Proceedings of the 2000 Winter Simulation Conference
J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, eds.

RECURSIVE SIMULATION TO AID MODELS OF DECISIONMAKING

John B. Gilmer, Jr.
Frederick J. Sullivan

Wilkes University
P.O. Box 111
Wilkes-Barre, PA 18766, U.S.A.

ABSTRACT

Recursive simulation involves a simulation, or an entity in the simulation, creating another instance of the same simulation, running it and using its results. This is particularly applicable to decisionmaking in a military simulation. Simulation can be used by the simulated command elements to evaluate the implications of the possible choices. The simulation that is already running provides a conveniently available evaluation tool if it can be run recursively. This has been done with the "eaglet" simulation, and data collected that indicates that the quality of decisionmaking by the simulated command elements is improved.

1 BACKGROUND

Representing the human decisionmaking process in military simulations is a critical challenge. The fact that such simulations are used in studies to inform management decisions is an indication that combat simulations are a useful tool to aid the decisionmaker. Why not let simulated decisionmakers also use this tool? Indeed, as the simulated decisionmaker is already operating in the context of a simulation, there is a simulation readily at hand: the one which is now running.

Recursion is an operating principle much valued by computer scientists. Yet, recursive use of simulation in this manner to aid modeling of military decisionmaking has not been used to the authors' knowledge. There are reasons for this: writing recursive software is not trivial, especially for something as complicated as a simulation. Not only the simulation, but also the study process including framing the experiment and evaluating the results must be included.

In the particular simulation used for this experiment, prior development of "Multitrajectory Simulation" techniques made the mechanics of recursion much more manageable. In multitrajectory simulation, random events may be handled by creating clones of the simulation state, and having each continue its trajectory with a different outcome of the event. The "eaglet" simulation had already been developed with the features needed to support multiple states and cloning of states for events that may be deeply embedded in the simulation's function hierarchy. In comparison, the modifications necessary to support cloning and recursion for the entire simulation were not difficult.

2 THE SIMULATION

The "eaglet" simulation was developed specifically to investigate multitrajectory simulation issues. It was intended to have many of the same kinds of features as "Eagle", used by the Army for Force on Force analysis, to the extent necessary to examine multitrajectory issues, but to be much simpler. Both Eagle and eaglet explicitly represent decisionmaking by simulated military commanders. Eagle can be considered "free play" in the sense that it is the orders of these simulated commanders, and similar orders taken as input, that constrain the movements and operations of the military units, rather than prescribed sectors or paths to which the units are bound independent of the C2 representation. This representation puts a premium on good modeling of the command control function, and is essential if one is to study the impact of processes that influence the battle via the influence of command. The eaglet simulation includes only some of these capabilities. For example, the version used in this study does not include operation planning by simulated commanders. However, it does allow modeling of alternative commitments of force in response to a situation. It is this feature which has been the focus of the use of recursive simulation.

"eaglet" represents nominally battalion level aggregated units organized hierarchically into brigades and divisions. The units move along paths that allow for alternative routes. That is, the routes contain branch points from which a unit may decide to move in either of two or three different directions, eventually reaching the same destination given by the unit's objective. The simulation includes options for stochastic representation of acquisition, loss of acquisition, attrition, movement choices, and decisionmaking. Forces may be represented either as a single abstract number for "force" or as a list of numbers of different kinds of weapons (e.g. tanks, APC's, infantry, trucks, etc.). The attrition equations are Lanchester Square Law: the attrition inflicted is proportional to the number of weapons firing. There are modifications for unit orientation, the operation being conducted, and range. The version of eaglet used for this study did not include terrain.

