

ISSUES IN EVENT ANALYSIS FOR RECURSIVE SIMULATION

John B. Gilmer Jr.

Division of Engineering and Physics
 84 West South Street
 Wilkes University
 Wilkes-Barre, PA 18766, U.S.A.

Frederick J. Sullivan

Department of Mathematics and Computer Science
 84 West South Street
 Wilkes University
 Wilkes-Barre, PA 18766, U.S.A.

ABSTRACT

Recursive simulation allows decisionmaking entities within a simulation to themselves use simulation as a way of projecting their situation into the future. In these imagined futures, events occur that can significantly affect the entity, and if the information about those events can be captured and related to the entity's present, better decisionmaking may result. This paper explores this concept, and some of the issues that arise, in the context of force on force military simulation.

1 INTRODUCTION

The term "recursive simulation" is used to describe entities represented in a simulation themselves using simulation as a technique to inform their decisionmaking. The simulation used by the decisionmaking entity within the outermost, or "base" simulation run, may be a different simulation from the base simulation, in which case one might refer to the simulation used within as the embedded simulation. Alternatively, the entity might use the same simulation engine as the base simulation, perhaps with different resolution or representational choices, in which case this would be recursive simulation in the strict sense. Figure 1 illustrates the concept as applied to considering alternate Courses Of Action (COAs in the figure).

The techniques and issues addressed in this paper are focused primarily on representing decisionmaking by human entities in military simulations. The context assumed is conventional force on force ground modeling. This is perhaps not the most relevant domain when one considers the nature of current operations, but it has been studied extensively in the past and thus provides a familiar context for exploratory work.

The assumption is that the entities within the base simulation are using simulation to inform their decisionmaking much as a human user of simulation would. This implies a certain structure for using simulation as a decision aid by that entity: One example, perhaps the most obvious, is the use of simulation to evaluate alternative choices. First, an issue requiring a decision is identified. The important potential consequences of the decision are then considered, including the time frame and scope (in geographical terms perhaps) over which the entity is concerned. Appropriate Measures of Effectiveness are chosen to measure the extent to which a given projected outcome is consistent with the entity's goals. Possible choices, in the form of alternative courses of action, are formulated. The entity's understanding of the current situation is then formalized as the initial state of the recursive simulation run scenario. The different courses of action (perhaps including "no decision") then modify each instance of the scenario to create the initial states for each different trajectory (or collection of trajectories) associated with the particular courses of action. The simulation "run" is completed to final states for the different trajectories, which are then evaluated with respect to the Measures of Effectiveness selected. Now back in the base simulation, the decision can be informed by the respective metric results associated with the different courses of action. For example, the course of action having the highest aggregate Measure of Effectiveness would typically be chosen. The base simulation would then continue forward.

Another context in which recursive simulation might be used lies outside the scope of a particular decision. An entity might want to look into the future and project

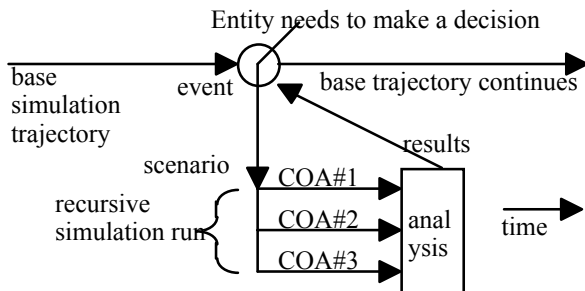


Figure 1: Recursive Simulation Example

whether, given its current course of action, it expects to meet its goals. The operation would be similar to the earlier case except that only the current course of action would be projected. Instead of comparing among alternatives, the comparison is of the projected Measures of Effectiveness against goals. Failure to find satisfactory projections might then trigger the planning and decision process to search for an alternative, or notification and communications to seek additional help or resources.

In the examples described above, the only information conveyed back to the base simulation is the set of values for Measures of Effectiveness associated with each course of action. Yet, often it is not what one might call the Measure of Effectiveness “results” of a simulation exercise that are most informative, but the knowledge gained as part of the process of performing the study. If one looks only at final Measures of Effectiveness, one does not see the cause and effect played out in representative simulation trajectories, and the decisionmaker loses valuable insights that may well be more important than the simulation end states. Human beings are able to do this, because of the versatility of their intelligence. But in recursive simulation, that facility is not available. How can automated assessment of recursive simulation gain for the decisionmaking entity insights that are comparable?

