Proceedings of the 1986 Winter Simulation Conference
J. Wilson, J. Henriksen, S. Roberts (eds.)

JOINING EXISTING SIMULATION PROGRAMS

ROBERT G. SARGENT
Department of Industrial Engineering
and Operations Research
Syracuse University
Syracuse, NY 13244

ABSTRACT

The main purpose of this paper is to make people (developers, users, managers, researchers) aware that there are numerous factors and issues that affect joining separate simulation models and programs. The primary content of the paper is the identification of the different factors and issues that affect joining existing simulation programs. Two fundamental ways of joining existing simulation models are identified and discussed: where none of the simulation models interact with any of the others during simulation runs and where some or all of the simulation models do interact with others during simulation runs. The concept of "amount of reusability" of simulation models and programs is introduced. Also, the need for research is discussed.

1. INTRODUCTION

As more simulation models are developed and the investments in them increase, managers and others are starting to ask that these models be joined to solve problems. As many simulation people know, there are usually several reasons why this cannot be done or done easily. This usually comes as a surprise to managers and is frequently difficult for them to accept.

This desire to join existing simulation models is not limited to any specific application domain. I have recently been exposed to the desire to have existing simulation models joined of military systems, of production and manufacturing systems, and of computer systems. What is desired to be joined can also be quite varied. Some actual examples are the joining of two existing simulation models (programs) of specific systems, the desire to join a special purpose simulation language with a simulation model of a specific system, and the desire to join two special purpose simulation languages that model the same type of system.

This paper has three primary objectives. The first is to make people (users, developers, managers, researchers) knowledgeable and aware of this subject and its problems. The second is to take the (initial) step of identifying factors and issues that affect the joining of existing simulation programs. The third objective is to encourage research in this area.

This paper is organized into six sections. Section 2 describes different ways that existing simulation programs can be joined. Section 3 identifies factors and issues that affect the joining of existing simulation programs. Section 4 discusses the concept of "amount of reusability" of existing simulation programs. Section 5 discusses conducting research on this subject, and Section 6 is the summary.

2. WAYS OF JOINING EXISTING SIMULATION PROGRAMS

The first issue which should be addressed when there is a desire or request to have existing simulation models and programs joined is "what are the objectives or goals" to be accomplished; then the issue of whether or not it is necessary to join the models and programs in order to accomplish the objectives or goals should be addressed. If it is determined that the models and/or programs should be joined, the simplest way to do so should be found. The questions "is it technically feasible", "is it practical", and "is it cost effective" must be addressed.

There are two fundamental ways of joining existing simulation models. The first fundamental way is when the simulation models can run separately. Three different ways that this may occur are (1) the outputs of some simulation models are inputs to other simulation models (Some model outputs may require transformations or analysis prior to being used as inputs into other models.), (2) experimental results on some models determine the experiments to be performed on other models, and (3) a combination of ways (1) and (2). It is usually desirable to have the models joined in this fundamental way, if possible, because it is usually significantly simpler than the following fundamental way.

The second fundamental way of joining existing simulation programs is where some or all of the simulation models are to interact with each other during a simulation run. This is usually difficult (and frequently impossible) to accomplish. Some major factors and issues that give different ways of joining simulation models and programs are (1) will the models be interfaced, integrated, or a combination of both, (2) will the models use the same or different implementation languages, (3) will the programs use the same or different computers, (4) will the programs use the same or different software systems, and (5) where is simulation control to be contained (e.g., in a specific model or in a new program). The factors and issues affecting the joining of simulation programs in this fundamental way are presented in more detail in the next section.

3. FACTORS AND ISSUES

There are numerous factors that affect what is required to join existing simulation models and programs. In this section we identify those factors and the issues they generate when it is desired that some or all of the models interact during a simulation run. (Only implicitly will the factors and issues that affect the joining of simulation programs that models run separately be identified as this is, in comparison, relatively straightforward.) These factors and issues determine whether it is technically feasible, whether it is practical, and whether it is cost effective.