The decisionmaking of interest occurs in the context of operation plans being executed by the entities in the simulation representing military units. These orders are developed from templates by an automated operation planner that is used as a preprocessor to the simulation proper. (For the multitrajectory studies, this planner applied random variations that were used for automatically generating scenarios that were similar but not identical.) Figures 1 and 2 below illustrate the template and the contingency and route structure for a brigade level "attack" operation. The key decision to be made by the command element after the operation is initiated is the commitment of the reserve. It is this decision which is informed by the recursive use of the simulation. In addition, individual unit level decisionmaking based on effectiveness, the presence of enemy units and flank threats etc. is represented.

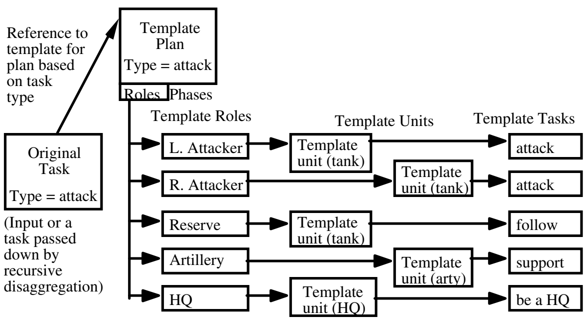

Figure 1: Brigade Attack Planning Template

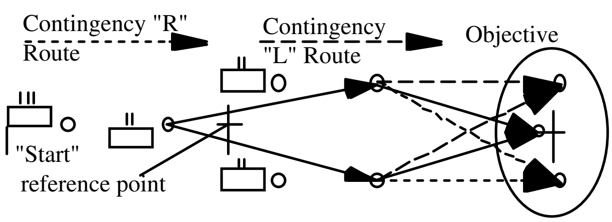

Figure 2: Template With Contingency for Reserve

Figure 3 shows a two division sized scenario used for this study. Both divisions consist of two brigades plus a large reserve battalion that is used as a reserve. These relatively small divisions (only 7 maneuver battalions) do have a real-world prototype in the German 1944 Volksgrenadier division. The use of a battalion sized reserve rather than a full brigade was necessary since this version of "eaglet" does not include the planner necessary to plan a brigade level operation responsive to the situation that exists as of commitment. That operation cannot easily be pre-planned, since the particulars are highly variable. Both Red and Blue units have this organization, but the sizes of the Red units are somewhat smaller and their weapons less capable. The figure also shows, for the Blue units, the routes those units may take. Note that there are three reserve units, one for each brigade, and the division reserve. These are labeled 10, 16, and 4 respectively for Blue. The grid lines are nominally 5 km. apart. Some of the HQ and artillery units are off the left and right edges.

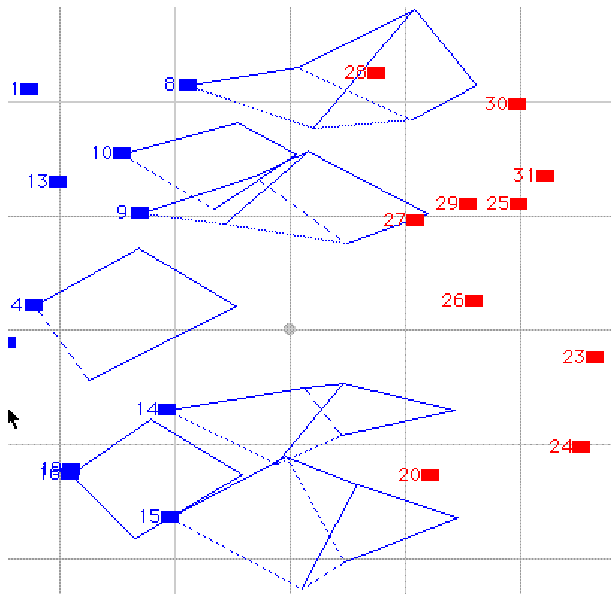

Figure 3: "eaglet" Two Division Scenario

A key attribute of certain processes in "eaglet" is that they are (potentially) stochastic. Decisions made at divergent route points for individual units have already been mentioned. Each choice is characterized by a probability. If deterministic mode is chosen for the simulation run, the choice having the highest probability is used. If acquisition events are resolved stochastically, units outside a certain radius are not seen, those inside a smaller radius are seen, and those at intermediate distances are seen with a 50% probability. If instead the event is resolved deterministically, a threshold distance is used half way between the two radii mentioned above. (The radii values depend on the operational activity of the unit.) Acquisition loss works similarly, but at larger radii. Stochastic resolution of attrition has been provided in "eaglet" but was not used in this study. The event of most interest to this study is the decisionmaking event, addressed below.

