Proceedings of the 2008 Winter Simulation Conference
S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler eds.

ASSIGNMENT OF PROBABILITIES TO EVENTS FOR COMBAT SIMULATION

John B. Gilmer Jr.

Division of Engineering and Physics
Wilkes University
Wilkes-Barre, PA 18766, USA

Frederick J. Sullivan

Dept. of Mathematics and Computer Science
Wilkes University
Wilkes-Barre, PA 18766, USA

ABSTRACT

Multitrajectory simulation allows explicit management of random events by allowing particular events to be resolved by random draw, by a deterministic choice, or by creating new states to allow following multiple trajectories. The policy for resolution method is under the control of the analyst, and may depend on event type, trajectory probability, or even some metric indicating the trajectory importance. However, taking advantage of the technique for exploring possible outcome spaces requires a probabilistic modeling of events that, in simulations of ground combat, are often treated as deterministic, such as decisionmaking. Even for events such as attrition, which have long been modeled as stochastic, how should the full event outcome set be sampled if one is only to keep two or three samples? This paper explores these issues with the goal of outlining what kinds of data would be needed to fully exploit multitrajectory methods in a combat simulation context.

1 BACKGROUND

Combat is inherently stochastic, with many types of random events influencing the details of how the simulated reality develops and the ultimate outcome. Multitrajectory simulation is a technique that focuses on these random events, and attempts to manage the representation and operational treatment of randomness in a way that will better achieve the analytic objectives of simulation rather than simply making random draws. The basic approach is that when a random event occurs, the state trajectory through time is "split" or "cloned" at the event, and each of the resulting states represents a different possible outcome of the event, as illustrated in Figure 1. With additional events, further bifurcations of the trajectories through time occur. With any large number of events the number of potential trajectories becomes enormous. So, it is necessary as a practical matter to choose which trajectories to continue simulating, which to truncate as not worth further development, and which to assign as being so similar to another trajectory that the other trajectory can represent them. In multitrajectory simulation, these decisions on the management of the trajectories can be made based on criteria of importance to the analyst using the simulation. For example, trajectories that are more probable, those that by some metric seem to be more important, and ones that are in some sense very different from others, may be more worthy of continuation.

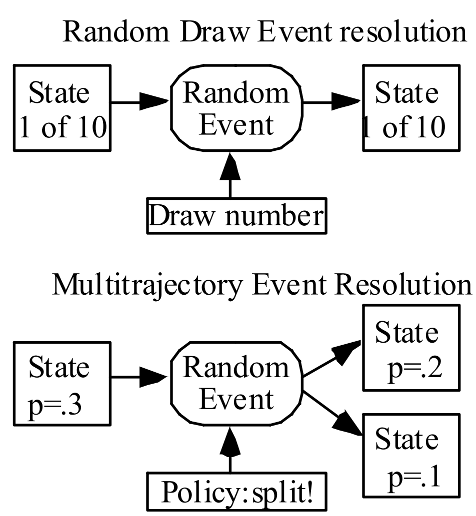

Figure 1: Illustration of Event Resolution Alternatives

The multitrajectory simulation methods and issues have been explored in the past using a simple simulation of ground combat, "eaglet", that was designed to be similar in many respects to the U.S. Army's "Eagle" simulation, but very much simpler. Different ways of implementing the multitrajectory mechanism were explored, and there were developments of ways to analyze the resulting state spaces, the event importances, and criteria for making multitrajectory choices.

One of the important issues that was apparent early in the development of multitrajectory simulation is that many events that are in fact random in some sense are often modeled using deterministic methods. For example, in the modeling of command decisionmaking, it is common to use a "decision rule" that has a "guard" and a “predicate.” The “guard” expression (usually a Boolean algebra function) is evaluated to determine if the rule will "fire." If the guard is “true,” the “predicate” defines what actions are taken. As described, this is a deterministic mechanism. There may be stochastic elements to the input if, for example, one of the logical conditions might be the detection of an enemy force, and that detection mechanism is stochastic. But one would actually expect randomness in the rule firing itself as well.

The purpose of this paper is to review various types of combat simulation events that may be or ought to be stochastic, and describe stochastic ways to model those events, and suggest ways to develop the probabilities that would need to be assigned to each possible outcome. This goes beyond the existing materials already published concerning "eaglet" in which probabilities were assumed to have been developed a-priori as inputs in the scenario data. In "eaglet" only a small number of explicitly stochastic or multitrajectory were implemented: movement selection, detection / acquisition of contact with enemy units, loss of acquisition, decision rule firing, and attrition. This paper will focus on those events, and attempt to more fully examine how these probabilities might be developed, with the assumption that similar methods will be applicable to other types of events as well.