The factors which have been identified have been divided into five sets: Models, Computer Systems, Simulation Software Systems, Simulation, and Documentation. Each of the five sets of factors is discussed below in separate subsections with the issues they generate. The discussion is kept brief as it is not the purpose of this paper to discuss these factors and issues in detail. Also, research is needed on most of them, as is discussed in a later section.

3.1 Models

The model factors that affect joining existing simulation models are given in Table 1. A fundamental issue is how are the interacting models to relate to each other. Has some form of model structure been used or can one be (easily) developed that relates the existing models together. Examples of structure are (i) models and submodels and (ii) different levels of functional specification (e.g., using the structural analysis of DeMarco). If structure exists, have the interfaces and data flows been defined. If a (total) model structure exists, the issue of how the models relate can probably be answered. Then it can be determined if the models can be interfaced or integrated easily, not easily, or not at all. If no structure exists, or if one cannot be easily developed, then joining the existing simulation models together probably will be difficult or impossible.

If the models can be related structurally, the next issue to be addressed is what type of world views have been used in the simulation models. Are they the same or different. If different, can they interact together. Associated with the world views is whether or not the simulation models have been developed using specific simulation languages such that the world views are language dependent, e.g., using GPSS. Thus, not only must the world views be compatible but if they are language dependent, compatibility must include that factor as well.

Two other model factors are the types of the models and the fidelities of the models. Are the models causual or statistical. Can the existing simulation models interact based upon the model structure, world view, and the type of model. Are the levels of the model fidelities such that it makes sense that they interact based upon the objectives or goals of the study. The entity-attribute factor must be addressed if the models have any common attributes or entities. If any common entities exist, it must be determined if they have the same attributes, which model will control them, and are the data accuracies the same. Also, it must be decided how to handle, if any exist, attributes in one model that are entitites in another.

TABLE 1: MODEL FACTORS

- STRUCTURE
  - DECOMPOSITION
  - INTERFACES
  - DATA FLOWS
- WORLD VIEWS
  - TYPE: PROCESS, EVENT, ACTIVITY, DIFF. EQUATIONS
  - LANGUAGE DEPENDENT/INDEPENDENT
- TYPE
  - CAUSUAL
  - STATISTICAL
- FIDELITY
  - LEVEL (LOW-HIGH)
  - ENTITY-ATTRIBUTE RELATIONSHIPS

3.2. Computer System

The computer system factors are listed in Table 2. An important factor is whether the existing simulation programs run on the same computer system. If they do and are to remain on this (host) computer system, the computer system issues to be addressed are much simpler. If they do not, a fundamental issue is whether they are to run on the same (host) computer system. If the simulation programs are all to run on the same host computer system, it must be determined if it has sufficient speed to give an adequate response (turnaround) time for the simulation experiments, and sufficient storage (memory) space to handle the simulation programs and associated software. If the existing simulation programs do not all currently run on the same (host) computer and are to do so, it must be determined that they can. Factors that must be considered are the languages of the simulation programs and associated software, word size of the computers, any special hardware and software features of the computers being used, and if the host computer operating system will allow the simulation programs to interact.

TABLE 2: COMPUTER SYSTEM FACTORS

- SIMULATION PROGRAMS
- SIMULATION SOFTWARE SYSTEMS
- SINGLE PROCESSOR
  - SPEED
  - STORAGE
  - SOFTWARE
    - OPERATING SYSTEM
    - LANGUAGES
  - WORD SIZE
  - SPECIAL FEATURES
- MULTIPLE PROCESSORS
  - COMMUNICATIONS
  - CONTROL
  - TIME FLOW CONTROL OF MODELS
  - SPECIAL FEATURES

If instead of using a single host computer, the simulation programs are to run on their current computers, there is a different set of factors and issues. Two issues are how is communication to occur among the computers and what effect, if any, do communication time delays have. Another issue is which computer, if appropriate, is to be the controlling one. A critical issue is how are the models to move through time (e.g., using a Time Flow Mechanism) and be kept synchronized (a possible difficult problem) when they are on different processors (computers). Also, if any special computer features are used, it must be determined that connecting the computers does not cause interference. If some simulation programs and associated software are to be moved from other computers to one of the multiple host computers, the issues in the previous paragraph must also be addressed.

