMP [16,17]) enabling the user to code portions of his model directly in FORTRAN,
  b) FORTRAN allows access to a large variety of available subroutine packages and application programs, and
  c) FORTRAN is still the widest distributed programming language for scientific computations, especially in industry.
- Basic knowledge of differential and integral calculus, as well as of numerical mathematics and statistics.
- A sufficiently profound knowledge of the application area the student is working in, and particularly of the mathematical laws governing the behaviour of systems in this application area.

All of these prerequisites make it preferable to offer a course of simulation during the second half of the undergraduate studies, that is: from about the fifth term on.

2. FORM OF THE LECTURE

When teaching simulation during a course in industry, time constraints force us to organize the course as a concentrated workshop with a duration of no more than a week. It is obvious that under these conditions the course must be somewhat narrow in scope, basically concentrating on some principles of simulation and on an introduction to syntax and semantics of a particular simulation language.

When teaching simulation during lectures at university level, it is preferable to distribute the course over at least one entire term since this gives the students a chance to deepen their understanding of the presented topics by computer exercises. Since the students are expected to make mistakes while coding their first simulation programs, they may have to submit their programs several times before they are even syntactically correct. This procedure is time consuming and, therefore, entirely out of question for a concentrated workshop. This painful "try-and-error" procedure is, however, unavoidable on the way to practical application of simulation techniques. The attendees of an industrial simulation course are expected to be motivated for the course by "real" problems they have to solve. It is, therefore, most likely that, immediately after the course, they will apply the freshly acquired knowledge to the solution of their problems. This assumption does, however, not hold for students at a university. They are basically motivated by the fact that it is "their duty" to attend a certain number of lectures each year. After a short workshop on simulation, they would immediately concentrate on another topic again by never surmounting the difficulties of the beginner [4]. This threshold is afterwards much more difficult to take when the lecture belongs to the past (the subject matter being no longer so freshly present). A good number of such students would never apply simulation techniques in their further life, simply because they failed to lose their fear to approach the computer in time. Then, attendance of the simulation course would certainly have been a badly invested effort.

Also the contents of a simulation lecture at university level will substantially differ from the topics discussed in an industrial course since there usually will be sufficient time available to include, to a greater extent, presentation of techniques for construction and assessment of models, to discuss numerical, program-technical, and organizational aspects, to show how simulation results are interpreted correctly (simulation results are not necessarily "true" just because the computer displays 14 digits (!)), and to investigate in aspects of simulation studies from the point of view of their economy.

Our experience has shown that simulation lectures are profitably partitioned into three parts of approximately equal length, all three of them being of equal importance:

a) A more theoretical part in which primarily techniques for model creation, techniques of simulation, and frequently used mathematical algorithms are presented.

b) A syntactical part in which -- basing on an available modern simulation package (like CSMP or GASP) -- the linguistic and program-technical attributes of the software are discussed to such an extent as it is needed to enable the students to "solve" their computer exercises by coding small scale simulation programs during:

c) an exercise portion in which -- by coding small scale simulation programs from diverse scientific and technical disciplines -- the students may deepen their understanding of both theoretical and program-technical concepts, and in which typical simulation mechanisms (described in chapter 3 of this paper) are illustrated.

The importance of (c) has been discussed above. The timewise separation of (b) from (a) seems fruitful since otherwise some students are inclined to believe that they have understood the principles of simulation, simply because they know that e.g. in CSMP a continuation of a statement is indicated by three consecutive periods (!). This separation need, however, not even be noticed by the students.