This paper focuses specifically on issues of recognizing and analyzing critical events in these recursive simulation runs. The word “event” here is used in the context of the modeled reality, rather than in the domain of software technique. The underlying simulation executive might be event or time stepped, but the “events” of an “event stepped simulation” would include low level mechanisms such as objects arriving at nodes in a graph representing the geography. In this paper, in contrast, the term “event” is reserved for occurrences having an effect noticeable by the decisionmaking entity, that are possibly governed by chance, and involving one or more entities of interest to the decisionmaker, and which occur at a particular time in a given simulation trajectory. The term “entity” is used for the modeled things (usually military units and terrain regions) in the simulation rather than “object” to emphasize the focus on the modeling domain rather than software domain. (In implementing a model in software, a given entity might be represented by many objects.)

In the past, recursive simulation has been used to inform choices among alternatives, including whether a given decision rule should “fire” or not (Gilmer and Sullivan 2000). The technique of embedded simulation within simulation has been used within JWARS for similar purpose (Argo et al. 2002). The technique of capturing additional information from recursive runs to add to a decisionmaking entity’s understanding of the situation has been explored for a simple “cats chasing mice” context (Agarwal and Gilmer 2004). Analysis of events has been explored for purposes of better selecting representative trajectories in multitrajectory simu-

lation, but not for purposes of enhancing an entity’s understanding of its situation (Gilmer and Sullivan 2001). This paper is intended to look specifically at issues that arise when such event information is to be captured to assist such automated decisionmaking.

The examples used to illustrate the issues raised are drawn from the use of the “eaglet” simulation (Gilmer and Sullivan 1999). This simple force on force simulation models nominally battalion sized units in a manner somewhat similar to, but much simpler than, the “Eagle” simulation. It has been used primarily for research into multitrajectory and recursive simulation techniques. A very small 4 unit scenario is shown in Figure 2 to illustrate the nature of the model. In this sample two Blue units attack a Red unit, and a second Red unit moves up to reinforce when the need for reinforcement is recognized. Several different routes can be taken by the different units as shown. By time = 50, there are 10 different trajectories, having variations in chosen movement paths, attrition experienced, and decisions made by Red to commit Unit 4.

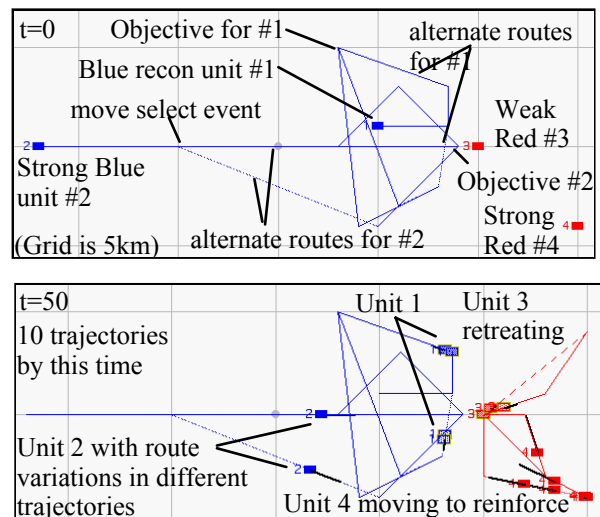


Figure 2: Small “Eaglet” Example

2 SELECTION OF EVENT TYPES TO BE EXPLICITLY REPRESENTED

In earlier work exploring multitrajectory simulation, “events” in the sense addressed in this paper were potentially random choices or occurrences that were explicitly modeled (Gilmer and Sullivan 2001). The mechanism for events was, as a software technique, implemented by a call to a “chooser” object which selected the result and (if desired) recorded the choice made and information about the event.

