MANAGING SIMULATION PROJECTS

Dr. Joe K. Clema
Pratt & Whitney Aircraft Group
Government Products Division
P.O. Box 2691
West Palm Beach, Florida 33402

ABSTRACT: A brief look at the past, an examination of the present, and a forecast for where we are going in the fields of modeling and simulation are presented. Some of our major problems and difficulties are identified and several recommendations for improving the state of our art are discussed. The management techniques discussed provide for a powerful, systematic and repeatable approach toward extending the application of our field to large-scale systems of any kind.

1. THE PAST

During the first third of a century that the digital computer has existed we have foundered in an abyss of difficulties. A lack of understanding of computer modeling and simulation techniques coupled with little understanding of the software life cycle have led to monumental and costly failures. These problems, coupled with a management that had little (or no) digital computer background was frequently a deadly combination. Due to the lack of experience of the participants it was often impossible in the post-audit to identify the factor or factors which led to the failure. Even today, we still pay the penalty of working in a field which is more akin to art (witchcraft ?) than a proven scientific discipline with a methodology guaranteeing a high probability of success. Thus, we have a small number of artisans with proven track records in consulting, industry, academia, and government who are generally successful, and a second group of simulation practitioners, large in number, who perhaps have not acquired the necessary skills and experience to achieve the success desired by the customer.

Against this background has evolved a body of knowledge which is rapidly bringing the trilogy of fields of management, software engineering, and simulation into perspective. Some of the problems (e.g., analysis and design approaches, coding and debugging techniques, simulation languages, mathematical and statistical methods) are becoming reasonably well-defined and relatively stable. On the other hand, we still face some rather irksome problems that will likely not be solved in less than ten to twenty years (cost and schedule estimating, testing, verification and validation methods, effective management and control techniques, establishing and measuring levels of fidelity, and effective communication and visibility means for describing what we do and how we do it to our team members, customers, and upper management.)

2. THE PRESENT

It is my opinion that we are beginning to emerge from the day of the craftsman and artisan. The transition has begun, but the evolution will be very slow. More and more managers are evolving from the software ranks, and though software-experienced personnel do not necessarily equate with good management, they will at least more fully appreciate software problems, be able to communicate the difficulties, and bring resources to bear on the real simulation obstacles. Today, as in the past, we see numerous papers, articles and books on various simulation applications. These writings are generally concerned with a highly specialized area, e.g., energy, health care, avionics, etc. Even when promoting a generalized approach, technique, or tool, the advocated method does not seem to reach a large audience because of lack of interest in the specialized application. We need to gather and collect these general methods in a central repository and make the information and tools available.

TH0079-4/80/0000-0235$00.75 © 1980 IEEE.
Simulation with Discrete Models: A State-of-the-Art View
T.I. Ören, C.M. Shub, P.F. Roth (eds.)

Today we are deluged with languages. Far more than we can ever use effectively, and unfortunately, many have little to recommend their use. It appears that in the mid-eighties even FORTRAN will be retrofitted with an event generation capability turning it into a simulation language of sorts. With the Department of Defense developing its own new language (ADA - to be released in the mid-eighties), which possesses both real and non-real-time capabilities modeled after SIMULA, the simulation practitioner is hard pressed to select a language from all the available languages. Many feel that in general, the specific language makes little difference in the end. There is also little doubt that in certain specific instances languages like GPSS, SIMSCRIPT, GASP, ACSL, etc., do make a difference in the ease and rapidity with which a particular project can be implemented. However, one wonders in the long-run, if a general-purpose procedural language incorporating something like GASP or SLAM might not ultimately be the best way to go because of the ease with which FORTRAN programmers can be found. Experienced GPSS and SIMSCRIPT users on the other hand have significant advantages in simulating certain types of applications and a person experienced in these languages can be a very valuable asset. With the difficulty of choosing a language today one wonders what will happen with the next update of FORTRAN and the release of ADA. I think we all agree that we have too many languages, and that it would be beneficial to examine them in depth and settle on one or two as the best of the lot.

On the management side we need to establish a step-by-step repeatable methodology to provide a systematic approach with a high degree of success. Table 1 lists a set of proven practices that have been shown to dramatically improve productivity and produce a more reliable product.