The selection of the simulation software used for the exercises is of secondary importance. Most universities will have to base their decision anyhow on what is available for the locally installed computer system. There exists for any type of simulation problem a variety of different simulation languages which are comparable in programming comfort and also concerning the features they offer. Taking the limited programming experience of students into account, it seems advisable to use an integrated programming package with a language definition of its own (simulation language) rather than to rely on a FORTRAN subroutine package. Simulation languages are somewhat easier to use. The student can, therefore, put more emphasis on the modeling task rather than being forced to concentrate on rigid coding formats or other coding particularities. Subroutine packages, on the other hand, may have some advantages for experienced users involved in a large scale simulation study. It would, of course, be nice to use one programming package throughout the lecture to minimize the number of hours being spent on (b). Unfortunately, there does not yet exist an appropriate piece of simulation software which, on one side, is an easy to use simulation language, and, on the other hand, allows one to code simulation programs for a continuous system, discrete event, and combined continuous/discrete system simulation. This is one of the reasons why we are currently involved in developing our own simulation language (COSY [2]) which will meet these requirements. Currently, we use CSMP during the first part of the lecture which deals with continuous system simulation. During this part of the lecture, the student gets acquainted with many basic principles of simulation which are equally valid for discrete event simulation. He, furthermore, gains some programming experience during the exercises. For the second part of the lecture which deals with discrete event simulation, we could use GPSS. Disadvantages of this choice would be that the student would not easily see the similarities and common principles of both types of simulation problems. For this reason, we decided to base this part of the lecture on GASP-V [1] being a FORTRAN subroutine package for continuous, discrete, and combined simulation, by accepting the discomfort of extensive FORTRAN programming as the smaller of two evils (this concept has also its advantages as shown in [3]). This choice allows furthermore to introduce smoothly also the concepts of combined continuous/discrete system simulation. In future, we shall switch to COSY which is a front end to GASP-V. A PASCAL coded preprocessor translates COSY programs into FORTRAN by generating subroutines which are to be executed together with the GASP-V system.

For the exercises, it has proved useful -- both for the modeling phase and for the testing and debugging phase -- to let the students work in groups of two or three together.

Concerning the operational mode, it seems to us that students who do not yet dispose of much programming experience should use batch operation rather than interactive programming. When operating from a terminal, the students tend to correct each error immediately after they detect it, and then rerun the program without carefully investigating what are the effects of this correction on the model as a whole. Sitting at a computer terminal, the inexperienced student finds himself almost automatically in a stress situation (since the computer "waits" for input) which often inhibits global structural errors and/or complicated numerical problems to be properly analysed and resolved. It is, therefore, better to let the team sit around a table in peace with the computer printout in front of them in order to have them discuss arising problems carefully and without being under time pressure. The slower turn-around of the batch operational mode will usually be compensated for by the smaller number of runs required. For more experienced users, e.g. for post-graduate students, the use of a conversational system seems appropriate, at least during the modeling phase. Full interactive operation -- also during production runs -- is restricted to small scale problems since simulation problems tend to be intensive both concerning execution time and produced output.

It is of utmost importance to carefully select the documentation which is placed at the students' disposal. A carefully selected text containing all major parts of the lecture can be of high value since it frees the students from taking extensive notes. In this way, the students have sufficient time to concentrate on the explanations of the lecturer. Only poor lecturers have to fear a good text! There are several valuable text books on the market which discuss different aspects of modeling and simulation (e.g. [4,10,13]). Unfortunately, many of them are either too theoretically oriented, too large in size, or concentrate too much on some numerical or statistical problems (like numerical integration or random number generation) to be of a great help to the beginner. For this reason, and due to the fact that the

English language in which most of the available text books are written may prevent some of our students from making optimal use of such a text, we decided to use a manuscript of our own.

For the syntactical part (b), there should always be a complete software manual at the students' disposal.

3. CONTENTS OF THE LECTURE

For the lecture, the contents of which we are going to describe in the following, there is a total of about 50 hours available for both theory and exercises. It is, therefore, very important to balance carefully between breadth of topics and depth of elaboration. The catalogue of topics presented in this paper resulted as a reasonable tradeoff based on a twelve years experience, but it is certainly only one of possible alternatives.

3.1 Construction of models

Although the construction of models is certainly one of the most important topics to be discussed, it seems very difficult if not impossible to offer practical exercises in modeling which can be solved by the students within the available time. For this reason, the construction of models is presented through examples during the lecture only, whereas, for the exercises, the students are mostly given prepared mathematical representations of the systems to be simulated in terms of algebraic and differential equations. This can be seen to be a risk as the students will not necessarily realize how difficult and time consuming the steps from a "real" system under investigation through a conceptual model to a formal mathematical representation of the system are -- even for very experienced specialists of modeling. Although there is not sufficient time available to discuss sophisticated techniques for model construction (e.g. parameter estimation techniques) in detail, it seems important to bring those techniques to the students' attention (mainly through examples) and, for the rest, to refer to specialized lectures of comparable length in which those techniques are discussed as central topics of the performance. A valuable discussion of this topic can be found in [4].