The analyst could select for events (by class of event, unit, time, or other criteria) to be resolved as a deterministic choice or as a random draw, in which case the trajectory would continue to represent the same fraction of the over-

all run. For a stochastic choice, the probability of the trajectory is multiplied by the probability of the event outcome that occurred, to track the trajectory's probability. (The sum of all trajectories' probabilities could initially be above one if multiple initial states are used, but eventually the sum will become quite small, as the trajectories typically represent only a small fraction of possible outcomes.) Knowing the relative probabilities may be useful. A given event may also be selected in a multitrajectory fashion by creating a new trajectory to receive one possible event outcome while the trajectory encountering the event would receive the other event possible outcome. In this case the assigned probability of each resulting trajectory would be representing the probability of the original trajectory multiplied by the probability of its particular outcome. Mixed management techniques have also been used, such as multitrajectory treatment up to a resource limit, then stochastic beyond that limit for the least probable or important trajectories.

The selection of event types was made under assumptions that these were the most important kinds of events. Those chosen for inclusion in this research are:

1. Attrition events: losses inflicted by an entity on a hostile entity. These losses could be random, that is, drawn from a distribution. The distribution used was discrete. The outcome of the event is noticeable to the decisionmaking entity as (most crudely) a change in the entity's effectiveness, or a change in numbers of some asset below a predetermined threshold, or a change in category when comparing combat strength with an enemy force, or possibly as a value evaluated against loss rate criteria. An example of such an event would be that unit #4 inflicts losses of 2 tanks on unit #7 at time t (representing a given time interval). A random effect might be that the losses might be 1 greater or 1 fewer.
2. Detection events: A entity becomes aware of the presence of (and possibly additional information about) another entity, usually an enemy entity. This might be the culminating event of a more elaborate process that models intelligence gathering, surveillance, acquisition, and related processes, but ultimately the decisionmaker becomes aware of the enemy entity or it doesn't, the latter possibly despite some indications of threat not rising to a level necessary to trigger recognition. A typical event might be modeled as having a probability that unit #4 detects unit #7, given the range and capabilities of unit #4. One event outcome is success, in which case unit #4 adds information about unit #7 to its understanding of the situation, or detection failure, in which case this does not happen.
3. Loss of Detection events: An entity loses awareness of a particular entity, often due to loss of sensory contact due to increasing range, for example, but possibly including destruction of that entity, or loss of acquisition capability on the part of the decisionmaking entity. An example would be that Unit #4 has a representation of unit #7 in its understanding of the situation, but due to increased distance between them as Unit #7 moves away, Unit #4 can no longer consider its most recent information on Unit #7 as meaningful. This likely would be represented by dropping Unit #7 as an entity of which Unit #4 is aware.
4. Decisionmaking events: Assuming that a decision takes place by the "firing" of a decision rule, then the testing of such a rule and the consequent firing or not firing of the rule can be considered an event. Randomness in whether such an event fires could be due to random effects in generating the values of variables cited in the particular rule, or by randomness represented in the rule itself. For example, unit #4, based on detection of Unit #7, might decide to attack. Given the ratio of strength and the situation, this choice may not be obviously consistent with Unit #4's mission, and thus a random draw representing chance variations in what commanders might do in this situation might be used.
5. Movement selection: A particular variation on decisionmaking events is route selection. In following (or constructing) a route from one location to another there may be branch points at which a choice may be made. This choice can be considered an event important to the entity's future, depending on what circumstances surround each choice, even when the ultimate destination is the same.

Beyond this list of examples, one could easily add communications events (whether a message is successfully received or not, or perhaps at the more aggregate level whether communications are maintained). Events to vary movement speed or many other parameters may be appropriate. What events are of sufficient importance to be worth explicitly recognizing; for which modeling the awareness of such an event by the entity may be important enough? It may be that ultimately some facility for recognizing a new type of event is needed.

As a simulation trajectory runs forward in time, it is possible to make a sequential list of all of the events and outcomes that define that particular trajectory. Originally this was used for diagnostic purposes. A particularly interesting trajectory was re-run by having the simulation refer to this record whenever an event occurred rather than making a draw or using some other method of event resolution,

so ensuring a duplicate run. During the rerun graphics might be turned on or additional diagnostic outputs enabled which would be unmanageable for a large number of simultaneous trajectories.