Long ago, in the mid 1980’s, a session including experts in Red doctrine and simulation developers attempted to develop command decision rules that could be used in the CORBAN simulation. There was a disconnect between what the simulation designer expected, rules that could be expressed as Boolean functions, and the way the experts expressed Red’s decisionmaking. The same is likely to be true again if multitrajectory simulation is used in analytic grade work: there will be a need to assign probabilities for processes that have long been treated as deterministic or as having a continuous distribution, both of which are problems to the multitrajectory method. The authors do not pretend to have the expertise to develop analytic grade algorithms. But in presenting the methods described in this paper, we hope that potential users of the multitrajectory method will have a better understanding of some of the data development that would be required.

2 MOVEMENT SELECTION EVENTS

These events pertain to path, or route, planning. The "eaglet" simulation used a link / node network to represent routes, as shown in Figure 2. In a traditional simulation a "route" for a unit to follow would consist of a series of links between nodes, with the first node representing the unit's current (or assigned) location, and the final node an assigned objective. Intermediate nodes would represent waypoints, for example road intersections, that help define the path to be taken. In simulations having a regular pattern tiling the terrain space, such as a square or hexagonal grid, the centers of the hexagons can be thought of as potential nodes and lines connecting adjacent tiles as potential links. In simulations that do not employ a regular tiling, some characterization of the terrain to establish "no go" and "go" areas, potential paths, and the nodes they connect, is necessary. Whether that is done by massive preprocessing of the terrain into a link / node map applicable to all units ahead of time, or whether similar processing is done locally for each unit when route planning for that unit is necessary, will depend on the particular simulation. Either way, some dynamic aspects of route choice may affect scoring and choice probabilities due to possible observation (conditional on the known locations of enemy forces) and the nature of the mission.

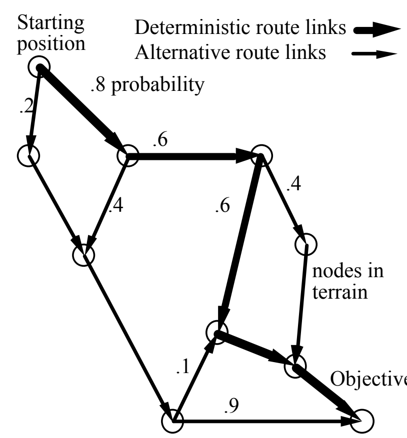

Figure 2: Route network for ground movement

Route / path planning is a complex subject applicable not just to simulations of the sort of interest, but also to robotics. It is beyond the scope of this paper to examine how the link / node representation is developed. But once such a representation exists, route planning is a matter of examining possible paths through the network, and choosing one. Or, for a multitrajectory simulation, choosing a subset of the link / node network that represents all possible paths that a unit might reasonably choose in reaching an objective.

In a conventional simulation, planning a route is typically deterministic. For example, in CORBAN a tree of paths was developed, with cumulative scoring of the advantages of each potential route as it developed, and routes which violated certain constraints (such as taking a unit farther from its objective) abandoned. The score for each link depended on the potential trafficability of the link, and the extent to which that link would bring the unit closer to its objective. Ultimately, the route with the highest score was chosen. While CORBAN used a hexagonal grid, the method is just as applicable to other types of terrain networks, and is very flexible in the constraints and scoring weights to be assigned to different attributes. For example, scoring components may be made to depend on the nature of the operation being conducted. Is cover or high trafficability more important? Is staying close to the center axis of the line between present location and the objective important? Then the scoring algorithm can reflect that.

A stochastic mechanism for movement path selection would assign the highest probability to the path having the highest score, and lower probabilities to other paths which are at least somewhat close to that path in their score. Paths with much lower scores would be discarded; it is assumed that a planner would not seriously consider those paths. For a multitrajectory simulation, it would seemingly be desirable to represent not just one path or a set of discrete paths, but all of the possibilities as if they represent a lattice of possibilities with probabilities associated with the branch points, as shown in Figure 2 above. This kind of representation is useful in the case of operations (such as for enemy units) where the events to take one link or the other at each node represents a hypothesis that can be tested against observations.

