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

CIRCUMSTANCE DESCRIPTORS: A METHOD FOR GENERATING PLAN MODIFICATIONS AND FRAGMENTARY ORDERS

John B. Gilmer, Jr.

Wilkes University
Post Office Box 111
Wilkes-Barre, PA 18766, U.S.A.

ABSTRACT

Circumstance Descriptors are offered as a way to organize spatial and other military knowledge that may be difficult to formulate, particularly the kinds of details that are most often illustrated by example. The goal is better modeling of military command elements in simulations. These Circumstance Descriptors are applied to assimilate features, both terrain objects and units, into a frame based Understanding of the Situation that organizes these into roles oriented around the decisionmaking unit's plan. A Circumstance represents a configuration of objects that may be present on the battlefield. If recognized, the effect is to splice new roles into the frame, extending it to cover the new features. A prototype has been built which demonstrates the use of these Circumstance Descriptors in both the context of planning and execution.

1 BACKGROUND

Representing the human decisionmaking process in military simulations is a critical challenge. This is particularly true when the need is for more than just choices among well defined alternatives, such as military plan generation and modification in a complex environment. Yet, decisionmaking, and the use and management of information in war fighting, is seen as a key capability in the structure of the U.S. armed forces. If simulation is to be used to represent armed conflict for purposes of acquisition, training, or concept development, then a reasonably good representation of decisionmaking in general and command / control in particular is needed.

This paper addresses this issue in the context of Army force on force modeling, particularly for analytic simulations where the use of human players to model decisionmakers in a simulation is not practical. Large scale simulations used for analyses may need to be run many times for variations of the scenario, and in order to assess the sensitivity of the results. Fully automated representation of the command control function is essential. If sufficiently good models can be developed, these would also be useful in other simulation applications. However, achieving good performance in the quality of planning and decisionmaking is difficult.

Shortcomings in modeling have often been overcome by techniques such as scripting that avoid the need to fully represent the full scope of what an actual military command might do. For example, pre-planned paths or sectors can eliminate the need for coordinated operation planning in the model. However, this comes at a significant cost: First, scenario development now requires not only the initial conditions, but also details that would normally be generated by the commanders of the forces involved. Second, the scripted actions provide only a limited scope for decisionmaking, and reflect to some extent assumptions of the scenario designer which may not provide for some of the things an actual commander might do. Finally, having scripts or procedures which relieve the model of a need for representing the more difficult aspects of command and control also means that the model cannot be used to examine many issues dealing with how command and control operates, particularly many pertaining to information warfare.

Here we will be concerned with simulations which attempt to actually model the command control process explicitly. In such a model, forces are relatively unconstrained; they can move anywhere physically possible within the simulated battlefield. A representation of command control must be present if the forces are to be coordinated and organized to conduct operations in a reasonably realistic manner. That representation must, in turn, be informed by a knowledge base about military operations, organized in such a manner to facilitate planning and decisionmaking within the model.

The Circumstance Descriptor concept was originally developed by Martin Marietta Advanced Technology Laboratories, Moorestown NJ (Now Lockheed Martin, and in Camden). A project from 1993 to 1994 aimed at implementing the Understanding of the Situation and a decision mechanism based on Circumstance Descriptors in

ModSAF, to assess and demonstrate its applicability to Distributed Simulation semi-automated forces (Gilmer & Krecker 1995). This paper reports on work done in 2000 having a similar goal, but in the simpler context of the "eaglet" simulation, on an unfunded basis during the author's sabbatical.

2 REPRESENTING MILITARY KNOWLEDGE

Military knowledge comes in many flavors, some of which are relatively easy to capture and put into a form usable by a model of the command control process. The best example of this is a decision rule that references readily measurable and well defined scalar quantities, and gives a binary result. For example:

> "An attack needs a local 3 to 1 superiority of force
> to have a reasonable chance of success"

Here we have reference to the ratio of two quantities for which some reasonable metric can be found, perhaps a weighted sum of numbers of weapons. The ratio can then be compared to a numeric value, 3, and the result can be used to inform a decision on whether an attack is to be made or not. This is an example of a reduction process: From relatively more information (strengths and various other factors that may be cited in a rule) a single datum is derived: a binary decision in this case.