3. THE FUTURE

The repeatable simulation development process is beginning to emerge. Although, it is not finalized, the framework from which it will emerge is portrayed in Figure 1.

The simple fact is that digital simulations (and particularly large, multi-disciplinary simulation systems) are among the most complex intellectual activities ever undertaken by man. They demand firm management methodologies and strong discipline. The future holds the demand for larger, more complex, more sophisticated systems to be developed. These systems will be in the fields of social science, energy, economics, distributed systems, and adaptive programming, as well as the more conventional fields such as aerospace and computer system simulation. I believe that we will see the field of artificial intelligence make its greatest advances simulating human decision making with heuristic methods. On the horizon are new simulation languages, continued language enhancements, automated tools, powerful interactive executives, more real-time simulations and distributed simulations to provide rapid turnaround of large-scale multidisciplinary systems runs.

To attack these complex systems in an effective manner requires adherence to the modern software practices promoted by the software engineering advocates. Thus, a phased approach based on the software life-cycle (Figure 1) is imperative. The 40-20-40 rule should be fully understood. This rule states that 35 to 45 percent of the resources should be expended on analysis and design; 15 to 25 percent of the resources will be expended on coding and debugging; and 40 to 50% of the resources will be expended on testing.

The customers, analysts and designers must emphasize the development of easily understood and measureable requirements, plus an unambiguous design. Graphic aids including block diagrams should be used to facilitate development team understanding, as well as portraying the software approaches to the customer. Test procedures should be established during analysis and design so that the validity of the simulation can be determined. It should be easy to show the customer that the requirements are met and prove to him that the project has accomplished the goals.

Today, work should generally be of an interactive nature. Interactive executives have been developed which facilitate the setup, operation, and maintenance of a simulation. Several papers (see References on simulation tools) have been written describing these tools, and the trend is to include such capabilities as a part of a simulation language.

We must actively work in many areas to more fully promote and utilize our powerful tool to its full potential. We must define and understand the simulation life-cycle in detail. It must be recognized that the management techniques vary from life-cycle phase to life-cycle phase. We must limit the number of languages we use to a select few so that our capabilities become portable and useful to others. We need to significantly improve the whole field of testing, verification, and validation; and finally we need a system for making valid results credible to the customer who may not understand simulation.