The simplest way of assigning probabilities to different paths would be to assign weights proportional to the scores. However, this will likely lead to fairly uniform distributions over the relatively few paths that pass the threshold test. Yet, it is assumed human decisionmakers would be rather sensitive to small advantages, so that a high scoring path would likely have a disproportionately larger probability than its score. One could subtract some constant, say a proportion of the lowest score considered, to correct for this, but then the algorithm becomes more dependent on the score of the suboptimum path, which could be quite sensitive to the threshold effects in constructing the path. Thus, there are several arbitrary constants that would be needed. The algorithm described below is thought superior (though more complex) in requiring only one such arbitrary constant (referred to as "sigma").

The "deterministic" path is presumably that which would have achieved the highest score during a search algorithm to find possible paths. The other paths that traverse from initial position to objective represent routes would have lower scores. If one simply scored paths, a method for establishing probabilities for the different routes would be to assume that the highest score is the mean of some distribution, and that other routes are skewed toward one side or another. Without having to track explicit sidedness, it is assumed that we plot these lower scored paths to the left, as shown in Figure 3 below. This initial example assumes only two paths satisfy some threshold for consideration. This method satisfies the "reasonableness" criterion that as the two scores get close, the probabilities approach 50%. Experimentation and input from subject matter experts would be needed to establish the parameters of the probability density function to be used. Here we assume it is normal

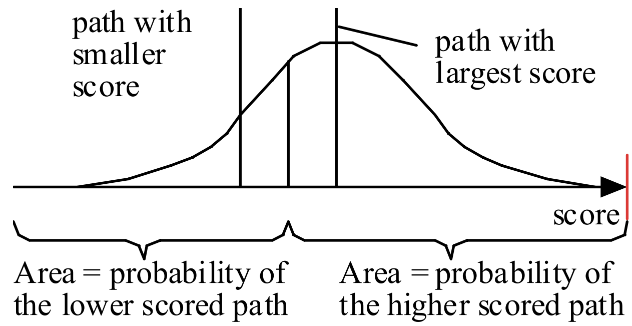

Figure 3: Example calculated probabilities for two paths

Things become more complicated if there are more than two viable paths. One would like for a three path case with nearly identical scores to have a distribution that is nearly even. This can be achieved by again making the distribution mean the score of the best path, but then transforming the locations of the other path scores to put the next lowest to the left scaled down so that an equal score would result in a probability of 1/3 if the score is nearly identical to that of the largest score. This can be accomplished with a normal distribution be subtracting .86 sigma from the second highest score. (The sigma would be the same one used in the two path case.) Figure 4 illustrates this case. The third score would be placed to the right by an equivalent transformation. Thus, for the score nearly equal to the maximum the criterion (mid-point between the maximum choice and this second choice) would be at .43 sigma, giving the (slightly) lower valued scored path a .33 probability.

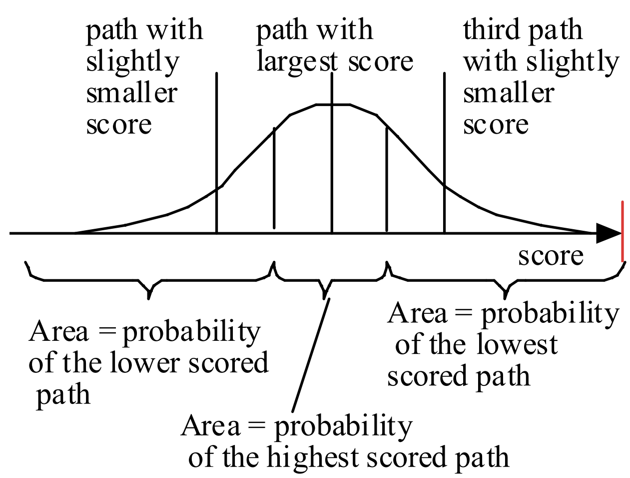

Figure 4: Example of probabilities for three paths

If the second score is significantly lower, it would be farther to the left leaving it with a yet smaller probability, and the maximum score path with a larger probability. Likewise, the third place score would be transformed by adding twice the difference from the maximum score (to give it a value above the maximum) and adding an additional .86 sigma, placing it far enough to the right of the maximum score in the distribution to give it a maximum probability of .33 when the)original) scores were nearly equal. Figure 5 below shows the case where the scores are significantly different.