There is much more to the command control process than choosing between two or a few alternatives. In particular, processes that are constructive need to start with a decision, for example to launch an attack, and must develop the details of a plan that actually meets the circumstances. This kind of process increases the scope of information, in that one starts with a choice or a small set of alternatives and elaborates considerable detail needed to carry out the operation. This is much harder to model well. This process, "planning," can be done in a number of ways, each with its own requirements for representing military operations.

One approach that has been used is the constraint based planner. One begins with objectives and assets, and an assumption that anything is possible. The possibilities are successively pruned by the application of rules which individually can be relatively simple, such as giving limits on the movement speed of forces. This paper does not directly address this style of planner, although the circumstance descriptors methodology may well be applicable.

An alternative is to rely on knowledge structures larger than individual rules that define stylized ways of conducting particular operations. Indeed, military doctrine is frequently presented in this way. An attack may be "frontal", "flanking" or an "envelopment", and these are illustrated with notional diagrams that convey the identities of the component forces, relationships, and the course of the operation over time. Figure 1 shows an example that is perhaps cleaner of details than most such illustrations (War Department 1990, p214). Figure 2 shows another such diagram with more details: very specific spatial relationships among various elements of the force (War Department 1990, p218).

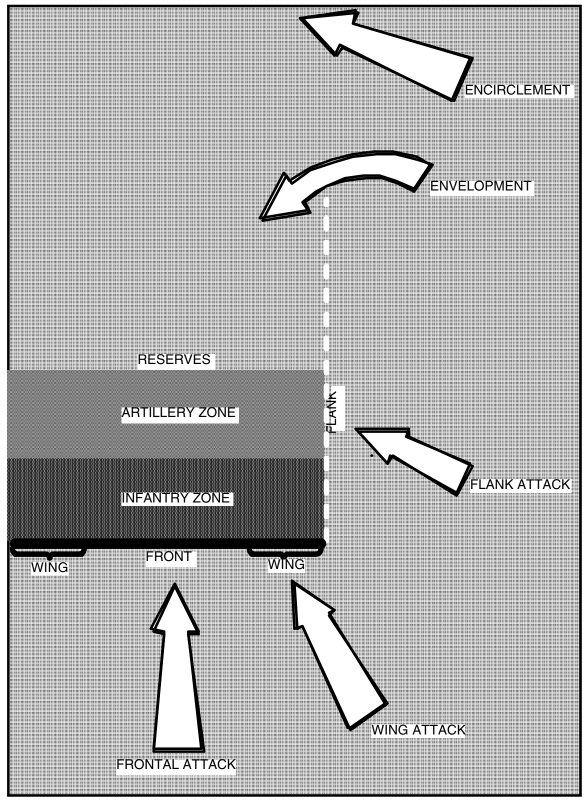

Figure 1: Notional Diagram Illustrating Forms of Attack

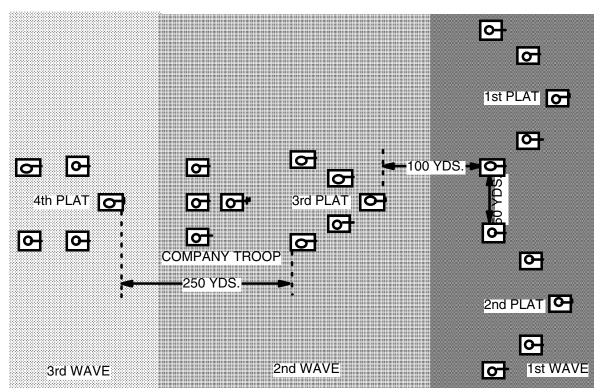

Figure 2: Diagram of a "Blunt Wedge" Tank Formation

This kind of knowledge representation has been used in a series of military simulations of land combat beginning with INWARS (Integrated Nuclear and Conventional