The "eaglet" simulation is more fully described in Gilmer (1998). In addition, various papers and other documentation are posted in the following location: `<http://calvin.mathcs.wilkes.edu/mts>`.

3 PROGRAMMING CONSIDERATIONS

Multitrajectory simulation is inherently difficult to program. The principal difficulty is the problem of reentering a simulation at a specific point without redoing all of the preceding computation, and several different techniques have been devised to do this, as described in the reference above. Not surprisingly, the same techniques can be used to start up a recursive simulation from any desired point in the main simulation.

In non-recursive simulations, it is convenient, when navigating among the complex data structures (states, units, etc.) to make some of the simulation variables global (or, equivalently, static members of the simulation class). The techniques mentioned above include methods to restore the values of these variables to appropriate values when a particular simulation trajectory is finished. However, these techniques also use (global) simulation class variables as part of the multitrajectory control mechanism. These variables can no longer be global in recursive simulations, and so must become instance variables of a simulation object. When a recursive simulation is started, a new simulation object is created to control it. However, this is a relatively minor change. If the multitrajectory mechanisms had not already existed, something like them would have had to be invented to support recursive simulation.

4 THE DECISIONMAKING MODEL

Most decisionmaking in "eaglet" is based on rules that can refer to characteristics of the unit, such as its effectiveness, whether it is in combat, at its objective, and such. These binary rules are potentially stochastic, with variations on the thresholds for "effectiveness" and "being at the objective" used to discriminate among different probabilities for rule firing. Foe example, the following rule applies to units with an intent of "delay" or "defend":

52: IF unit is at objective, in contact, and ineffective,
THEN intent = "withdraw",
operational activity = "delay", and
generate a new objective 10 km to the rear

(The unit is in serious trouble and withdraws to preserve itself. The "delay" is used to attempt an orderly withdrawal.)

There are actually three different measures of the unit's "effectiveness". The default (deterministic) thresholds are at 85% (marginally effective) and 70% (ineffective). A second set of thresholds are 5% higher, and a third set are 5% lower. The outcome of testing this rule is the sum of the 0/1 outcomes from testing it under each set of criteria. A rule that does not fire under any of these sets of criteria has outcome 0 and never fires, even in stochastic mode. Similarly, one that evaluates true under all criteria has outcome 3, and will fire always, regardless of resolution method. But for those with outcome levels 1 and 2, in stochastic mode a probability is assigned to whether the rule fires or not. Currently the probabilities assigned are 40% and 80% respectively for the simplest rules that evaluate only one condition. Rule 52 above, which requires 3 conditions to be true, has a probability of firing at level 2 of 80% cubed, or 51%, and half that probability of firing if only satisfied at outcome level 1. Criteria for units being at, approaching, or closing on an objective are subject to similar variations in distance. Some rules which are evaluated using code instead of Boolean products have probabilities assigned in that code.

In the version of "eaglet" used for this study, such rules are tested periodically, at a time step of five minutes. This "memoryless" mode is the simplest to implement, and corresponds to an assumption that decisionmakers are constantly considering their options. A different version of "eaglet" has provision for "hysteresis" in rule testing: a rule once tested and failing to fire is not tested again until the probability increases or the operational context changes, such as when another rule fires. This corresponds to a mode in which the decisionmaker does not reexamine a decision unless circumstances change. The latter approach has significant benefits to multitrajectory simulation as well as recursive simulation, in that fewer decision points are reached, and consequently fewer invocations of recursive simulation use are necessary. The probabilities to support each of these models of decisionmaking are different. Probabilities needed to characterize decisions in analytic grade simulations will require careful study and derivation that are well beyond the scope of work done with "eaglet". The numbers used in "eaglet" are admittedly rather arbitrary.