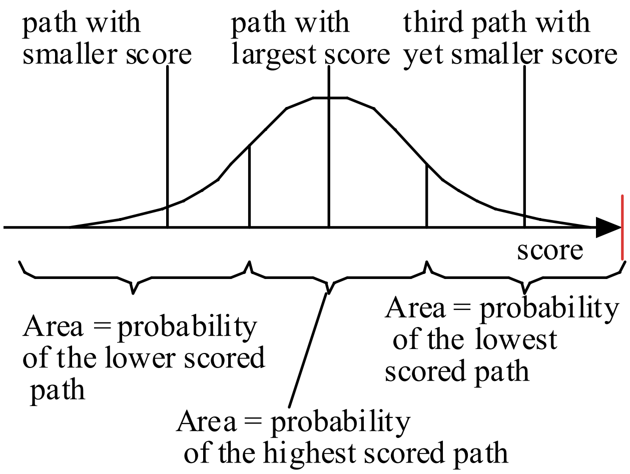

Figure 5: Example paths with three much different scores

If there are four paths worth considering, the method could be extended by placing the best path at the middle, the second place path to the left as for the three path case, and sharing the upper tail of the distribution between the other two paths. The specific score transformations to do so have not been derived, but would be similar in principal to the three path case. Ultimately, then, the probabilities of the paths need to be transformed into a set of probabilities associated with each of the nodes.

The following example is given in Figures 6 and 7 for how this approach might work. The figure shows a group of links and nodes of potential paths from an initial location A to the unit's objective F. The "score" for a given link is simply the length of its component along the direct axis from initial position to the objective, that is, along the dashed line, divided by the actual length. (Links that deviated from the direct path from any given point by more than 60 degrees had been discarded as too indirect.) So, the entire paths that are competing would be A-D-E-F, A-B-C-F, and A-B-E-F.

As it turns out, all have the same score (due to the regular array and flat scoring) of .904. We assume that a Gaussian distribution is used with a sigma of .5. Then A-D-E-F would correspond to normalized Gaussian random variable value of 0, A-B-C-F would correspond to a score of -.86, and A-B-E-F to +.86. Taking the mid-points (bounds on integration interval for the three regions) at -.43 and +.43, we get a probability of .333 for A-D-E-F and the same for the other two choices. At point A, since paths having a total probability of .67 go right, that choice would have a .67 probability and the choice for link A-D would have a .33 probability. At point B, since paths having equal probabilities go in different directions, each choice has a .5 probability. (The other nodes, D, E, and C, have only one deterministic choice; there is no event associated with that node.)

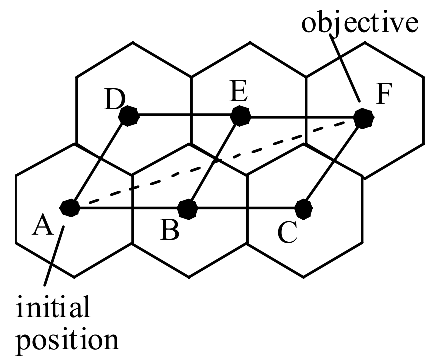

Figure 6: Example of paths found in hexagonal grid

A less regular and more complex example is shown in Figure 7. Here the viable paths scored using the same scoring method as above are A-D-E (high score of .954), A-B-C-E (score of .878), A-D-C-E (score of .877), A-B-D-E (score .787), and A-B-D-C-E (score .708). If we limit consideration to the highest three scores, then A-D-E corresponds to the normalized Gaussian outcome of 0, A-B-C-E to a normalized Gaussian -1.012, and A-B-D-E (with about the same probability) to +1.014 This puts the bounds for area determination at -.506 and +.507, giving a .39 probability for A-D-E and .31 for the other two. With a sigma of .2 instead of .5, the respective normalized values are 0, -1.24, and +1.25 yielding probabilities of .47, .27, and .27 for the three paths. Translating this into node probabilities, at A the probability of selecting A-D is .73 and A-B is .27. Since none of the three paths uses B-D, node B has a deterministic choice of B-C. At node D the link D-E has a .64 probability, and D-C has a .36 probability. Node C is deterministic to E.