Students, particularly interested in model construction techniques, are given the opportunity to write one of their term projects on a topic where they can use these techniques extensively (cf. chapter 5).

3.2 Structuring of continuous models

One of the matters the students should learn as early as possible, is the structuring of models, starting from a raw structure, by stepwise refinement. This is necessary both to provide for lucidity of the program flow and to improve the efficiency of its execution.

The raw structure includes division into model parts for the initialization, for control and computation of the dynamics, and for execution of final computations and eventually optimizations.

For the realization of the fine structure, the students require knowledge of the sorting algorithm in connection with parallel model structures, of the logic flow control in connection with procedural (sequential) model structures, and of the utilization of subprograms.

The concept of hierarchical structures is illustrated by macros. (Although it has been recently shown that macros are not really modular modeling elements [2,5], it is not necessary to present those limitations of macros already in an introductory course. When COSY becomes available, it may, nevertheless, prove useful to introduce the enhanced module concept as well.)

3.3 Structuring of discrete models

The basic modeling element of discrete systems is the event. The event notion is, therefore, the basis of our presentation of discrete models. The event notion can, furthermore, be used to bridge the gap to continuous systems in that it can be shown that the most appropriate way to handle discontinuities in otherwise continuous models (we call these models combined models) is to treat them as discrete events (time events and state events). In this respect, GASP-V provides us with the most appropriate mechanisms for introducing discrete modeling. The understanding of these mechanisms is deepened by exercises in which the students learn to code both purely discrete and combined GASP-V programs. After a very short time, the students are able to master quite complex situations like modeling a drive-in bank with jockeying [12].

Unfortunately, the event notion gets somewhat cumbersome and difficult to survey as soon as the system under investigation grows in size. Then, we require some mechanisms to group events together. Such mechanisms are presented during the lecture (process orientation, activity scanning), but currently no exercises are offered on these concepts since GASP-V does not provide for corresponding facilities.

3.4 Structuring of data

When modeling large scale systems, not only the model structures grow in size, but usually also the amount of data associated with the model. This concerns input data (like function tables), intermediate data (like entries in a waiting queue), and output data (storage and retrieval of state trajectories, histograms, etc.).

In continuous simulation, the data handling concepts required for small scale simulation studies are merely trivial. It is, therefore, not necessary to introduce more advanced concepts (including proper data base management) during an introductory course of simulation.

In discrete simulation, on the other hand, appropriate data structuring mechanisms are vital even for small scale problems. It is particularly important to carefully introduce file handling mechanisms (for the event queue, and for waiting queues).

3.5 Validation of models and verification of simulation results

It is essential that the students learn from the beginning to carefully distinguish between the system they want to analyse and the model representing that system under certain experimental conditions. This distinction is introduced best by presenting several entirely different models for one and the same physical system. As an example, there are several models introduced for a traffic control situation:

a) a macroscopic model, in which the single vehicles are not represented, but only traffic density and traffic flow (entirely continuous model),

b) a microscopic model, in which the single vehicles are represented by entries in a file system, jumping from one decision point to another (intersections, traffic lights, gas stations) in discrete steps (entirely discrete model), and

c) a submicroscopic model, in which each car is represented by a set of differential equations intermixed with discrete events representing the decisions of the driver (combined model).

Having presented these models, it can be discussed which of them to choose for what type of investigation. Validation of a model basically means to determine the class of experiments for which the model can yield meaningful results.

Another way the students can learn to critically assess a given model is by discussing models for ill-defined systems. For one of the practical exercises, the students are given before-hand a brochure of 25 pages length in which the interactive simulation system DARE-ELEVEN [10] is introduced together with the world model by Forrester [7]. After having studied this brochure at home, a group of students may use a PDP-11/34 computer, on which the DARE-ELEVEN system is installed, to play around with the world model. They learn how to interpret the obtained state trajectories correctly, what questions may be answered by the model and which may not.