Theater Warfare Simulation) in the early 1980's and continuing in CORBAN (Corps Battle Analysis simulation) and others (Gilmer 1984, 1986b). Schemes such as Command Instruction Sets have many points of similarity. Dr. J. Aldrich, designer of INWARS, applied the notion of "Frames" pioneered by Marvin Minsky to concepts of military operations. There is much also in common with scripts. A frame includes a set of roles which are to be filled by the various actors in the military operation. The roles and frame also include spatial relationships, so that the structure can be seen also as a template. Associated with each role for a military unit are template orders which would be issued to a subordinate unit, or hypothetical actions of an enemy unit. Roles may be for terrain, such as barriers. The frames also include explicit phasing of operations as an execution matrix, with possibly multiple contingencies at various points that depend on rules that guide the course of the operation. Finally, there are rules for when the frame is applied, and for when it can no longer be considered applicable. Figure 3 illustrates a simple frame used in the "eaglet" simulation, and the corresponding spatial structure with contingencies for commitment of the reserve either left or right.

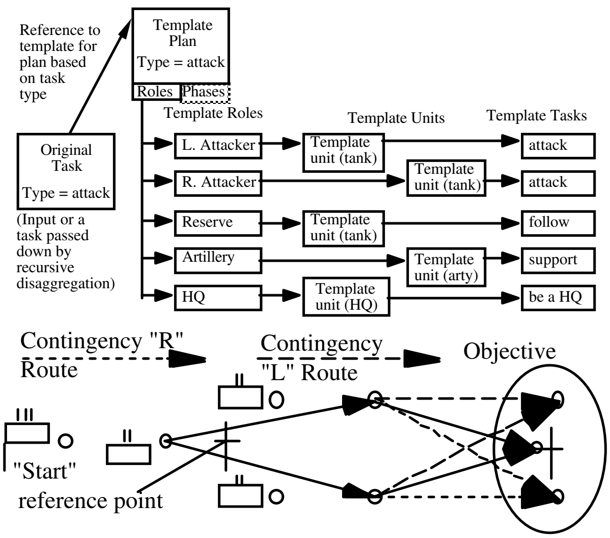

Figure 3: Brigade Attack Planning Template

The frames described above have been used with some success. But they have significant limitations, particularly at lower tactical levels where terrain effects become dominant. While the formation of Figure 2 may show spatial relationships, these do not survive undistorted when confronted with particular situations. This is seen in yet another figure from an actual description of military doctrine. Figure 4 shows what amounts to the front edge of the tank formation seen earlier attacking a defense position (War Department 1990, p221). Notice that the defense is not just stylized. The illustration shows particular hills, entrenchments, gun positions, buildings, woods, roads, and a stream. One cannot make a useful frame directly out of this! It would apply to only that one particular place in the world, which probably does not even really exist. Indeed, the purpose of the diagram is to illustrate by example features that may exist in an attack situation, but cannot be precisely defined in either configuration or quantity. Such diagrams seem to be more common than the more notional type. Figure 5 is another example. The principles that the diagram illustrates are not enumerated explicitly. The human reader has the intelligence to form from such examples more general principles that are drawn on in planning and other aspects of command and control.