3.3. Simulation Software System

A simulation software system is increasingly being used in simulation today. Since it may affect joining existing simulation models and programs, e.g., it may contain the experimental frame or database, I have chosen to include it in this paper. The simulation software system factors are listed in Table 3. If a simulation model does not have a (simulation) software system associated with it, the appropriate factors presented here should be considered with the simulation model factors presented in the next subsection. The fundamental issue here is whether the simulation software system(s) will operate on the host computer(s) and be able to handle the simulation models that are to interact. The number of factors (listed in Table 3) which affect this is large which results in a very large number of combinations. If only two or three models are to be joined, the complexity is much less than if several models are to be joined. Critical issues are (i) how is the "total model" to be setup, (ii) how are the experiments to be designed, (iii) how are the models to be run, and (iv) how are the data to be collected, analyzed, and reported.

TABLE 3: SIMULATION SOFTWARE SYSTEM FACTORS

- MODEL
- COMPUTER SYSTEM DEPENDENT/INDEPENDENT
- SIMULATION
- LANGUAGE
- DATA BASE MANAGEMENT SYSTEM
  - TYPE OF DATA BASE
  - DATA STRUCTURE
- CAPABILITIES
  - MODEL SETUP
  - SIMULATION RUN CONTROL
  - DEBUGGING CAPABILITIES
  - DESIGN OF EXPERIMENTS
  - DATA ANALYSIS
  - ANIMATION
  - REPORT GENERATOR
  - SPECIAL FEATURES

3.4. Simulation

The simulation factors that affect joining existing simulation models and programs are listed in Table 4. The issues regarding these factors have all been discussed in the previous three subsections. This occurred because the sets of factors are not independent of each other. The factors regarding time flow are presented in more detail here. It is critically important that the movement of the various simulation models through time be synchronized correctly. This may be difficult to accomplish.

TABLE 4: SIMULATION FACTORS

- MODEL
- COMPUTER SYSTEM
- SIMULATION SOFTWARE SYSTEM
- MODEL IMPLEMENTATION LANGUAGE
  - HIGH LEVEL GENERAL PURPOSE LANGUAGE
  - GENERAL PURPOSE SIMULATION LANGUAGE
  - SPECIAL PURPOSE SIMULATION LANGUAGE
  - COMPUTER DEPENDENT/INDEPENDENT
- TYPE: DISCRETE, CONTINUOUS, REAL TIME
- TIME FLOW
  - TYPE
    - EVENT-TO-EVENT
    - FIXED TIME STEP (Δt)
    - MESSAGE DRIVEN
    - REAL TIME
  - DATA STRUCTURE AND ALGORITHM
  - MULTIPLE PROCESSING AND MULTIPLE MODELS

3.5. Documentation

The documentation factors that affect joining existing simulation models and programs are given in Table 5. These factors are documentation of the simulation models, software systems, and computer programs. One set of documentation describes the simulation software systems, the models, and the model algorithms (if appropriate). Another set of documentation describes the computer codes of the simulation software systems and the simulation models.

If adequate documentation is not available, joining the models and computer programs may be difficult or impossible. Since good and complete documentation is extremely costly, it frequently occurs that adequate documentation does not exist. The specifics needed in the documentation depend on how the models and programs are to be joined which is determined by the factors and issues in the previous subsections.

TABLE 5: DOCUMENTATION FACTORS

- SIMULATION SOFTWARE SYSTEMS
- SIMULATION MODELS
- MODEL ALGORITHMS
- COMPUTER CODES
    - SIMULATION SOFTWARE SYSTEMS
    - SIMULATION MODELS

4. REUSABILITY

It should be clear from Section 3 that *when* it is possible to join existing simulation programs (i) *at least* a reasonable amount of effort will be required and frequently a large amount of effort will be needed and (ii) most likely not all of each simulation model and program will be able to be used. Because of this latter point, I believe that people should approach the joining of simulation programs with the view that only parts of existing models and programs can be used. Thus, I introduce the concept of "Amount of Reusability". The amount of reusability of existing models, software, and programs will depend upon the factors and issues in Section 3 and the objectives and goals of the study.