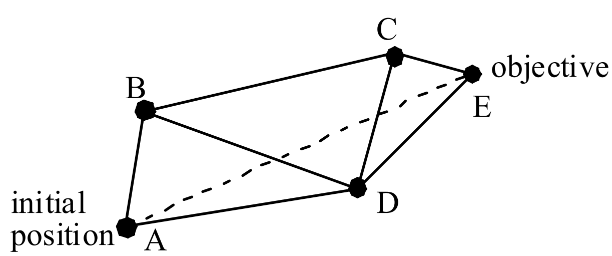

Figure 7: Example of paths found in continuous space

A different way to calculate probabilities is to assume that the path scoring and probabilities can be calculated together working toward the objective, so that at the first node from which two paths reach the objective, a score can be determined for each next link, and a probability calculated as described above. Then the probabilities are calculated without looking farther. That is, links rather than complete paths are scored. The algorithm then moves back to the next node, and is repeated. This has the advantage that the number of alternatives being considered at any one time is the number of links leaving a node, almost certain to be much smaller than the total number of paths, and usually two or maybe three. The three path case can even be subsumed by considering a three way path bifurcation to be a pair of two way bifurcations with the two nodes very nearly in the same location, and the second node along the most probable path. For the A-E network above, at A the scores for A-D and A-B are .968 and .551 respectively. With a sigma of .5 this 2 way choice puts A-D at a normalized value of 0 and A-B at a normalized value of -.834, giving probabilities of .66 for A-D and .34 for A-B. At B the links B-C and B-D score .994 and .893 giving probabilities of .54 and .46 respectively. At D link D-E is direct with a score of 1.0, and D-C gets a score of .875. This yields probabilities of .56 and .44. This method is obviously inferior in the sense that it determines probabilities without looking ahead. However, none of the score to probability conversions involved more than two choices, and all of the paths are possible, not just the top three. Clearly improvements on this method are possible.

Perhaps there are other and better ways to transform a set of scores associated with choices into probabilities. The algorithms described here are intended to be representative, to demonstrate that some such method is possible and it should be fairly reasonable in terms of the probabilities one would expect.

3 ACQUISITION PROBABILITY

Acquisition is the representation of the ability of the simulated command element to become aware of particular enemy forces. Thus, it represents quite a variety of sensor, fusion, and cognitive processes which might be modeled from physical principles but in aggregate are more difficult. Yet, the reduction of this complex process to a random draw (or worse, a deterministic process) has been common in the past, and a necessary simplification in aggregated combat models.

In "eaglet" acquisition is a time stepped process, with units making trials against enemy units at regular intervals (nominally 5 minutes). Figure 8 shows the way acquisition worked. Within some radius acquisition is certain, and outside some radius it always fails, and in between there is a random chance. Once units are acquired, they stay acquired until acquisition is lost by the acquisition loss event. The radii are scaled by a factor that depends on the type of operation the unit is conducting. Note that this was intended to be a very simple representation in keeping with the prototype nature of "eaglet". Algorithms for unit detection, and recognition of a unit given detection of various signatures, was beyond the scope of "eaglet" development. Indeed, the "eaglet" algorithm is "memoryless." Detection trials are repeated at regular intervals until detection is achieved (or the range becomes too large), where a more realistic treatment would reflect that acquisition depends on a more complex process. In reality a failure to recognize a unit given a certain signature pattern might result in subsequent acquisition probabilities of zero, until the signature changes such that a new attempt is made to recognize it.

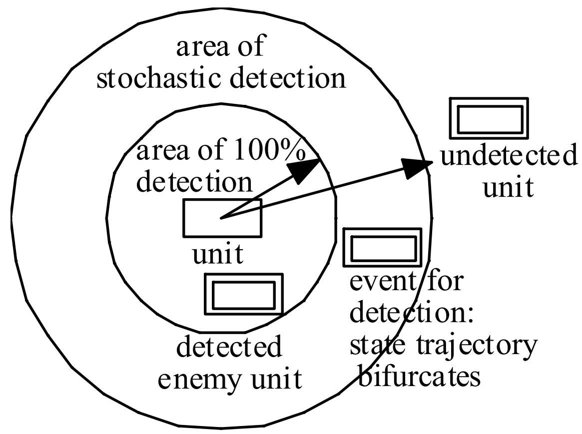

Figure 8: Acquisition events in "eaglet"

In this simple memoryless model, then, the issue is what probability to associate with different ranges, and other factors such as the operation being conducted by the target, the cover it may have, and potentially a host of other factors. Ultimately this is the kind of probability that, on a per trial basis, relates back to sensor on target data. One cannot assume that trials against the individuals making up a target unit are necessarily independent. It should be possible to derive from data and reasonable assumptions unit versus unit detection probabilities for individual trials. If there is any one process that ought to be modeled as a stochastic process in a combat simulation, this is probably it. (Note that the probabilities derived for a memoryless acquisition process need to depend on the time interval between trials.)