Certain decision rules, including those for the commitment of reserves, are implemented directly with C++ code instead of data to a more general purpose rule mechanism. The code for one of the reserve commitment rules (rule 11) can be paraphrased as follows:

11: IF 1st subordinate % strength is below X and
1st subordinate combat situation is Y
THEN (rule fires)

For the different levels of evaluation, the value of X is 60%, 70%, and 80%. The combat situation Y is more complicated, and is an average sum of Boolean variables that characterize for the unit whether if is firing, fired upon, or has a left or right flank threat. Thresholds for Y are 1.0, 1.4, and 1.8. This rule has a 40% chance to fire at level 1, and 80% chance at level 2. The consequence of the rule firing depends on how it is used. In the order structures of this scenario, it is used to trigger a contingency order for the reserve to move to the support of the first subordinate. A similar rule (Rule 12) supports the 2nd subordinate.

5 RECURSIVE SIMULATION USE

A version of "eaglet" was modified to include in the decision routine provision for calling a function, "doC2study" between obtaining the outcome level for a rule and taking action on that outcome, for outcome levels of 1 and 2 (the rule fires under some but not all sets of criteria). The C2study function merely returns the outcome level for all cases except those where a reserve commitment decision is being made. In those cases, it creates a new simulation object, and initializes it with a number of states, half of which are to proceed with outcome level 1 and the other half with outcome level 2, for the rule being evaluated. The number of states and the policies for treatment of (potentially) random events are specified in data that applies to the given level of recursion. When a state in the new simulation is continued with outcome level 1 and a policy of deterministic resolution, then the rule does not fire. If the level is 2 and the resolution is deterministic, then the rule does fire. When the recursive runs are completed, data are collected for average blue and red losses for each of the sets. The outcome corresponding to the most successful set (for the side making the decision) is then returned from the study, and the rule under consideration will then fire (or not fire) accordingly. Success was evaluated in terms of the better Blue-Red loss differential, rather than loss exchange ratio, in order to avoid computational problems in cases where losses were very low.

To control the use of recursive simulation, six new choice policies were added to those already used for multitrajectory studies :

- 0 Events resolved deterministically
- 1 Events resolved stochastically (random choice)
- a Events deterministic, except reserve commitment, which are resolved after a recursive simulation study
- b Events are deterministic except, for Blue only, the reserve commitment decisions, for which a recursive simulation call is made to study and advise
- c Events are deterministic except for Red reserve commitment, which gets the recursive treatment
- d Events are stochastic, except reserve commitments which are controlled by recursive simulation results
- e Events stochastic, except Blue reserve commitments are governed by recursive simulation results
- f Events stochastic except Red reserve commitment

(Policies 2 to 9 specify multitrajectory methods that were not used in this study.)

These controls apply individually to each type of event. The new policies (a to f) apply only to decisionmaking. The controls also vary with the level of recursion. Typically, the recursion runs will be done with fewer replications and simpler controls. In addition, each level of recursion has specifications for how attrition is to be modeled (individual weapons or aggregate strength) and the number of trajectories to be run.