These records of events and outcomes, for a set of trajectories (perhaps recursively initiated by a decisionmaking entity), can be analyzed to understand the importance of particular events to the outcome. An example of such event tracking is shown in Table 1 below for a four unit "eaglet" run for three of 100 trajectories the run. For each of the three trajectories, the table gives the trajectory number, the trajectory probability, Blue and Red losses and Loss Exchange Ratio (the Measures of Effectiveness), the number of events, and a list of the particular events encountered and their outcome. Event types include 5: detection events, 1: movement selection events, and 4: detection loss events. (Attrition events were not modeled as random in this run. Only the first nine events for each trajectory are shown.) In this format for an event list, the other data for each event was not recorded, since the original purpose of the list was to allow a trajectory to be replayed exactly. Events would necessarily occur in the same order given the same choices (either 0 or 1 for each of these events).

Table 1: Example of Event Records for a Simulation Run

t#	p(t)	Blue	Red	LER	event count	type/outcomes
3	0.021	25.2	27.5	1.09	30	51 51 10 10 10 40 50 40 41
19	0.005	29.5	27.8	0.94	32	50 50 50 50 11 11 50 10 50
25	0.007	45.0	34.0	0.76	44	51 50 51 10 11 10 50 50 10

A related issue is whether to manage the outcomes of random events so as to give a greater representation to lower probability trajectories that may be particularly important, or whether to allow outcomes to be determined randomly. One could also manage with an objective of giving greater or more balanced representation to the outcomes of more important events. The management of event outcomes has been studied in earlier research on multitrajectory simulation mentioned earlier, and will not be addressed further here. It is assumed in this paper that event resolution is by random draw, although the approaches and issues discussed are not limited to that case.

3 IDENTIFICATION OF COMMON EVENTS ACROSS TRAJECTORIES

Given event classification and instrumentation, one can obtain a list of the events and their outcomes for each of several trajectories starting from a common scenario but varying either randomly or due to different courses of action. However, how does one recognize that an event is "the same event" in two different trajectories, so that the impact of the event can be compared? One can be stricter or looser in defining correspondence. For example:

- Criterion 1: To be recognized as "the same" event, events in different trajectories must have the same entity identifications (the party or parties to the event, such as shooter and target) and occur at the same time or within some very small delta t.
- Criterion 2: As above, but with the delta t much larger, but such that the sequential occurrence order for the corresponding events in the two trajectories must match.

Consider a simulation run having two trajectories as shown in Figure 3. The two diverge at event 2, which might be, for example, a route selection event. This means that they make a detection trial and fail to detect an enemy unit (events 3 and 4) at slightly different times, and similarly do detect the enemy unit (event 5) at slightly different times. It is reasonable to assume that events labeled 3, 4, and 5 in the two trajectories are "the same event" for purposes of comparing these and perhaps other trajectories. The narrow time difference allows this identification. However, if the results of route selection were skewed enough, the correspondence might be difficult to find. In Figure 4, events of trajectory 2 are so early compared to those of Trajectory 1 that following Criterion 1 would match events e3 and e4 of Trajectory 1 to e4 and e5 of trajectory 2, and would fail to find a match for the other events.

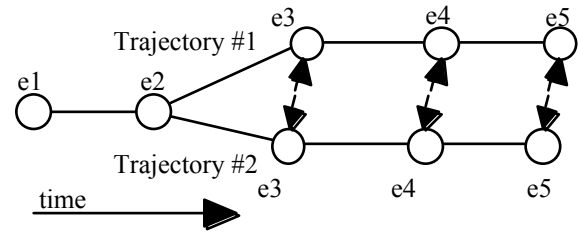


Figure 3: Matching Similar Events with Close Corresponding Times

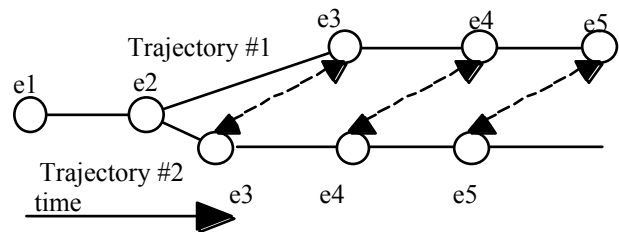


Figure 4: Matching Similar Events with Varying Corresponding Times, Perverse Case

The pitfall of looking for correspondences in order is that this kind of skewing of event trains for particular entities will allow other events to intervene making such matching more difficult, especially when many different trajectories are being considered. In establishing corre-

spondences between events, additional criteria such as location (x,y coordinates for the units) may be added, and may help identify event correspondences and make those identifications more robust.