A more complex treatment might explicitly model acquisition of signatures sufficient to recognize a unit, and then as a second step recognition of the meaning of the signatures as unit acquisition. This would seemingly be straightforward, based on probabilistic principles and data from field tests and subject matter experts. It eliminates the worst aspects of the memoryless model in the simplest manner.

An alternative would be to divide acquisition into two events: whether an enemy unit is acquired or not, and the time needed to acquire it. If the enemy unit moves to exit the range of acquisition prior to the time to detect, then it is not detected. This method is probably more convenient if the underlying model software is organized on an event stepped rather than time stepped basis. (The relative advantages and disadvantages of each of these ways of organizing a combat model are outside the scope of this paper.) The event to detect would be derived from basic data using field test data and probabilistic principles. The time to detect would be based on process issues, and likely would be imagined as some sort of Gaussian distribution around some mean time.

Here is where a very sticky issue arises. In a normal stochastic simulation, a random choice taken from a continuous distribution is no less convenient than a choice taken from a discrete distribution, and the number of possible choices in the latter case has no particular importance. However, if the event is to be treated in multitrajectory fashion, then the distinction becomes an important resource issue. Each discrete choice results in a state using memory and processing resources. Multiple events with many choices compounds this issue rapidly. A continuous stochastic choice is a better representation than the discrete choice set possible with the multitrajectory method, but can reflect the full scope of possibilities only with a large number of trials. A multitrajectory treatment gives perhaps a better or more representative collection of possible outcomes of several events with fewer trials, but individually cannot represent all of the possibilities. If the event is not very important in its effect on the ultimate outcome, it may even make sense to leave it as a deterministic event using the mean value. (This issue applies to many other kinds of events as well, especially attrition.)

In such a case, we would like a discrete distribution to substitute for the original "correct" continuous distribution. In a sense, this becomes an intermediate compromise between a "deterministic" mechanism and a full, continuous, stochastic mechanism. The issue is how far to go in either direction. One would like to choose probabilities such that the resulting discrete distribution has the same mean and sigma as the original continuous distribution. That means that if a two outcome discrete distribution is used, the two outcomes should each be at plus sigma and minus sigma from the mean, and be equally probable. (If the original distribution is not symmetric, the two outcomes should not have equal probabilities, so that the skewness is correct.) The problem with this is that neither outcome represents the mean. A three outcome discrete distribution giving the mean, and the plus two sigma and minus two sigma choices is another possible compromise, that does include the mean as the most likely outcome, but since it is a three outcome distribution, it uses more resources. (In this case the mean would have probability ¾, and each of the others 1/8 each for a symmetric distribution. With choices of the mean and plus or minus 1.4 sigma, the probabilities are ½, ¼, and ¼ which is probably better.)

(The multitrajectory mechanism used in "eaglet" leaves these choices of what representation to use for individual event types in the control of the analyst. It is possible, for example, to designate acquisition events as being "multitrajectory" and acquisition loss events (which are usually less important) as "stochastic" and attrition events (which are usually still less important) as "deterministic". Full multitrajectory treatment means resources, and usually should be reserved for the most important events. It is possible to make simulation runs with limited numbers of trajectories in order to estimate the importance of various events, and use that information in subsequent simulation operations to manage the multitrajectory treatment of events. This has been demonstrated.

In summary, development of probabilities for unit acquisition will probably be found to be the best supported of any event type based on data already available. It should be possible to base these probabilities on field tests, application of probability theory, and the wisdom of subject matter experts. When the treatment of the event is more complex than the memoryless model of Boolean trials, especially if it involves a random draw for time to acquire, then some simplification to a discrete draw sufficiently equivalent in effect given the event importance is needed.

Acquisition loss is the process by which a unit drops a unit from its list of enemy units it is tracking. This represents that the enemy can no longer be detected sufficiently well to be tracked, or that it has ceased to be a concern. Analysis of event importance in "eaglet" found that this was one of the least important events. Many of the same considerations that apply to acquisition would also apply to acquisition loss.