Verification of simulation results is another delicate matter. The (credulous) students are inclined to believe in the outcome of a simulation study for the simple reason, that the computer prints out results with 14 digits! Although it seems evident that simulation software should, in future, be able to verify the correctness of simulation results with respect to the model automatically, this goal has not yet been reached to a satisfactory extent. It is, therefore, important to rouse the alertness of the students in this respect. This is best done by letting the students analyse the same model by use of different numerical algorithms (e.g. different integration techniques) for comparison of results.

3.6 Combined continuous/discrete system simulation

The relevance of the discrete event notion for the numerical treatment of discontinuous systems has been mentioned above. Beside of these obvious advantages of combined modeling, it seems fruitful to introduce a common terminology for continuous and discrete modeling. There are so many concepts in common to both of these techniques, that it seems stupid to stick to two different terminologies just for historical reasons. Utilization of a combined simulation software for the exercises is, also for this reason, advantageous.

3.7 Integration

Although numerical integration algorithms form certainly the heart of any continuous simulation program, it is not necessarily beneficial for an introductory lecture on simulation to discuss numerical integration techniques in great detail. It is sufficient to give a classification of integration methods (implicit versus explicit techniques, predictor-corrector techniques, one-step versus multi-step methods) showing, by means of examples, which class of algorithms is most appropriate for the treatment of what class of problems. In particular, it is important to show the numerical difficulties inherent in stiff system simulation. Rules of thumb -- basing on the most influential factors like: required accuracy, smoothness of system behaviour, and stiffness of the problem -- are to be given to enable the students to choose an appropriate algorithm off-hand.

3.8 Distributed system simulation

The majority of parabolic and hyperbolic partial differential equations found in models for technical systems can be transformed into sets of ordinary differential equations by the method-of-lines approach. This is, therefore, the only technique which we introduce in our lecture of simulation. Some simple difference and error formulae for the spatial discretization in cartesian and polar coordinates are introduced, and the interaction between grid-width in space and step-size in time is explained. A simple example of a diffusion problem can be solved by the students during an exercise.

3.9 Stochastic models and statistical analysis

As a preparation to the modeling of noisy continuous systems, and, in particular, as a basis for the modeling of discrete systems (which are practically all of a stochastic nature), it is useful -- although not indispensable -- that the students understand how (pseudo) random numbers are generated on the computer, and how various probability distributions can be obtained. Of particular importance for continuous systems is the Gaussian distribution (white noise), whereas discrete systems frequently make use of exponential and Poisson distributions.

Of special interest to continuous simulation is the modeling of frequency dependent noise processes. In this context, it is essential to show the interaction between the sampling interval for the involved random number generator and the spectre of the resulting noise distribution in the frequency domain.

In discrete simulation, the evaluation of non-terminating runs is of particular interest. In this context, questions are to be answered like:

- Starting from any set of initial conditions, how long does it take until the steady state behaviour has been reached?
- What is the minimum run-length required to achieve a given confidence interval?

Both variance estimation and variance reduction techniques are presented in this context.

3.10 Boundary value problems

Time limitations inhibit a very detailed discussion of boundary value problems. Only problems with one single unknown initial condition are discussed by presenting some of the appropriate methods (shooting method with binary interval reduction and with regula falsi). Inverse Hermite' interpolation is discussed in another context (location of discontinuities). One complete example program with two unknown initial conditions and explanation of how it works is made available for home reading.

3.11 Identification, optimization

Optimization and identification of technical systems lead in many cases to the problem of minimizing (or maximizing) a nonlinear performance index. The treatment of such problems is presented in a lecture of its own, and cannot be fully discussed during our lecture on simulation techniques. In this context, only some basic principles are touched, including discussion of a unidimensional search technique (e.g. interpolation or golden section), and of the idea behind gradient methods (e.g. by presenting the method of steepest descent).

3.12 Interpolation

All continuous simulation systems available today offer mechanisms for interpolation of measured data which are stored in a tabular form. Interpolation routines in one and two dimensions use mostly Lagrange polynomials for this task. Since these polynomials are discontinuous at the boundaries, it seems essential to discuss also spline interpolation formulae (at least in one dimension), although most of the available simulation software systems do not offer such algorithms yet.