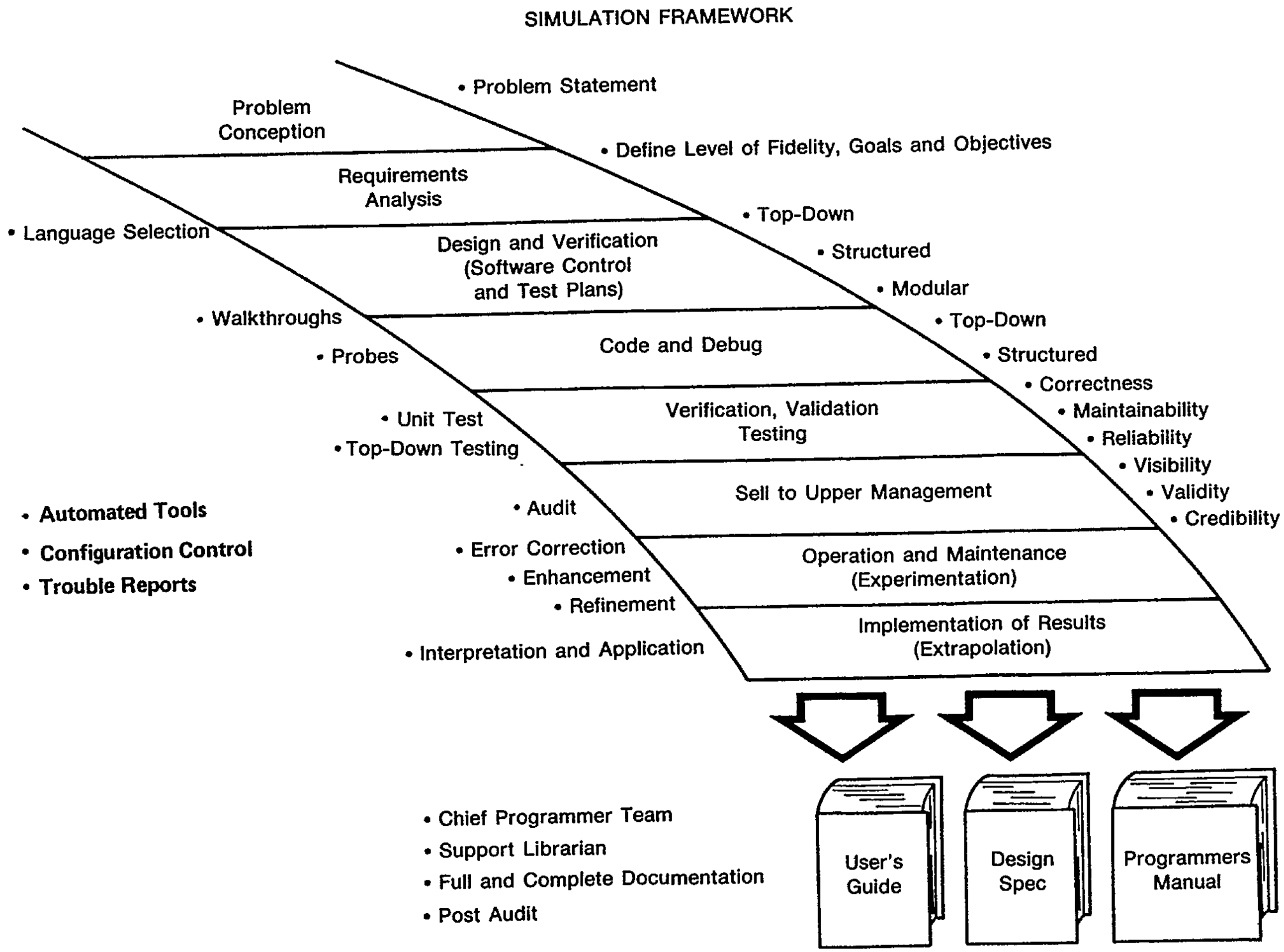
SIMULATION FRAMEWORK
Problem Conception
Requirements Analysis
Design and Verification (Software Control and Test Plans)
Code and Debug
Verification, Validation Testing
Sell to Upper Management
Operation and Maintenance (Experimentation)
Implementation of Results (Extrapolation)
• Problem Statement
• Define Level of Fidelity, Goals and Objectives
• Top-Down
• Structured
• Modular
• Top-Down
• Structured
• Correctness
• Maintainability
• Reliability
• Visibility
• Validity
• Credibility
• Language Selection
• Walkthroughs
• Probes
• Unit Test
• Top-Down Testing
• Audit
• Error Correction
• Enhancement
• Refinement
• Interpretation and Application
• Automated Tools
• Configuration Control
• Trouble Reports
• Chief Programmer Team
• Support Librarian
• Full and Complete Documentation
• Post Audit
User's Guide
Design Spec
Programmers Manual

Figure 1.

There exist several major questions and problems which we still face and cannot hope to rectify easily in the near term. These are the following:

1. How do we determine the degree of fidelity for a specific real-world problem?
2. How do we know when we should use simulation techniques and when to use analytical tools?
3. How do we convince upper management that our simulation results are meaningful after we have convinced ourselves?
4. What is a unified theory for modeling and simulation?
5. Do we need more than one computer language for modeling and simulation?

4. SUMMARY AND CONCLUSIONS

The introduction of a very structured and repeatable framework for developing simulation models and simulation software is a major step forward. By subjecting the developmental roadblocks to scrutiny and defining a step-by-step methodology for broaching each difficulty, a powerful approach will be provided for extending the application of our field to any large-scale system. The capabilities and potential of our remarkable field have only been scratched; only through cost effective management techniques shall we achieve all the promises that the modeling and simulation techniques can provide in a way that satisfies the customers and users.

