Proceedings of the 1988 Winter Simulation Conference
M. Abrams, P. Haigh, and J. Comfort (eds.)

One system, several perspectives, many models

James O. Henriksen
Wolverine Software Corporation
7630 Little River Turnpike, Suite 208
Annandale, VA 22003-2653

ABSTRACT

This paper is a teaching piece. It is the latest in a series of papers by the author on the general subjects of modeling and problem solving. (See (Henriksen 1981), (Henriksen 1986), and (Henriksen 1987).) For the third year in a row, Dr. Alan Pritsker is also presenting a paper in a very similar vein. (See (Pritsker 1986), (Pritsker 1987) and (Pritsker 1988).) Section 1 presents a classic modeling problem first considered around the turn of the century (Lanchester 1916), long before the advent of digital computers. The system considered is a simplified battle between two opposing infantry forces. In subsequent sections, the battle is viewed from both a discrete-event and a continuous perspective. For each perspective, a variety of models is presented, and the comparative strengths and weaknesses of the various models are compared. For the student of modeling, the variety of models hammers home the point that for most modeling problems, one should consider several alternative perspectives before building a model.

1. STATEMENT OF THE PROBLEM

The objective of this exercise is to build a model to predict the number of red soldiers expected to survive a battle between red and blue armies of given initial sizes. The following simplifying assumptions are made:

A. Combat is waged entirely by riflemen, firing at long range. As a consequence of this assumption, a rifleman can aim with equal ease at any (uniformly) randomly chosen adversary.

B. No two soldiers ever select the same target. (In a real battle, some casualties would be hit by more than one bullet, particularly when their army was overwhelmed.)

C. The battle is fought until one side is completely annihilated. (There are no prisoners taken.)

D. Every shot fired is lethal. In a real battle, non-zero probabilities would exist for completely missing a selected target and for inflicting non-lethal wounds.

E. Every shot, and the resulting fatality, is instantaneous. As a consequence of this assumption, a soldier cannot shoot "on the way down."

F. The red and blue armies are equally efficient, i.e., neither side has an advantage in marksmanship.

2. DISCRETE-EVENT SIMULATION MODELS

2.1 The "Active Soldier" World-View

A system such as a military battle can be viewed from a variety of perspectives. One very natural perspective which can be implemented straightforwardly in almost any discrete-event simulation language is to view system behavior through the eyes of the individual soldier. The behavior pattern of a soldier is as follows:

A. Select an enemy soldier to aim at.

B. Aim and fire.

C. Remove the enemy soldier from the model.

D. Repeat this behavior pattern as long as "I" (the individual soldier) am still alive and any enemy soldiers remain.

In the above behavior pattern, simulated time must elapse during steps A and B. Since the soldier executing the behavior pattern is himself a potential target, on any given iteration of the behavior pattern, he may not make it to step C. To properly handle the demise of a soldier, two approaches are possible:

A. Some form of preemptive scheduling can be used to "unschedule" a soldier's scheduled completion of activity A or activity B, replacing this activity with an impromptu exit from the model.

B. In some discrete event languages, preemptive scheduling may present implementation difficulties. As an alternative, a "fatality switch" variable could be included for each soldier. Step C of the behavior pattern could then be expanded to include setting the fatality switch for the enemy soldier, and a step could be inserted between steps B and C to have the soldier test whether his own fatality switch has been set by an enemy soldier. If he found his own switch set, he would exit the model. This exit would be "late," in the sense that simulated time would have elapsed between the setting of the switch and the soldier's reaction to that unfortunate event. Late recognition of the switch setting wouldn't make any difference to the operation of the model, since the soldier would exit prior to inflicting another enemy casualty. For an expanded treatment of this sort of alternative to preemptive scheduling, see (Henriksen 1987).

2.2 The View from Above

Consider the perspective of an imaginary observer looking down on the battle from above. From this perspective, the battle appears as follows:

A. Observe which side fires next.

B. Reduce the size of the opposing army by one soldier.

C. Continue this behavior pattern until one side has been annihilated.

At any given time, "who fires next" is a uniformly distributed random variable, determined by the ratio of the remaining sizes of the opposing armies. For example, if 100 red soldiers and 50 blue soldiers remain, the probability that a red soldier is the next to fire is 2/3. In general, if there are r red soldiers and b blue soldiers, the probability of a red soldier firing next is r/(r+b) and the probability of a blue soldier firing next is b/(r+b). With each shot fired, this probability changes. Thus if many battles are simulated, even a badly outnumbered army will occasionally win, if it has a "streak of good luck."