Discussion of these techniques can be useful for a diversity of other problems too, including the computation of spatial derivatives for distributed systems simulation, location of discontinuities, unidimensional search in optimization, and solution of boundary value problems.

3.13 Special-purpose simulation techniques

In case there is sufficient time available during a simulation course, the horizon with respect to presently available simulation tools and techniques may be broadened significantly by giving an overview over the special-purpose software packages used in various disciplines of engineering, informatics, and business administration [11,14]. Both the numerical algorithms and the modeling techniques used in such programs often differ significantly from those described in the previous sections of this chapter.

Table 1 compiles some of the more important characteristics of such software systems. This table is, however, not meant to be exhaustive.

Beside of this very brief introduction, there are also other lectures being offered which concentrate on computer-aided analysis and synthesis of electrical networks. These lectures are not further discussed in this paper.

4. OTHER TOPICS OF INTEREST

4.1 Enhanced discussion of distributed system simulation

In the lecture, presented in chapter 3, we did not give much attention to numercial algorithms. This approach seems justified by the fact that, for many simulation problems, the selection of the most appropriate numerical algorithms is not crucial. Very often, one may find that the same problem may, for instance, be simulated by use of many different integration algorithms with comparable accuracy and only slightly varying execution time. This statement does, unfortunately, not hold for the numerical treatment of partial differential equations. In distributed system simulation, there is a strong link between the problem to be solved and

SIMULATION AREAS	CHARACTERISTICS, PROBLEMS, METHODS
Electrical Networks a) analog	Systems of nonlinear equations (Newton-Raphson), roots of polynomials, Laplace transform, Fourier analysis, Monte-Carlo analysis, sensitivities, noise calculations, sparse matrix techniques
b) digital	Logic and truth tables, delays, macro-modules
c) power	Sparse matrix techniques, L-U-decomposition, short and open circuits, transient peaks
Communication Networks	Queuing theory, circuit and packet switching, polling, protocol and program verification
Computer Systems	Capacities, loads, access times, response times
Structures in Civil and Mechan. Engg.	Finite elements, deformation analysis, eigenvalues, eigenvectors (Householder-Sturm)
Corporate Models	Multidimensional data bases, sorting, sensitivities, regression, forecasts
Econometric Models	Time series analysis, parameter estimation, systems of nonlinear equations (Gauss-Seidel, Van der Giessen), extended statistics
Management Games	Production, market and financial planning, optimization (Simplex), forecasts

Table 1 : Characteristics of special-purpose simulation software systems

the algorithm to be used. An unlucky choice of numerical algorithms may easily lead to execution times which are by orders of magnitude larger as compared to the optimal combination. Appropriate selection, on the other hand, requires a very profound understanding of the numerical mathematics involved which cannot be expected from undergraduate students other than from the mathematics department. For this reason, an enhanced discussion of numerical techniques for the simulation of distributed systems ought to be postponed to an advanced course offered to postgraduate students only.

4.2 Real-time simulation

A topic of particular interest to control engineers is the synchronization of the simulation clock with real time. Applications include training devices (like flight simulators), and digital control devices (like observers, adaptive controllers). Here again, the numerical techniques involved in the solution become crucial. It is, for instance, often infeasible to use step-size controlled integration algorithms for this task. The programmer is then fully responsible by himself for the accuracy achieved and for the numerical stablilty of his algorithm. Although an exhaustive discussion of this challenging topic should be postponed to a lecture being attended only by control engineers, it would be nice to show the common basis of this topic and simulation during the introductory lecture discussed above. This could be done by presenting a film on a flight simulator and/or by giving a demonstration of a digital control system. So far, neither of them has been realized.

4.3 Large scale system modeling and simulation

As it has already been mentioned, new problems arise when applying both modeling and simulation techniques to large scale systems. Currently, we see no proper way how to include such a discussion into a lecture on simulation techniques. It seems much more promising to let small groups of students explore these difficulties during term projects in which they are faced with a particular large scale system to be modeled (like the water supply system of a city).

5. TERM PROJECTS

All our students have to carry out one or several term projects (depending on the department in which they study). They are free to select a topic of their choice. Each year, about 8 - 10 students (that is: approximately 20% of the attenders of the lecture) decide to write one of their term projects in simulation.