The ultimate effect of shortfalls in establishing a correspondence between two events in different trajectories that a human would consider to be the same event, is that the opportunity to recognize the significance of that event is lost. One can no longer compare between as many different samples of trajectory end state for each of the different event outcomes. At worst, the list of events for the trajectory set as a whole becomes long and sparse, with few trajectories identified with each event. Lengthening the time window risks wrong correspondences.

Table 2 is a sampling of 8 event records drawn from an "eaglet" run having about 20 entities and 377 trajectories that illustrates the issue. This is a further developed version of the event list mechanism shown earlier in Table 1. Across the full set of trajectories there were 3368 different events after correspondences were established, but each trajectory had only a small number, about 300 typically. Event correspondence criteria were fairly tight. Initially, all trajectories encountered the same events (events numbers 1 to 7) which were initial condition variations, and each had either outcome 1 or outcome 2. But subsequently, many events were not identified in many of the trajectories. For example, for the sample shown, event 8 occurred only in trajectory 2, where it had outcome 1. Events 27 to 29 did not occur in any of these 8 trajectories, and as time evolved the matrix gets sparser. Many events occurred in only a few trajectories, and often only one of the two or more possible outcomes were represented. (The event descriptors that gave the other data pertaining to the events, such as time, unit, event type, and such were a separate listing.)

Table 2: Example of Event Records (20 Unit Scenario: 377 States, 3368 Events)

Trajectory, event number and (in columns beneath) outcomes for events 1 to 31
1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
2 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
0 1 1 1 1 1 1 1 0 1 1 0 0 0 1 0 1 1 0 1 2 2 2 1 0 1 1 0 0 0 1 0
1 2 1 1 1 1 1 1 0 1 1 0 0 0 1 0 1 1 0 1 2 2 2 1 0 1 1 0 0 0 1 0
2 1 2 1 1 1 1 1 1 1 1 0 0 1 0 0 1 1 1 1 0 0 2 1 1 0 0 0 0 0 1 0
3 1 1 2 1 1 1 1 0 0 1 0 0 0 1 1 1 1 0 0 2 2 2 0 0 1 1 0 0 0 1 0
4 1 1 1 2 1 1 1 0 1 1 0 0 0 1 0 1 1 0 1 2 2 2 1 0 1 1 0 0 0 1 0
5 1 1 1 1 2 1 1 0 1 0 1 0 0 1 0 1 1 0 1 2 2 2 1 0 1 1 0 0 0 0 0
6 1 1 1 1 1 2 1 0 1 1 0 1 0 1 0 0 1 0 1 2 2 2 1 0 1 1 0 0 0 1 0
7 1 1 1 1 1 1 2 0 1 1 0 0 0 1 0 1 1 0 1 2 2 2 1 0 1 1 0 0 0 1 0

One fallback possibility is that instead of focusing analysis on the impact of individual events, the impact of classes of events can be assessed. For example, all detection attempts by unit #4 against unit #7 could be lumped into one pool. This risks losing an important impact of one

particular event in a pool that extends over a larger time and area. It would be possible to restrict such pooling to relatively low frequency events that otherwise could not be analyzed at all. This more flexible approach has not been explored.

A related issue in the software domain is whether to construct an event list as shown in the table, finding correspondences "on the fly" as the simulation runs, or whether to do this as postprocessing. The former approach is computationally expensive but saves memory; the latter requires an enormous amount of storage, potentially much greater than the trajectory states themselves. Performing "on the fly" event analysis has the advantage of being able to feed back useful information to the decisionmaking entity asynchronously, that is, before the analysis is completed, which may be useful if a very important event is discovered, and if the base simulation is running in real time (for example, in a training simulation). It can also be useful in guiding the management of random events as the trajectory set is developed. This kind of "on the fly" processing may be done whether the trajectories are simultaneous or sequential, but the correspondences may be different in these cases unless the correspondence algorithm is carefully constructed to be processing order independent.