This world-view enables an important simplification of the simulation model: *time becomes irrelevant.* By using a uniform distribution to model "who fires next," we determine the sequence in which casualties occur. This sequence is time-independent. This allows us to replace our simulation model with a Monte Carlo model, i.e., a model which performs random sampling, but includes no time delays.

3. A DISCRETE PROBABILITY MODEL

Although the Monte Carlo model of section 2.2 is a distinct improvement over the simulation model of section 2.1, further simplifications can be made. Our objective is to compute the expected number of red soldiers to survive the battle. Let Er(r,b) denote the expected number of red soldiers to survive a battle between r red soldiers and b blue soldiers. For any r and b, Er(r,b) can be computed as the sum of Er(r,b-1) and Er(r-1,b), with each expectation weighted by its probability of occurrence. Er(r,b-1) is the expected number of red soldiers to survive, given that the next fatality is a blue soldier. The probability that a blue soldier is the next fatality is r/(r+b). Likewise, Er(r-1,b) is the expected number of red soldiers to survive, given that the next fatality is a red soldier. The probability that a red soldier is the next fatality is b/(r+b). Thus Er(r,b) can be expressed as follows:

$$Er(r,b) = Er(r,b-1) * (r/(r+b)) + Er(r-1,b) * (b/(r+b))$$

Note that the above definition of Er(r,b) is *recursive*; i.e., if we write a function to evaluate Er(r,b), in the process of calculating the value of Er(r,b), we must invoke Er(r-1,b) and Er(r,b-1). Writing such a function is easy in languages which directly support recursion, e.g., the "C" language. In languages such as Fortran, recursive functions cannot be written directly. Ease of implementation notwithstanding, recursive functions always require one or more terminating conditions. For Er(r,b), the terminating conditions are as follows:

$$Er(r,0) = r$$
$$Er(0,b) = 0$$

An example of a recursive function for evaluating Er(r,b), written in C, is shown in Figure 1. Much faster, non-recursive evaluation of Er(r,b) is possible by first evaluating and saving the values of Er(1,0),...,Er(1,b), and then evaluating and saving the values of Er(2,0),...,Er(2,b), etc. Note that Er(2,j) is a function of Er(1,j) and Er(2,j-1), both of which can be computed and saved in a vector of length b prior to evaluation of Er(2,j).

4. A CONTINUOUS SIMULATION MODEL

Thus far, our combat models have assumed that attrition occurs in discrete steps (individual fatalities). With some loss of accuracy, attrition can be viewed as a *continuous* process. Accuracy is lost by ignoring random variation. Loss of accuracy is discussed in section 6, below.

The *rates* at which r and b change over time are as follows:

$$\frac{dr}{dt} = -K\,b$$

$$\frac{db}{dt} = -K\,r$$

The differential equations shown above can be interpreted in words as follows:

> The instantaneous attrition rate for the red (blue) army is the number of casualties inflicted per unit time by the blue (red) army, which is proportional to the current size of the blue (red) army. For example, K might equal 10 fatalities / minute / soldier. Thus, if b = 1,000, dr/dt = -10,000 soldiers / minute.

Modeling the above differential equations is a trivial exercise in virtually any continuous simulation language. Because the equations are simple and "well-behaved," a hand-coded implementation of Euler's method could also be written very quickly. (Euler's method is the simplest of all integration methods. For a given value of x, f(x+h) is approximated by f(x) + h f '(x), where h is a "small" increment, which is the step size of the algorithm.)

5. THE LANCHESTER N-SQUARE LAW

In (Lanchester 1916), Frederick William Lanchester, a British mathematician, presented a mathematical analysis of combat which has become famous in military modeling circles. Lanchester considered red and blue armies of unequal efficiency. Using M as the efficiency of the blue army and N as the efficiency of the red army, he expressed the following differential equations:

$$\frac{db}{dt} = -N\,r * \text{constant}$$

$$\frac{dr}{dt} = -M\,b * \text{constant}$$

For conditions of equality of forces, he reasoned that the attrition rate of the red army, divided by the size of the red army, must equal the attrition rate of the blue army, divided by the size of the blue army. For example, a red army of 10,000 soldiers losing 1,000 soldiers per hour is equal to a blue army of 20,000 losing 2,000 soldiers per hour. In other words,

$$\frac{db}{b\,dt} = \frac{dr}{r\,dt}$$

or

$$\frac{-N\,r * \text{constant}}{b} = \frac{-M\,b * \text{constant}}{r}$$

or

$$N\,r*r = M\,b*b$$

In Lanchester's words, "the fighting strength of a force may be broadly defined as proportional to the square of its numerical strength, multiplied by the fighting value of its individual units." Lanchester further reasoned as follows, for forces of equal efficiency:

$$\frac{dr}{db} = \frac{b}{r}$$

Therefore,

$$r\,dr = b\,db$$

If one assumes arbitrarily small values for dr and db and considers the reductions in fighting strength which result when armies of sizes r and b are reduced to sizes (r-dr) and (b-db), respectively, red fighting strength is reduced from r*r to (r-dr)*(r-dr), and blue fighting strength is reduced from b*b to (b-db)*(b-db). Expanding the expressions for reduced strengths yields the following:

$$(r-dr)*(r-dr) = r*r - 2\,r\,dr + dr*dr$$
$$(b-db)*(b-db) = b*b - 2\,b\,db + db*db$$

Since dr and db are arbitrarily small, the dr*dr and db*db terms above can be ignored. Since we know that r dr = b db, we can see that the strengths of the red and blue armies are reduced by the same amounts, 2 r dr = 2 b db. Therefore, for given sizes r and b, r*r - b*b is constant, or r*r - c*c = b*b.

If we assume that the red army is larger than the blue army, the blue army will always be annihilated under the Lanchester formulation. Therefore, for starting sizes of R and B, when b goes to zero, the final strength of the red army will be given by SQRT(R*R - B*B).

6. COMPARISON OF RESULTS

Programs were written to implement four of the models discussed above. The discrete event simulation models were written in GPSS/H, and the remaining three programs were written in C. The GPSS/H models were run for 50 replications, to get "reasonable" results. The results from running these programs are shown in Table 1.

The following observations are offered on the results shown in Figure 1:

A. For armies of equal size, the continuous models predict zero survivors for *both* sides. This is because the continuous view of this problem ignores random variations. In the real system, "lucky streaks" occur, dramatically affecting the outcome of a battle. Thus, in the real system, the expected number of survivors, taken over many battles, is greater than zero for equally matched sides. For armies of significantly different sizes, all models produce roughly equivalent results.

B. The discrete probability model calculates exact results, but for large army sizes, the CPU time taken may become prohibitively large, since it is proportional to the product of the initial sizes. In addition, when large sizes are used, the computation of expectations may result in floating point underflows, since the probabilities can become exceedingly small.

C. The active soldier simulation model consumes the most CPU time, by a long shot. In addition, it requires large amounts of computer memory (to represent each and every soldier).

D. Results achieved by explicitly integrating differential equations converge to the Lanchester result as the step size of the integration method is made smaller. A more sophisticated integration method would have consumed far less CPU time than the simple Euler's method employed.

7. CONCLUSIONS

7.1 A Multiplicity of Models

The military system we have considered in this paper has been viewed from both discrete and continuous perspectives. Using the discrete perspective, a straightforward model, a more abstract model, and a probability model which produces a calculated, exact answer have been presented. Using the continuous perspective, the explicit integration of equations and a closed-form solution have been presented. In closing, we will compare the models on the basis of fidelity, elegance, execution speed, and ease of modification.

7.2 Fidelity

The continuous models presented in this paper embody a known departure from reality: fatalities occur in discrete units (of 1) at discrete instants in time. Fatalities are not an inherently continuous process. Nevertheless, adopting this viewpoint fails only when modeling armies of nearly equal size. When armies of considerably differing sizes are compared, all the models produce results which are sufficiently faithful to the operation of the "real" system.

7.3 Elegance

The "Active Soldier" discrete event model is totally inelegant; it is a "brute force" model. By contrast, the Lanchester formulation is the quintessence of elegance. While the explicit evaluation of discrete probability expectations is also an elegant approach, this method is costly and possibly computationally infeasible (due to floating point underflow problems) for large sizes of opposing armies. The "View from Above" discrete event model is an interesting midpoint on the scale of elegance. It is more abstract and more computationally efficient than the "Active Soldier" model, but is not nearly as appealing as either the Lanchester or discrete probability models.

```
double expect(nred, nblue)              /* Recursive function */

        int             nred,
                        nblue;

        {
        if (nred == 0)
                return(0.0);

        if (nblue == 0)
                return nred;

        return expect(nred, nblue-1) * nred + expect(nred-1, nblue) * nblue) /
                        (nred + nblue);

        }       /* End of expect(nred, nblue) */
```

Figure 1 - Recursive Calculation of Expected Number of Survivors

Table 1 - Comparison of Results

	100 Red vs. 100 Blue		1000 Red vs. 750 Blue	
	Red Survivors	CPU Time (Seconds)	Red Survivors	CPU Time (Seconds)
Discrete Event Simulation "Active Soldier View"	15.5	36.17	659.5	286.85
Discrete Event Simulation "View from Above"	16.0	12.08	664.9	29.54
Discrete Probability Model (Fast, Non-recursive)	16.5	2.01	660.5	135.56
Explicit Integration of Differential Equations	0.0	1.44	661.4	1.44
Lanchester N-Square Law	0.0	insig	661.4	insig

7.4 Execution Speed

It is interesting to note that the least sophisticated model has the longest running time, and the most elegant model consumes an insignificant amount of CPU time. This is as life should be, where virtuosity, if not virtue, is properly rewarded.