6 RESULTS AND DISCUSSION

The initial, and simplest, set of runs were entirely deterministic, except that at the base level Blue and Red reserve commitment decisions were in certain cases made after deterministic simulation runs were made to predict the benefits of each of the two choices (rule firing or not). Each such recursive simulation study consisted of two deterministic trajectories, one with the rule firing and the other where it did not. Table 1 summarizes the Blue and Red losses in each case:

Table 1: Deterministic Study

Losses	Blue	Red
No recursion	214	402
Both use recursion	250	387
Blue only uses recursion	208	449
Red only uses recursion	202	385

When Blue uses the simulation to inform its decisionmaking, it does better in terms of loss differential than when it used the default deterministic decisionmaking method instead. Red profits less from recursion. This may be due to differences in recursive and deterministic modelling of the other Red headquarters. Another possibility is that deterministic simulation of such a complex model gives results that may be considered chaotic: they can be very sensitive to small changes in initial conditions or events as they occur. Interestingly enough, when both sides use recursive decisionmaking, the losses are maximum for Blue.

In a sense, this study indicates the scope of possible impact from these particular decisions. For all events except the reserve commitment decision, each side can project quite closely the course of the battle. (The correspondence is not exact because in the base representation individual weapon types were used for attrition, while in the recursive runs an aggregated strength was used instead.) For decisions other than the one of the moment, the default deterministic decisionmaking was used in the studies initiated to inform decisions. Thus, the models used by simulated objects in their decisionmaking, and the method used for decisionmaking itself, were the same except for the relatively few, albeit important, commitment decisions.

The study was repeated, but with stochastic representation of events other than decisionmaking. Now the models used by decisionmakers in the simulation and the simulation models themselves still are the same, but the decisionmakers' models (the recursive calls) cannot correspond exactly to the actual future course of events due to randomness in the acquisition and movement process. The decisionmakers in the simulation are still using deterministic models for these processes, as well as decisionmaking. One hundred trajectories were run at the base level and averages taken for the respective losses. Table 2 summarizes these results. The Blue loss standard deviation estimates ranged from about 30 to 50 in these and later results, and Red losses also showed comparable variability. For these averages of 100 trajectories, then, we can expect that variations beyond 10 are significant.

Table 2: Stochastic Base Model for Movement and Acquisition with Deterministic Recursion

Losses	Blue	Red
No recursion	157	245
Both use recursion	221	357
Blue only uses recursion	215	373
Red only uses recursion	172	266

In these cases, the stochastic models used in the base simulation level give somewhat different results from the deterministic model; losses are much lower for both sides. Yet, the decisionmaking for both sides (where recursive methods are used) is informed by the results of deterministic models. Thus, the models being used to inform decisionmaking correspond less well to the (simulated) reality of the context in which the results are applied than in the earlier cases. Note that Blue's use of recursive simulation to inform decisionmaking results in greater Blue losses, but much greater Red losses, compared to the case of neither using recursion. This is understandable since the measure of effectiveness used in the recursive studies was simply loss differential. Red does barely better with recursive simulation help than without it. Perhaps this is because the scenario is weighted in Blue's favor, so that essentially Red has fewer good choices.

Another set of runs was made in which the decisionmaking as well as movement and acquisition in the base level were stochastic. Table 3 summarizes these results. These data are also averages for 100 trajectories. Now we can see that the benefits of recursive simulation are lost. Neither side benefits, as the changes are too small to be significant. Perhaps the deterministic models used for decisionmaking and the stochastic world in which the decisions are made are just too different for the studies used to inform decisionmaking to be meaningful.

Table 3: Stochastic Base Model for Movement, Acquisition, and Decisionmaking with Deterministic Recursion

Losses	Blue	Red
No recursion	195	313
Both use recursion	205	317
Blue only uses recursion	190	303
Red only uses recursion	195	318

If the simulated decisionmakers also use a stochastic representation in their studies, the picture changes. In these cases, each study to inform a reserve commitment decision consisted of six trajectories, which were stochastic for all events (movement, acquisition, acquisition loss, and decisionmaking). Now the models (but not the actual events) are the same for both the simulated world and the simulations that the decisionmakers use for making projections, with the one exception of modeling reserve commitment decisions themselves. The results are shown in Table 4. The "no recursion" entry is the same as above, repeated for convenience of comparison.