TABLE 1

MODERN SOFTWARE PRACTICES

- TOP DOWN AND STRUCTURED METHODS
- STRONG MANAGEMENT INVOLVEMENT
- WRITTEN ASSIGNMENTS TO IMPLEMENTORS
- EMPHASIS ON VISIBILITY
  - Documentation First - (and Detailed)
  - Software Architecture Drawings
  - Interface Diagrams
  - Data Dictionary
- FORMAL REVIEWS
- WALKTHROUGHS (EGOLESS PROGRAMMING)
- TEST PROBES
- FORMAL TESTS (UNIT, INTEGRATION, SYSTEM)
- CONFIGURATION MANAGEMENT AND CONTROL
  - Review Team
  - Master Library
  - Requirements
  - Design
- CHIEF PROGRAMMER TEAM

REFERENCES

Allen, B. and Joe Clema, The AVSAIL Simulator Complex, Ninth Annual Simulation Symposium, Tampa, Florida, March, 1976, Record of Proceedings, Vol. 6, pp. 123-133.

Allen, B. and Joe Clema, Man-Machine Interfaces for Large-Scale Simulations, 1974 ASSC, Automatic Support Systems Symposium for Advanced Maintainability, San Diego, Calif., 1974, pp. 221-227

Aron, Joel D., The Program Development Process, Addison-Wesley Publishing Co., Reading, Mass., Menlo Park, Calif., 1974, 264 pp.

Atwood, Jerry W., The Systems Analyst, Hayden Book Co., Inc., Rochelle Park, N.J., 1977, 225 pp.

Buxton, J.M. Et.AL. (Eds.), Software Engineering Concepts and Techniques, Petrocelli, New York, 1975, 306 pp.

Chase, Wilton P., "Management of System Engineering", John Wiley and Sons, New York, 1974, 228 pp.

Chu, Yaohan, "Digital Simulation of Continuous Systems", McGraw-Hill, New York, 1969, 423 pp.

Churchman, C. West, The Systems Approach, Dell Publishing Co., Inc., New York, 1968, 243 pp.

Clema, Joe and Allen, Byron, "Independent Verification/Validation Support Software", Proceedings of the IEEE 1976 National Aerospace and Electronics Conference, Dayton, Ohio, 1976, pp. 276-281.

Colella, A.M., O'Sullivan, M.J., and Carlino, D.J., Systems Simulation, Lexington Books, Lexington, Mass., 1974, 292 pp.

Collacott, Ralph (Ed.) "Simulators International Guide", Addison-Wesley Publishing Co., Reading, Massachusetts, 1973, 602 pp.

Dahl, O.J., Dijkstra, E.W., and Hoare, C.A.R., Structured Programming, Academic Press, New York, 1972, 220 pp.

Deutsch, Ralph, System Analysis Techniques, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1969, 472 pp.

Dijkstra, E.W., "The Humble Programmer", Communications of the ACM, October 1972, pp. 859-866.

Emshoff, James R. and Sisson, Roger L., "Design and Use of Computer Simulation Models", Macmillan Co., New York, 1970, 302 pp.

Fosdick, Lloyd and Osterweil, Leon J., "Data Flow Analysis in Software Reliability", Computing Surveys, Vol. 8, No. 3, September 1976, pp. 305-330.

Franta, W.R., "The Process View of Simulation", North-Holland, New York, 1977, 244 pp.

Hetzel, William C. (Ed.), "Program Test Methods", Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1973, 363 pp.

Horning, James J. and Wortman, David B., "Software Hut: A Computer Program Engineering Project in the Form of a Game", IEEE Transactions on Software Engineering, Vol., SE-3, No. 4, July 1977, pp. 325-330.

Hough, Louis, "Modern Research for Administrative Decisions", Prentice-Hall, Inc., Englewood Cliffs, N.J., 1970, 609 pp.

Ivie, Evan L., "The Programmer's Workbench - A machine For Software Development", Communications of the ACM, Oct. 1977, V.20, N.10, pp. 746-753.

Karplus, Walter J., "The Spectrum of Mathematical Modeling and Systems Simulation", Mathematics and Computers in Simulation, Vol. XIX, Number 1, March 1977, pp. 3-10.

Katzan, Harry, Systems Design and Documentation, Van Nostrand Reinhold Company, New York, 1976, 157 pp.

King, James C., "Limitations of Language Design for Reliable Software", Software Engineering Notes, Vol. 2, No. 3, April 1977, pp. 11-21.

Kiviat, P.F., Villanueva, R., and Markowitz, H.M., "The SIMSCRIPT II Programming Language, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1968, 386 pp.