Students interested in modeling may write a project on some complex modeling task. Examples of such projects which have been studied so far include among many others: modeling of electro/mechanical systems (e.g. an electrical discharge machinary for die sinking work), and modeling of thermal systems (e.g. a solar energy heating system).

Students being more interested in simulation techniques, may accomplish projects on simulation software design. A good deal of the work which went into the development of both GASP-V and COSY was realized by students during their term projects. Topics included: language design (formal definition of the COSY simulation language), compiler construction (translator from CSMP to COSY, development of the Jacobian generator for COSY), and numerical techniques (automated selection of integration algorithm).

6. CONCLUSION

At ETH Zurich, simulation techniques have become integral parts of the education of electrical engineers, mechanical engineers, mathematicians, and physisists. By carefully balancing between theoretically oriented topics, associated applied mathematics, presentation of simulation software, and solution of practically oriented exercises, both interest and initiative of the students can be permanently maintained. Even by means of a one-term lecture, it is possible to achieve such a high standard of knowledge that, afterwards, the students are able to tackle complex modeling and simulation tasks independently, e.g. during a term project.

REFERENCES

[1] Francois E. Cellier and Andre E. Blitz, GASP-V: a universal simulation package, Proceedings of the 8th AICA Congress on Simulation of Systems, North-Holland Publishing Company, (L.Dekker, ed.), 1976, 391-402.

[2] Francois E. Cellier and A. P. Bongulielmi, The COSY simulation language, Proceedings of the 9th IMACS Congress on Simulation of Systems, North-Holland Publishing Company, (L.Dekker, G.Savastano, G.C.Vansteenkiste, eds.), 1979, 271-281.

[3] Francois E. Cellier and A. A. B. Pritsker, Teaching continuous simulation using GASP, Simulation, vol. 34, no. 4, April 1980, 137-139.

[4] Yaohan Chu, Digital simulation of continuous systems, McGraw-Hill, New York, 1969.

[5] Hilding Elmqvist, A structured model language for large continuous systems, Proceedings of the Summer Computer Simulation Conference, Simulation Councils Inc., 1979, 8-14.

[6] Jay W. Forrester, Industrial dynamics, The M.I.T. Press, Cambridge, 1961.

[7] Jay W. Forrester, World dynamics, Wright-Allen Press Inc., Cambridge, 1971.

[8] Walter Gander, Juergen Harms, Peter Laeuchli, Francois Nicolet, Jakob S. Vogel and Carl A. Zehnder, Informatik fuer Ingenieure, Springer Verlag, Heidelberg, 1980.

[9] Jack P. C. Kleijnen, Experimentation with models: Statistical Design and analysis techniques, in: Progress in Modelling and Simulation, Academic Press, (F.E.Cellier, ed.), 1981.

[10] Granino A. Korn and John V. Wait, Digital continuous system simulation, Prentice-Hall, Englewood Cliffs, 1978.

[11] Hans Marti and Jakob S. Vogel, The value of a top management game in the education of production engineers, Proceedings of the WCCE 81: 3rd World Conference on Computers in Education, North-Holland Publishing Company, 1981.

[12] A. Alan B. Pritsker, The GASP-IV simulation language, John Wiley, New York, 1974.

[13] A. Alan B. Pritsker and Claude D. Pegden, Introduction to simulation and SLAM, Halsted Press, John Wiley, New York, Systems Publishing Corporation, Lafayette, 1979.

[14] André R. Probst and Jakob S. Vogel, Corporate modeling: An integral part of business informatics education, Proceedings of the WCCE 81: 3rd World Conference on Computers in Education, North-Holland Publishing Company, 1981.

[15] Robert G. Sargent, Verification and validation of simulation models, in: Progress in Modelling and Simulation, Academic Press, (F.E.Cellier, ed.), 1981.

[16] Frank H. Speckhart and Walter L. Green, A guide to using CSMP, Prentice-Hall, Englewood Cliffs, 1976.

[17] Continuous System Modeling Program III (CSMP III), Program Reference Manual, IBM Form nr. SH19-7001.