Table 4: Stochastic Base Model for Movement, Acquisition and Decisionmaking with Stochastic Recursion

Losses	Blue	Red
No recursion	195	313
Both use recursion	188	304
Blue only uses recursion	196	312
Red only uses recursion	196	312

Even now the results are not significantly different. In fact, they are statistically indistinguishable. It may well be that with the large variability of results three runs for each case was not enough to be useful.

A further set of runs was made in which the simulation runs performed inside the simulation also used recursion, giving three distinct recursive levels. This case, shown in Table 5, is the same as that above except for this extra level of recursion, in which each of the six trajectories used in a study used deterministic modeling at the recursive level. Even so, neither side shows a significant net benefit from the use of recursion. Again, it may be that 6 stochastic runs for a study (3 for each outcome) are too few to give a benefit.

Table 5: Stochastic Base Model for Movement, Acquisition and Decisionmaking with Deterministic Recursion

Losses	Blue	Red
No recursion	195	313
Both use recursion	194	311
Blue only uses recursion	183	298
Red only uses recursion	203	326

In all of these examples, the recursive runs of course took much longer than those which were not. This is due to the use of much more total simulation time. The recursive runs are on the average shorter, since they begin at the decision point rather than with the initial state. There are several things that can be done to make the recursive runs go faster, most of which were not done in this study:

1. Use simpler representation (done for attrition only).
2. Use coarser time resolution.
3. Use aggregation (most applicable to higher levels).
4. Limit recursive studies to fewer decisions.
5. Limit recursive scenarios to a narrower scope (only what is of immediate interest to the decisionmaker).
6. Limit the (simulation) time duration of the studies.

7 CONCLUSIONS

This study has shown that the recursive use of a simulation to improve the representation of Command Control is indeed possible. Indeed, it was not difficult to implement given the accommodations in software structure to support multitrajectory simulation. In some cases, particularly where the models are deterministic, the technique significantly improves the quality of the decisionmaking, as seen by the impact where one side but not the other gained this benefit. Even when the modeling used recursively is different from that at the base level, improvements were significant. However, if stochastic models are used for recursive studies to support decisionmaking in the model, one needs to use more than six replications if there is to be a significant benefit.

ACKNOWLEDGMENTS

The work described here is a further development from earlier research in multitrajectory simulation that was funded by the US Army Research Office under Grants DAAH04-95-1-0350, DAAG55-97-1-0360, and DAAG55-98-1-0451 with the sponsorship of the US Army Center for Army Analysis. Thanks are particularly due to Mr. Gerry Cooper, Col. Andrew Loerch, and Dr. Robert Alexander.

REFERENCES

Gilmer, John B. Jr., and Frederick J. Sullivan. 1998. Alternative implementations of multitrajectory simulation. *Proceedings of the 1998 Winter Simulation Conference,* ed. D.J. Medeiros, F. J. Watson, J. S. Carson and M. S. Manivannan, 865-872. Piscataway, NJ: Institute of Electrical and Electronics Engineers.

AUTHOR BIOGRAPHIES

JOHN B. GILMER JR. worked in the development of combat simulations, with a focus on C2 representation and parallelism, at BDM, Inc. He was the chief designer of the CORBAN combat simulation. He received his Ph.D. in Electrical Engineering from VPI and is Associate Professor of Engineering at Wilkes University. His email address is `<jgilmer@wilkes.edu>` and his web address is `<http://wilkes1.wilkes.edu/~jgilmer>`.

FREDERICK J. SULLIVAN is Associate Professor of Computer Science and Dean of Technology at Wilkes University. He previously taught at Rose-Hulman Institute of Technology and the State University of New York at Binghamton. His expertise is in operating systems and object-oriented programming. He received his Ph.D. in Mathematics from Louisiana State University. His email and web addresses are `<sullivan@wilkes.edu>` and `<http://wilkes1.wilkes.edu/~sullivan>`.