Krosner, Stephen P., "Test Data Generation Tutorial", Proceedings of the ACM Annual Conference, August 1972, Boston, Mass., 1156 pp. 1146-1147 pp.

Lucas, Henry C., "The Analysis, Design, and Implementation of Information Systems", McGraw Hill Book Co., 1976, 255 pp.

Maisel, Herbert and Gnugnoli, Giuliano, "Simulation of Discrete Stochastic Systems", Science Research Associates, Inc., Chicago, 1972, 465 pp.

Martin, Francis F., Computer Modeling and Simulation, John Wiley and Sons, New York, 1968, 331 pp.

Meadow, Charles T., The Analysis of Information Systems, (2nd Ed.), Melville Publishing Co., Los Angeles, Calif., 1973, 420 pp.

Meier, Robert C., Nerwell, William T., and Pazer, Harold L., Simulation in Business and Economics", Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1969, 369 pp.

Mihram, G. Arthur, "Simulation", Academic Press, New York, 1972, 526 pp.

Mills, Harlan D., "How to Write Correct Programs and Know It", Proceedings of the International Conference on Reliable Software, IEEE Catalog No. 75CH09407CSR, April 1975, pp. 363-370.

Myers, Glenford, Reliable Software Through Composite Design, Petrocelli, New York, 1975, 159 pp.

Naylor, Thomas H., Computer Simulation Experiments with Models of Economic Systems, John Wiley and Sons, Inc., New York, 1971, 502 pp.

Optner, Stanford L., "Systems Analysis for Business and Industrial Problem Solving", Prentice-Hall, Inc., Englewood Cliffs, N.J., 1965, 116 pp.

Oren, Tuncer I., "Software for Simulation of Combined Continuous and Discrete Systems: A State of the Art Review", Simulation, February, 1977, pp. 33-45.

Prasad, S. Benjamin, Modern Industrial Management, Chandler Publishing Co., San Francisco, Calif., 1967, 460 pp.

Pritsker, A. Alan B., "The GASP IV Simulation Language", John Wiley and Sons, New York, 451 pp.

Ralston, Anthony and Meek, C.L., (Eds.), Encyclopedia of Computer Science, Petrocelli, New York, 1976, 1523 pp.

Rosko, Joseph S., Digital Simulation of Physical Systems, Addison-Wesley Publishing Co., Reading, Mass., 1972, 427 pp.

Scallon, G.M. and Grupé, J.A., "Functionally-Oriented System Simulation for Computer Aided Design of Software/Hardware Systems", Proceedings of the 9th Annual Simulation Symposium, 1976, pp. 225-239.

Scarpino, Frank and Joe Clema, A General Purpose Tool for Interactive Simulations, Winter Simulation Conference, Gaithersburg, Md., 1976, pp. 475-484.

Schmidt, J.W., and Taylor, R.E., "Simulation and Analysis", Richard D. Irwin, Inc., Homewood, Ill., 1970, 644 pp.

Schriber, Thomas J., "Simulation Using GPSS", John Wiley and Sons, New York, 1974, 533 pp.

Shapiro, George and Rogers, Milton, "Prospects for Simulation and Simulators of Dynamic Systems", Spartan Books, New York, 1967, 304 pp.

Stephenson, Robert E., "Computer Simulation for Engineers", Harcourt Brace Jovanovich, Inc., New York, 1971, 504 pp.

Sutherland, John W., SYSTEMS Analysis, Administration and Architecture, Van Nostrand Reinhold Co., New York, 1975, 339 pp.

Wienwurm, George F., "On the Management of Computer Programming", Auerbach Publishers, Inc., New York,

Wyman, Forrest Paul, "Simulation Modeling: A Guide to Using SIMSCRIPT", John Wiley and Sons, Inc., New York, 1970, 211 pp.

Wymore, A. Wayne, Systems Engineering for Interdisciplinary Teams, John Wiley and Sons, New York, 1976, 431 pp.

Yourdon, Edward, Techniques of Program Structure and Design, Prentice-Hall Inc., Englewood Cliffs, N.J., 1975, 364 pp.

Yourdon, Edware, How to Manage Structured Programming, Yourdon, Inc., New York, 1976, 188 pp.

Yourdon, Ed and Constantine, Larry, Structured Design, Yourdon, Inc., New York, 1975, 599 pp.