# ON THE OPTIMUM STRUCTURE FOR WAR-GAME SIMULATIONS

A. T. Mollegen, Jr.
Military Systems Division
Mystic Oceanographic Co.
Mystic, Connecticut

The primary content of this paper is a consideration of the basic approaches which may be used in constructing a war-game simulation. It is assumed that the simulation sponsor regards the simulation solely as a means of obtaining answers, and that he desires dependable answers quickly and cheaply.

The life-cycle of a simulation is briefly discussed, and then the properties of a simulation as seen by (a) the simulation sponsor and (b) the simulation designer are discussed in some detail. A discussion is given of the comparative advantages and disadvantages of the Monte Carlo and probability-formula type simulations. An overall recommended approach is given, which consists of breaking the total game down into phases, and of developing independent simulations for each phase. Each phase can have its own optimum structure.

Table 1 shows the expected life cycle of a war-game simulation. If the simulation is not satisfactory to the sponsor, its life cycle may be terminated at any point.

The various factors of interest to the simulation sponsor are given in Table 2. Although the inexperienced simulation enthusiast may not want to accept the fact, it is usually the case that the most important factor is cost. Next may come availability, and while some minimum acceptable accuracy is an absolute requirement, the nature of many problems is such that quick, cheap, and approximate answers are much more desirable than long-delayed, expensive, and precise answers.

Now the basic purpose of a war-game simulation is to investigate situations for which experimental data has not been or cannot be gathered (such as situations involving hypothetical systems). The appropriate procedure is to construct a model using accepted principles of physics, validate (or "calibrate") it by comparison with experimental data, and then use it to predict what will happen in other situations. A simulation cannot be considered scientifically acceptable if it uses unexplainable formulas, if it has not been compared with reality, or if the simulation program, the physics involved, and the validation process are not all clearly documented.

The basic factors of interest to the simulation designer are shown in Table 3. These include the method of advancing the game, and the method of handling random factors.

Time can be advanced in the game by having the program move from event to event ("event-store" approach) or move from time point to time point. Time steps can be fixed, or can be adjusted according to what is happening in the game. The question of whether the event-store approach or the time-step approach is preferable seems to be dependent on whether the action being simulated is event-oriented or process-oriented. Decisions in the model can be handled by having the action be completely predetermined, or by having the decision rules which each combatant must use be predetermined.

A major factor of interest to the program designer is the way in which random processes are treated. Two basic approaches are the Monte-Carlo approach and the probability-formula approach. In the Monte-Carlo approach, computer routines are used which generate random numbers having statistics analogous to those of the process being simulated. In the probability-formula approach, probabilistic expressions are used directly to compute the probabilities of the possible outcomes of each random process. A comparison of the features of Monte Carlo and formula-type simulations is given in Table 4.

The Monte-Carlo models are analogs of what occurs in real-world processes, whereas the formula-type require the additional step of being probabilistically formulated. This additional step can be prohibitively difficult, especially when time constraints are involved. When this step is not prohibitively difficult, it may however prove to be a very good investment. Either type may use the event-store or time-step approach. A Monte-Carlo program is relatively flexible as to effectiveness criteria; a formula-type program has its criteria built in. The output of a formula-type program is directly a computed probability, but the output data from a Monte-Carlo program must be further processed before they can be used. The Monte-Carlo computer program is complicated, and because 50 to 100 replications of the game must be made in order to find a probability, it is effectively quite slow; a formula-type program is a simple program, and quite fast. Comparative effectiveness is easy to show with a formula-type model, but can be difficult to show conclusively with a Monte-Carlo model due to confidence-level problems with the game statistics. Simulation programs tend to repetitively cycle through model subroutines, and thus become slow if core memory is exceeded and subroutines must be stored on disk. Since Monte-Carlo programs tend to be large, this is a particular problem with Monte-Carlo programs. As the number of decisions

allowed to the game participants increases, the size of a Monte-Carlo program does not increase particularly rapidly, but the difficulty in de-bugging it rises extremely rapidly. On the other hand, the conceptual complexity of a formula-type model rises very rapidly with the number of decision possibilities, serving as a limit on the number of decisions which may be considered.

The above factors having been considered, what then is the optimum structure for war-game simulations? Since no explicit answer can be given to such a question, we offer the following suggested approach. Break the overall game down into phases, and simulate each phase in the manner most suited to the phase, defining the outputs of each simulation to be compatible with the inputs to the next. This way each program is more manageable, and is easier to generate, validate and use.

Acknowledgement

The opinions expressed are the author's responsibility, but have been strongly influenced by conversations with Mr. W. Voigt, Mr. J.Pulos, and Mrs. S. Voigt of the Naval Ship Research and Development Center, and Mr. R. Miller of the ASWSPO Systems Analysis Group.

Table 1. Outline of the Simulation Life Cycle

Phase A - Creation of the Simulation

1. Simulation Requirements
2. Math Models
3. Programming
4. De-bugging
5. Documentation

Phase B - Validation of the Simulation

1. Criteria Selection
2. Run Plans
3. Runs
4. Run Analysis
5. Repeat 2, 3, and 4, as Necessary
6. Documentation

Phase C - Problem Solving with the Simulation

1. Problem Analysis
2. Run Plans
3. Runs
4. Run Analysis
5. Repeat 2, 3, and 4, as Necessary
6. Documentation

Phase D - Modification and Use for New Purposes

1. Repeat Phase A
2. Repeat Phase B
3. Repeat Phase C

Table 2. Factors of Interest to the Simulation Sponsor

A. Cost

1. Non-Recurring
   (a) Phase A - Create
   (b) Phase B - Validate
2. Recurring
   (a) Phase C - Use

B. Availability

1. Response Time
   (a) Construction Delay
   (b) Pre-Question Delay
2. Special Personnel/Computers

C. Accuracy and Validity

1. Repeatability and Confidence Level
2. Validity

D. Flexibility

1. Different Effectiveness Criteria
2. Different Problems

E. Believability

1. Scientific Basis
2. Validation
3. Documentation
   (a) Simulation Program
   (b) Scientific Basis
   (c) Validation

Table 3. Factors of Interest to the Simulation Designer

A. Method of Advancing the Game

1. Time
   (a) Event-to-Event
   (b) Time Step
       -fixed step size
       -adjustable step size
2. Decisions
   (a) Predetermined Actions
   (b) Prestored Decision Tables

B. Method of Representing Random Processes

1. Monte Carlo Models
2. Probability Formula Models

Table 4. Comparative Features of Monte-Carlo and Formula-Type Simulations

| MONTE-CARLO | FORMULA-TYPE |
|---|---|
| Natural Models | Natural Models & Probabilistic Models |
| Event-Store or Time-Step | Event-Store or Time-Step |
| Flexible Criteria | Inflexible Criteria |
| Output Analysis Needed | Direct Output |
| Program Complicated | Program Simple |
| Simulation Slow | Simulation Fast |
| Differential Probabilities Hard | Differential Probabilities Easy |
| Computer Requirements Stringent | Computer Requirements Slight |
| More Decisions ⟶ More Checkout Problems | More Decisions ⟶ More Model Complexity |