7.5 Ease of Modification

Suppose that one wanted to build a more realistic model of a military battle. If this were the case, some of the simplifying assumptions made in section 1 would have to be eliminated. For example, the assumption that no two soldiers ever select the same target is very unrealistic, particularly when their side overwhelms the enemy. The more abstract a model is, the more difficult it is to extend the model to add additional fidelity to a complex system. For example, extending Lanchester's mathematics would be very difficult for most modeling practitioners. By contrast, the simplest model, the "Active Soldier" model, is the easiest to modify. This model is a rather uninspired direct program analog of the system being modeled; i.e., the logic of the simulation program parallels the logic of the system in an obvious manner. There is, in principle, no limit on the extent to which a program can be modified. In reality, practicality dictates upper bounds on program size and complexity, particularly if the programs are poorly structured to begin with.

The conclusions reached in this paper are very similar to those reached in (Pritsker 1986). Readers (and particularly students and teachers of modeling) who are interested in further examples should read or reread Pritsker's paper as well.

REFERENCES

Pritsker, A. Alan B. (1986) Model Evolution: A Rotary Index Table Case History. In: *Proceedings of the 1986 Winter Simulation Conference* (J. Wilson, S. Roberts, J. Henriksen, eds.). Society for Computer Simulation, San Diego, California, 703-707.

Pritsker, A. Alan B. (1987) Model Evolution II: An FMS Design Problem. In: *Proceedings of the 1987 Winter Simulation Conference* (A. Thesen, H. Grant, and W. D. Kelton, eds.). Society for Computer Simulation, San Diego, California,567-574.

Pritsker, A. Alan B. (1988) Modeling Viewpoints for Assessing Reliability. In: *Proceedings of the 1988 Winter Simulation Conference* (M. Abrams, P. Haigh, and J Comfort, eds.). Society for Computer Simulation, San Diego, California.

Henriksen, James O. (1981) GPSS - Finding the Right World-View. In: *Proceedings of the 1981 Winter Simulation Conference* (T. I. Oren, C. M. Delfosse, and C. M. Shub, eds.). Society for Computer Simulation, San Diego, California, 505-515.

Henriksen, James O. (1986) You Can't Beat the Clock: Studies in Problem Solving. In: *Proceedings of the 1986 Winter Simulation Conference* (J. Wilson, S. Roberts, J. Henriksen, eds.). Society for Computer Simulation, San Diego, California, 713-726.

Henriksen, James O. (1987) Alternatives for Modeling of Preemptive Scheduling. In: *Proceedings of the 1987 Winter Simulation Conference* (A. Thesen, H. Grant, and W. D. Kelton, eds.). Society for Computer Simulation, San Diego, California.

Lanchester, Frederick William (1916) Mathematics in Warfare Reprinted in: *The World of Mathematics* (James R. Newman, ed.). Simon & Schuster, New York, New York.

AUTHOR'S BIOGRAPHY

JAMES O. HENRIKSEN is the president of Wolverine Software Corporation, which he founded in 1976 to develop and market GPSS/H, a state-of-the-art version of the GPSS language. Since its introduction in 1977, GPSS/H has gained wide acceptance in both industry and academia. From 1980-1985, Mr. Henriksen served as an Adjunct Professor in the Computer Science Department of the Virginia Polytechnic Institute and State University, where he taught courses in simulation and compiler construction at the university's Northern Virginia Graduate Center. Mr. Henriksen is a member of ACM, SIGSIM, SCS, the IEEE Computer Society, ORSA, and SME. A frequent contributor to the literature on simulation, Mr. Henriksen served as the Business Chairman of the 1981 Winter Simulation Conference and as the General Chairman of the 1986 Winter Simulation Conference. He presently serves as the ACM representative on the Board of Directors of the Winter Simulation Conference.

James O. Henriksen
Wolverine Software Corporation
7630 Little River Turnpike, Suite 208
Annandale, VA 22003-2653

(703) 750-3910