4 ATTRITION

The process of attrition, generally used to refer to the combat processes proper by which military units cause losses to occur in enemy forces, are often a primary focus of combat simulation development. In aggregated simulations, for example one in which units are of battalion size or larger, the strength of the unit is usually tracked in terms of the numbers of various assets, such as tanks, APCs, trucks, and personnel. It is common to represent such numbers and attrition using continuous mathematics, so that a unit having 34.7 tanks suffers attrition of 2.3 tanks over the 5 minute combat calculation interval, leaving it with 32.4 tanks. Often such calculations of loss are deterministic. However, the basic processes represented, of individual weapon systems acquiring, firing at, perhaps hitting, and causing varying amounts of damage to enemy systems, is inherently both discrete and random. A stochastic representation of this attrition process thus may well represent the outcome of a blue on red attrition event as having some approximately Gaussian density function, and a random draw is made. But because of the very large numbers of such events in an aggregated simulation, attrition is often not a key driver in terms of random effects. In "eaglet" attrition event outcome variations were quickly found to be among the least important, and subsequent research was generally pursued leaving this type of event to be resolved deterministically.

If attrition of this aggregated sort was to be represented as a continuous variable stochastic process, then the same issues apply as outlined in discussing acquisition times. A stochastic random draw returns one choice, which can only be considered representative when many such choices are made. A Multitrajectory choice is inherently discrete. A two way multitrajectory choice should return the plus one sigma and minus one sigma values. This gives the correct mean and standard deviation for the outcomes although, of course, a much different density. One expects that over many attrition events the law of large numbers makes the overall result little different from what a Gaussian, or even deterministic choice of the mean, would have given. If a more representative distribution is desired, one can use a three way choice of the mean value with 50% probability, and the plus and minus 1.4 sigma values with 25% probability each.

It is also possible and advantageous to accumulate data concerning the effects of attrition and not resolve the event until later. For example, suppose a unit is subjected to several different attacks by enemy units during a particular time interval. One could simply accumulate the range of possible losses (minimum and maximum) for each, or modify a total range of losses as each such event occurs. This use of interval arithmetic could be used for other continuous events as well. As an alternative, each attrition event's mean number of losses could be accumulated, and an overall sigma for losses modified. Ultimately, the event would only need to be resolved when the information on the unit's status is needed, for example, when the unit itself fires at an enemy unit, or reports its status to a superior. This kind of "lazy evaluation" can be potentially expedient in reducing the number of trajectory bifurcations for other kinds of events as well. In "eaglet" total losses are accumulated for each 5 minute interval, and the actual "event" (in the multitrajectory sense) is the resolution into two trajectories representing either high or low representative cases, or three trajectories for the mean and two outliers, at the analyst may direct. There is no reason why resolution cannot be postponed indefinitely, until the actual unit strength is needed (for reporting or combat) or if the effectiveness of the attack must be reported for the attacking side.

As the level of aggregation decreases, it becomes more important to treat assets as being discrete, and attrition becomes either whether a discrete loss occurs (or possibly more than one) or even whether the unit itself is entirely destroyed. Even in an aggregated simulation, there may be some units, such as an AWACS aircraft, that should be represented that way. In such cases, it is necessary to fall back on the physics and targeting and damage mechanisms of the combat event to calculate kill probabilities. Those probabilities would then be associated with the events that resolve, given a decision to engage the target unit or asset, whether a loss occurs.

As an example, consider a tank platoon of three tanks being engaged by a single AT rocket launcher. For the event of the AT launcher determining whether to attack, the launcher operator must spot one of the tanks and retain it long enough to engage. If he has a 50% chance of doing so for any given tank, and trials are independent (they probably wouldn't be) then there is a .875 probability of launch. We will assume a 50% probability of hit given a launch, and 50% chance of "kill" given a hit, so the simulation would credit this engagement event with a .22 probability of attriting the tank platoon by one tank. (This is a very simplified example, but it is illustrative of the kind of phenomena that must be considered, and the aggregation of those phenomena using the rules of probability to account for larger numbers of assets in a unit.)

To consider losses to breakage and other non-combat effects in this category, the same approach would be used. For aggregated units asset losses would be a draw based not on enemy action but on known processes for representing the effects of wear and consumption. Where discrete losses need to be modeled, events having a probability of an asset loss for a unit would calculate the probability of such a loss based on the numbers and relevant aspects of unit operation, such as speed and terrain type.

5 DECISIONMAKING

Here the term "decisionmaking" is used to refer to the choice among a finite number of alternatives. It is not used for the process of fabricating courses of action, which also needs to be addressed, but must remain beyond the bounds of what can be included. Here "decisionmaking" will be discussed in the context of decisions made by the commander of a unit governing that unit's actions, although decisionmaking can also apply to many other processes, such as target selection for discrete engagements, which may occur below the command level. It is assumed that some form of decision rules govern decisionmaking, having an IF / THEN form. The condition, or guard, is typically a Boolean (logical) expression which can be evaluated to be either true or false. Figure 9 shows the decision structures used in "eaglet."