4 EVENT IMPACT ASSESSMENT

Given that one has identified particular events in a set of trajectories, how should one measure the importance of the event? The most obvious approach is to use the same Measure of Effectiveness that is being used for assessing the outcome of a course of action, and comparing the value of that for the group of trajectories in groups of trajectories having each outcome. For example, if Loss Exchange Ratio is the single MOE being considered, and an event results in an average MOE of .1 if the event had an outcome of "1" and .3 if the event had an outcome of "2", then this would seemingly be a very important event. If there are multiple Measures of Effectiveness, one can weigh them and form a vector, and then measure the distance between the average vectors for different sets of trajectories. Trajectories which do not include the event could be ignored or treated as having an intermediate outcome for the event, if that makes sense.

It may appear that actually measuring distances between final states might be desirable. This has been done for modest sized scenarios. But this will likely be impractical for most larger cases for several reasons. Some of the entity memberships might be different, so that for some state variables there are no comparable state variables in another final state. (This was not an issue in "eaglet" since the entity membership was static.) Not all state variables are of equal significance, and it may not be apparent how they should be normalized. Should the importances vary with the event type? (In "eaglet" the importances were

predetermined and static.) Much of the state of a simulation of the sort considered is relational, for example what entities can see which others, and this will resist straightforward comparison. Finally, the number of state variables can be very large. In experiments conducted with the very simple simulation “eaglet,” analysis required consideration of 20 state variables per unit and ignored the relational variables. Both the computational and memory burden of processing larger state spaces would be burdensome for more realistically sized recursive runs.

Intermediate between using just one or a few Measures of Effectiveness and using the entire final state space is to use a large set of aggregate Measures of Effectiveness. This might include indicators of whether there might be a large spatial difference between the final state sets for different event outcomes. For example, the first two or three moments for the distribution of an aggregate force metric could be used. If one assumes that a tank counts as two other armored vehicles, the center of mass of the tank equivalent strength in both x and y dimensions could be found, and the standard deviations of the distributions as well. If under a different event outcome these metrics are significantly different, then it is apparent that something noticeable in a spatial sense has occurred even if it does not appear to have an effect in the Loss Exchange Ratio metric.

Another issue is the nature of the Measures of Effectiveness themselves. Should they be focused on the particular decisionmaker, the parent organization and its mission, or the overall assessment of the outcome of the battle for that side? Ideally, it would be the aggregate well being of the side that would be most important. But humans do tend to focus on more local self and organization centered goals. To what extent should this tendency be modeled? How would one establish a balance in evaluating these competing interests, and how would one validate it? In work done within “eaglet,” the MOEs used were for the side as a whole, but other more self centered metrics, or sides using different balances of this issue, would make an interesting study.

Ultimately the issue is whether the resolution of measuring the event impact assessment is consistent with the numbers of samples which occur in the recursive trajectory sets. If one makes recursive runs with only two trajectories, one for each of two courses of action, event analysis does not even make sense because the contexts will be quite different, and few events will occur in both trajectories. If scores of trajectories are used for each course of action, then very likely some significant variation within the sets will be due to some of the more important events, and this would be observable against a few Measures of Effectiveness. Recall that for our sample of 377 trajectories for 20 units, a typical event occurred in only about 10% of the trajectories. If there are hundreds or more of trajectories, then one might discern more subtlety in out-

comes worthy of examining with respect to more measures of effectiveness. It comes down to an issue of affordability of the computational effort involved versus the benefits in fidelity gained in representing decisionmaking, a tradeoff that cannot yet be quantified in research that has been conducted.

Table 3 shows a list of event identifications and importances as determined using three different metrics. In this case the events were reorganized into classes, lumping similar events having different times together, as mentioned earlier. This particular set of data was based on comparing entire state outcomes rather than just MOEs.

Table 3: Example of Event Records (20 Unit Scenario: 377 States, 3368 Events)