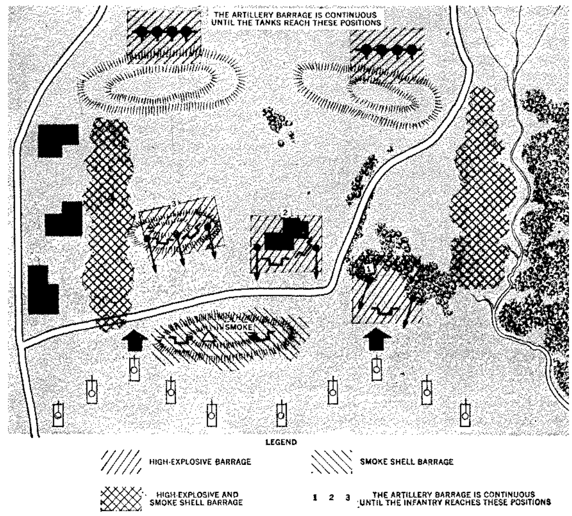

Figure 4: Artillery Support During a Tank Attack

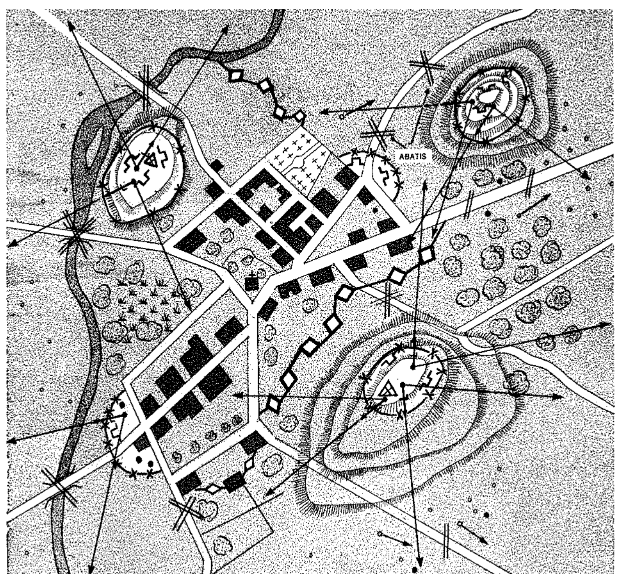

Figure 5: Infantry Positions Defending a Town

Capture of this kind of knowledge, that deals with how notional, stylized concepts are adapted to the many unpredictable particulars of military situations, remains a challenge. The author has developed the concept of "Circumstance Descriptors" as a means to address this issue, and has prototyped the mechanism in the context of a simple combat simulation, "eaglet".

3 UNDERSTANDING OF THE SITUATION

We are assuming that the simulation also includes models of how information on friendly and enemy units is gathered and comes to the attention of a particular unit commander. We can think of this as a map showing important terrain features on which positions and data about various units are marked. However, unless a human being is looking at such a display, understanding of the situation necessary to decisionmaking is absent unless there is some mechanism to ascribe meaning to the elements portrayed.

The frame approach is a starting point. The various sections of the five paragraph order can be mapped into particular parts of such a frame, forming an "Understanding of the Situation" that evolves with the operation. The provisions described here were included in CORBAN, although with more limited adaptability than one might like. The mapping from the five paragraph order to a frame is described in condensed form below (U.S. Army FM 71-1; Gilmer 1986).

Paragraph 1: Organization, Enemy and Friendly forces: Friendly and enemy forces that the commander is aware of are roles. The "orders" attached to these roles represent what these forces are expected to do. Particularly for enemy forces, these may be thought of as hypotheses which need to be tested as the operation evolves.

Paragraph 2: Mission: Defines the overall nature of the operation, time frame, objective. This can be thought of as an order from superior that was the starting point for the process of planning and detailing out the rest of the order.

Paragraph 3: Execution: Here the overall nature and phasing of execution defined, followed by particular assignments to each maneuver subordinate. This includes the execution matrix.

Paragraph 4: Support: The supporting roles and orders for artillery and other components, and priorities for logistic and air support would be included here.

Paragraph 5: Command: This would specify the movement of the command element itself as an entity on the battlefield, as well as addressing signal issues and such beyond the scope of this paper.

Rules (paragraph 3) are used for making transitions from one phase of an operation to another, or for recognizing a contingency having a response that is provided for in the concept of operation (e.g. committing the reserve to a threat to a particular part of a defensive position). In a human generated operation plan such things are often specified as being "on order." That is, the commander will decide when to initiate the action. For simulated decisionmakers, rules are required. Such rules may reference straightforward parameters such as time (e.g. "Begin phase 2 at 0500."), or some more complex condition ("Commit reserve role unit to x,y if left flank force ratio falls below 1:3"). Factors that can be considered include abstract measures derived from overall data about friendly and enemy forces such as force ratio to very specific questions. Such a criterion might be: "If hypothesis that enemy reserve has been committed to counterattack left is confirmed....."