Information necessary to support evaluation of these expressions, as well as furnish information for other processes, is typically retained by the unit as a representation of the unit's "perceptions", things it knows. This would include a list of enemy units it can perceive, the orders under which it is operating, and information reported by its assets and / or subordinates. Taken together, such information constitutes the unit's, and its commander's, "understanding of the situation." (If the command of a unit is disaggregated into separate staff functions, each of those elements in effect becomes a decisionmaker with its own tasks, understanding of the situation, and decisions to make. That level of complexity is not addressed explicitly here, though the same principles in developing random effects and assigning probabilities should apply.)

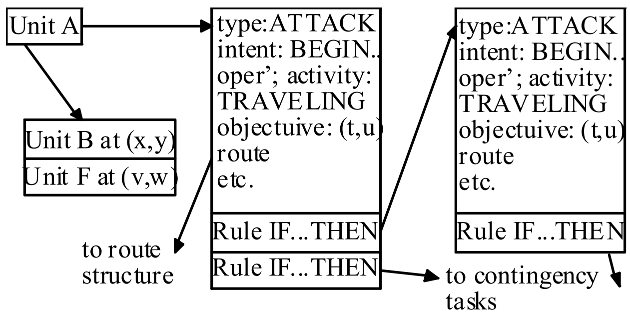

Figure 9: A model of information structures for orders

There are two ways envisioned by which stochastic elements could be injected into the process. One is that the information on which the decision is based is subjected to random effects, but the rules themselves remain deterministic. In "eaglet", this method was used. Two elements used in decision rules, a unit's "effectiveness" and whether it was "at its objective" were subjected to such variations. Each unit was always either "effective", "marginally effective", or "ineffective." The status was determined from the percent strength of the unit. But in fact, there were 3 different thresholds, the nominal threshold, an upper (pessimistic) threshold, and a lower (optimistic) threshold. Similarly, the condition for being at an objective tested whether a unit was centered within a nominal, lower (pessimistic), or larger (optimistic) radius of its objective. So, each decision rule was tested three times, with the pessimistic, nominal, and optimistic thresholds being applied. If none returned "true", the rule did not fire. If all returned true, the rule always fired. In cases where 1 or 2 firings occurred, arbitrarily chosen probabilities of .4 and .8 were used.

This method could be extended by applying similar threshold variations for other conditions that might be tested in a decision rule, such as fuel or ammunition state, degree of combat intensity, whether flanks are in danger, the recent own losses rate, or the perceived overall force ratio in the sector in which the unit is operating. It may be tricky to align the pessimistic / optimistic axis for all of the conditions appropriately for some rules. One could try every combination, with ultimately a many more than three evaluations needed, but it is not clear that this would be more representative of real decisionmaking.

The second general approach, which might be applied instead of or in addition to the random variations in criteria, is to make the rules themselves random. Typically decision rules are developed by subject matter experts working with modelers. The usefulness of the rules is limited by the mutual understanding that can be developed by these parties, and very complicated concepts for the rules may inhibit the development of good rules. It may be that development of good stochastic rules will be problematic. (It's hard already to develop a good set of deterministic rules!)

Another approach to stochastic rules is to supplement a deterministic rule with a rule which would result in the decisionmaker reaching a threshold for consideration of the action proposed, and a third rule for conditions under which the action is essentially compelled. Probabilities similar to those mentioned above could then be associated with each of the rule guard variations for a given action.

Ultimately, a good development of a stochastic model for decisionmaking needs to go well beyond issues of software technique and simplified models. It is hard to imagine the development of analytic grade probabilities without extensive investment in research with subject matter experts and varied conditions, then testing the rules and probabilities extensively.

6 CONCLUSIONS

There are many other types of events which have not been addressed in this paper, such as those associated with communications, assessment of intelligence information, and unit speed. However, it is expected that some of the concepts described for these most basic events will be applicable to some of those others. Ultimately, subject matter experts will have to address the issues of uncertainty in many contexts where deterministic approaches have been used, and developing "good" data is likely to be difficult. It is hoped that the approaches described here may serve in the interim for research purposes.

ACKNOWLEDGMENTS

Much of the work that led to formulating the issues 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

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.

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, as well as applications of computers. 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 language. His e-mail address is <sullivan@mathcs.wilkes.edu> and his Web address is <www.mathcs.wilkes.edu/~sullivan>.