En	Ty	U	D	Cnt	Metr1	Metr2	Metr3
0	1	4	13	995	.0037	.0081	.0022
1	1	4	17	407	.0198	.0080	.0022
2	1	4	21	336	.0038	.0044	.0020
3	1	8	25	995	.0044	.0078	.0022
4	1	8	26	559	.0038	.0077	.0021
5	1	8	27	423	.0179	.0091	.0023
6	1	9	31	995	.0048	.0077	.0022
7	1	9	32	376	.0138	.0088	.0020
8	1	9	33	606	.0066	.0090	.0022
9	1	10	37	995	.0044	.0078	.0022
10	1	10	45	569	.0080	.0049	.0020
11	1	14	49	995	.0044	.0078	.0022
12	1	14	50	567	.0055	.0089	.0021
13	1	14	51	419	.0074	.0084	.0023
14	1	15	55	995	.0044	.0078	.0022
15	1	15	56	559	.0058	.0082	.0021
16	1	15	57	421	.0032	.0064	.0023
17	1	16	61	995	.0044	.0078	.0022
18	1	16	65	344	.0072	.0062	.0021
19	1	23	73	168	.0020	.0028	.0022
20	1	23	77	723	.0069	.0047	.0021
21	1	29	81	5	.0000	.0000	.0021
22	1	29	86	527	.0015	.0028	.0021
23	1	34	91	861	.0074	.0045	.0021
24	1	34	96	3	.0000	.0000	.0038
25	4	4	23	156	.0030	.0020	.0024
26	4	4	24	661	.0025	.0034	.0020
27	4	4	27	30	.0000	.0000	.0037
28	4	4	28	242	.0181	.0088	.0022
29	4	4	29	6	.0000	.0000	.0035
30	4	4	32	997	.0156	.0060	.0022
31	4	4	34	154	.0040	.0024	.0024
32	4	4	36	20	.0000	.0000	.0028

(followed by the other event set data)

En: event number Ty: event type

U: Unit experiencing the event D: Other data

This table was constructed from a scenario of about 20 units (some unit ID's are vacant; only the first few score events are included here). The columns identify the event number, type of event (1 is movement selection; 4 is acquisition), the entity or unit, data associated with the event such as route node or target unit, the number of trajectories in which the event occurred, and normalized metric values (out of a total of 1 over all events).

Metric 1 in the list is the average effect method described above. Metric 2 identifies events which cause the final state vectors to be more nearly orthogonal to others. Angles are measured between a state and that of states having the opposite outcome, and the normalized sum of the absolute values of the sines is summed. This was intended to indicate the sense in which an event tends to expand the number of dimensions in the outcome space. Metric 3 measures the average of minimum distances between states with the nondefault outcome and other states. Other metrics have also been explored (Gilmer and Sullivan, 2001). Some metrics proved better than others; they emphasized different qualities in the final states. For example, Metric 3 does not downgrade importance events for which there are few samples, but tends to show less discrimination. The purpose needs to drive the choice of metrics used. Does one seek maximum effect in moving the outcome space, or in being distinctly different from other outcomes? Each may have practical uses.

5 USING THE INFORMATION

Given the identification of certain important events, perhaps as a prioritized list, how is the decisionmaking entity to make use of this information? As mentioned before, one can just ignore event information and just look at the final Measures of Effectiveness. In another method of extracting information that has been called "Circumstance Descriptors," certain role identifications in the future can be recognized and those identifications brought back to the present. In the simple cats chasing mice example, the identification of another cat competing to catch the same mouse can be converted from the recursive run back to the cat's understanding in the base trajectory, affecting the decision of the cat on whether to chase that particular mouse or perhaps even whether to attack the competing cat (Agarwal and Gilmer, 2004). This can be thought of as being event related (discovery of the competing cat) but it is the availability of an existing mechanism that effectively searches for an entity to fill a role (competing cat) that enables this. We would like a more general method that would recognize the events themselves. This would allow the entity to act on the event itself, rather than on just the related information (the entity that is a threat).

One approach is to make the event itself an object (in the software sense) or entity (in the modeling sense). In the present (in the base simulation trajectory) of the con-

cerned entity (the one whose future is being examined), the event is a possible future event that occurs in one or more of the recursive trajectories, and has an important effect. The different outcomes would be associated with other entities that would (presumably) already be entities in the simulation, such as two paths leaving a road junction, or an enemy unit that might or might not be sighted. The association between the event outcome and the impact of the event needs to be established. For example, if not sighting the enemy unit is associated with a good outcome (favorable MOE impact), then one might want to make a decision to ignore that enemy unit even if it were detected. It might be a decoy or a distraction, and is perhaps some other unit's problem. Thus, if the decisionmaking entity maintains a list of enemy units and what it knows about them, this unit could be marked in this way. (The reader may wonder how the unit could show up in the recursive runs if the entity does not already know about it. In this case we assume that the enemy unit would have been mentioned in the intelligence section of an operation plan or in other intelligence passed from another friendly entity. When the decisionmaker constructs the scenario, the enemy entity would thus be present, but in an initial state reflecting the current lack of direct detection.)