Different amounts of reusability will occur for the various elements involved in joining existing simulation programs. The elements are listed in Table 6. Model structure(s) and models were discussed in Section 3 and, therefore, the idea of amounts of reusability in these two areas should be clear. In many cases, algorithms are contained in the models. Most likely they will be usable when the models or some portions of the models are joined together. Thus, the amount of reusability will usually be high for these elements. Simulation software systems have conceptual designs which may be reusable if a simulation software system must be modified or redesigned. The simulation software system and simulation model computer programs may be used in part or totality. Thus, they have amounts of reusability, which hopefully are high as software development is costly and time consuming.

TABLE 6: REUSABILITY ELEMENTS

- MODEL STRUCTURES
- MODELS
- MODEL ALGORITHMS
- SIMULATION SOFTWARE CONCEPTUAL DESIGN
- SIMULATION SOFTWARE SYSTEM PROGRAMS (CODES)
- SIMULATION MODEL PROGRAMS (CODES)

5. RESEARCH

Two types of research should be conducted in this area. The first type needed is to study how existing simulation models and programs can be joined. This paper should be considered only as an initial step in this subject. Almost all of the topics (e.g., factors and issues) presented in Sections 2 and 3 need research.

The second type of research needed is basic or fundamental work in simulation modelling, simulation software, and simulation support systems to develop new technology to enable models and simulations to be easily joined, adapted to new uses, and to have a high amount of reusability. This research should include developing new ways to model, how to develop model structures (including new ones), models that can be used for model specification which are language independent, simulation languages which allow mixes of world views, and ways to have multiple models move through time and be kept synchronized. Only limited research is being done on these topics (e.g., the work in modelling using event graphs, condition specification, and multifaceted modelling) but *much* more is needed. I encourage others to do research in this area.

6. CONCLUSIONS

Joining existing simulation models and programs is an extremely complex task and subject. There are numerous factors and issues that affect if and how simulation models and programs can be joined. Usually only parts of models and software can be used when they are joined and thus the concept of "amount of reusability" was introduced. The effort required to join existing simulation models and programs, when possible, can be high. Research is needed to better understand how to join existing models and programs and how to find new ways to build models and programs to allow them to be joined more easily and to have a high amount of reusability.

ACKNOWLEDGEMENT

This work was sponsored by the Air Force Systems Command, Rome Air Development Center, Griffiss Air Force Base, New York 13441-5700.

AUTHOR'S BIOGRAPHY

ROBERT G. SARGENT is a Professor of Industrial Engineering and Operations Research and a member of the Computer and Information Science Faculty at Syracuse University. Dr. Sargent has served the Winter Simulation Conferences in several capacities, including being a member of the Board of Directors for ten years, Board Chairman for two years, General Chairman of the 1977 Conference, and Co-editor of the 1976 and 1977 Conference Proceedings. Professor Sargent was Department Editor of Simulation Modeling and Statistical Computing for the *Communications of the ACM* for five years, has served as Chairman of the TIMS College on Simulation and Gaming, and has received service awards from ACM, IIE, and the Winter Simulation Conference Board of Directors. He currently is an ACM National Lecturer, a member of the Executive Committee of the IEEE Computer Society Technical Committee on Simulation, and a Director-at-large of the Society for Computer Simulation. Dr. Sargent received his education at the University of Michigan. His current research interests include model validation, simulation methodology, simulation applications, performance evaluation, and applied operations research. Professor Sargent is a member of AIIM, the New York Academy of Sciences, Sigma Xi, ACM, IIE, ORSA, SCS, and TIMS, and is listed in Who's Who in America.

Robert G. Sargent
Department of Industrial Engineering
and Operations Research
Syracuse University
Syracuse, New York 13244-1240
(315)423-4348