In a model of intelligence more sophisticated than those used by the authors, the event and associated information could fit into a semantic network modeling all that the decisionmaker knows (and projects) about the simulated world, and could then be used by more general purpose tools. This requires an explicit structure that reflects beliefs about alternative paths into the future, because the event may possibly not occur, and the different outcomes and their consequences will need to be characterized as random. Ultimately, the goal is to affect the probabilities by actions taken in the present. Doing this will be quite a challenge, but with a mechanism to perform recursive simulation and collect data about future events, it may be possible to begin exploring this possibility.

6 CONCLUSIONS

The techniques discussed here have been explored by the authors somewhat piecemeal while pursuing other objectives, such as exploring ways to tailor event outcomes to produce a more useful or representative set of trajectories. Recursive simulation and Circumstance Descriptors have not yet been applied to reasoning about events. Exploring fully the issues described here is a long range goal of ongoing research.

ACKNOWLEDGMENTS

Much of the work that led to formulating the issues and understandings of them presented here is derived from projects 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, particularly Mr. Gerry Cooper, Col. Andrew Loerch, and Dr. Robert Alexander. The authors are indebted to these persons for their encouragement and support.

REFERENCES

- Agarwal, R. and J. B. Gilmer, Jr. 2004. Back from the future: using projected futures from recursive simulation to aid decisionmaking in the present. *Military, Government, and Aerospace Simulation*, ed. Kevin J. Greavey. 61-64. San Diego, California. The Society for Modeling and Simulation International.
- Argo, H., J. Huynh, D. MacQueen, M. Grell, H. R. Blacksten, P. Bross, J. Prince, and C. Burdick. 2002. Plan building in JWARS. 421-430. *Proceedings, 11th Conference on Computer Generated Forces and Behavioral Representation*.
- Eagle combat model, TRADOC Analysis Command--Operations Analysis Center Fort Leavenworth, Kansas, Undated but prior to 1995. This briefing document was used as a reference describing Eagle when the simpler simulation "eaglet" was designed
- Gilmer, J. B., and F. J. Sullivan. 2000. Multitrajectory simulation performance for varying scenario sizes. *Proceedings of the 1999 Winter Simulation Conference*, ed Jeffrey A. Joines, Russell R. Barton, Keebom Kang, Paul A. Fishwick. 958-963. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.
- Gilmer, J. B., and F. J. Sullivan. 2000. Recursive simulation to aid models of decisionmaking. *Proceedings of the 2000 Winter Simulation Conference*, ed. J. A. Joines, R. R. Barton, K. Kang and P. A. Fishwick. 958-963. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.
- Gilmer, J. 2000. Circumstance descriptors: a method for generating plan modifications and fragmentary orders. *Proceedings of the 2000 Winter Simulation Conference*, ed. J. A. Joines, R. R. Barton, K. Kang and P. A. Fishwick. 303-311. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.
- Gilmer, J. and F. Sullivan. 2001. Combat simulation trajectory management to enhance outcome space variety, final report made to Army Research Office. Wilkes-Barre, Pennsylvania. Wilkes University.
- Gilmer, J. and F. Sullivan. 2001. Event importance analysis for military simulations. *Proceedings, World Multiconference on Systems, Cybernetics, and Informatics*, ed. N. Callaos, Q Guo, S. Pierre, S. Muravyov. Vol. XVI, 39-44. International Institute for Informatics and Systemics.

AUTHOR BIOGRAPHIES

JOHN B. GILMER JR. teaches electrical and computer engineering at Wilkes His research interests include military simulation with a particular focus on modeling decisionmaking. His e-mail address is jgilmer@wilkes.edu.

FREDERICK J. SULLIVAN is an associate professor of mathematics and computer science at Wilkes University. His research interests include simulation, computer operating systems and programming language. His e-mail address is sullivan@mathcs.wilkes.edu and his Web address is www.mathcs.wilkes.edu/~sullivan.