To summarize, information collected about the state of the battlefield is "understood" by mapping it into the roles of the operation representation, and keeping the roles up to date. Decisions can then be based on rules which test the state of both overall aggregate measures such as force ratio and fuel state as well as specifics that can be associated with the various roles. Figure 6 illustrates this for the case of a "bounding overwatch" in which subordinates move forward alternately. One subordinate is in the role of "mover", the other one is in overwatch state. The two unit roles would exchange tasks at the next phase of the operation. There is also a role for an artillery unit that might be able to give support, but this role is unfilled. There are no roles in this particular frame representing the bounding overwatch for the enemy unit or a friendly neighboring unit that the commander is aware of.

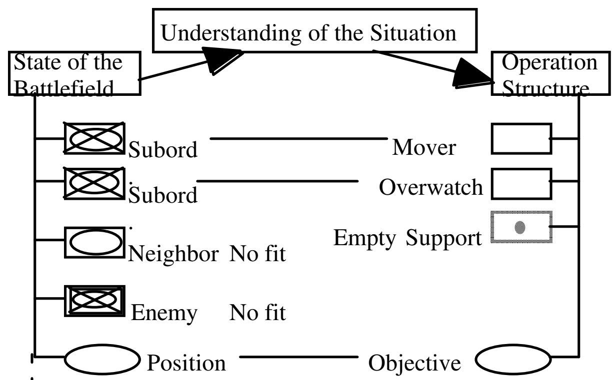

Figure 6: Understanding of the Situation

Now we can see where the limitation of frames as used in CORBAN becomes apparent. First, one can imagine including different phases/contingencies for an enemy reserve being committed left, right, or remaining uncommitted. However, there may well be numerous conditions that might need a response. These conditions may also occur in combination. The resulting complexity of the phase/contingency structure easily gets out of hand. A few phases at most is manageable, yet the author can recall some plans developed in the early days of CORBAN that had perhaps a dozen or more to account for various combinations of circumstances.

Another limitation can be thought of as a bound on the "imagination" of this frame based understanding. The frame as shown simply cannot accommodate information that does not fit into its "framework." For example, there may be a role for an enemy unit, perhaps labeled "enemy reserve" in the frame defining the concept of operation, and applicable to a particular unit as instantiated in the plan. The role would perhaps initially be empty, until an enemy unit is detected and found to satisfy the rules for filling the role. But, suppose a second such unit is detected? Or, suppose some enemy force is detected which does not fit the specifications of any role in the frame? One could say that this other enemy unit is "seen, but not understood." A unit that cannot be placed in a role can be considered only in abstract measures that aggregate information, such as overall force ratio. The frame does not allow for specific reasoning about what this new "unimaginable" enemy unit might be doing, or how it specifically might affect the operation. If this enemy unit cannot be "understood" in this sense, it is not possible for the simulated commander to take specific appropriate action.

The limitations are particularly apparent when it comes to terrain. The sector assigned to a unit may have many important terrain features in a large number of configurations, well beyond the scope of manageability of a frame as so far presented. A role for an aggregated feature such as a barrier or defense position can be found using a suitable algorithm perhaps. But this may leave various pieces of key terrain unaccounted for, with impacts not understood within the concept of operation. This issue was not a pressing problem in INWARS, with 10 km. terrain resolution, and terrain was not explicitly featured in the role structure. CORBAN, with 3.5 km. resolution, used algorithms to site unit positions in appropriate terrain within limited regions around the locations specified by a frame. Aggregated barriers, chokepoints, and other roles could be specified. While this may have worked tolerably well in that simulation, where they would be of manageable number, at higher resolutions the dominance of terrain features overwhelms the frame concept as described.

4 CIRCUMSTANCE DESCRIPTORS

Circumstance Descriptors are a construct that is intended to capture the kinds of particulars that appear in tactical diagrams such as those seen earlier, but are hard to express in text rules. As such, they represent fairly large "chunks" in the representation of military knowledge. They essentially amount to a rule system for the extension of frames that adapt the more general template to particulars that are too specific to be universal. It is not claimed that this purpose cannot be accomplished through more conventional rule sets. Rather, that the circumstance descriptor provides an organization for these rules that will allow appropriate rules to perhaps be more easily formulated, integrated into a knowledge base, and tested and executed in simulation runs. The key is that the circumstance descriptor seems to be a reasonable way of defining the principles that are expressed in the illustrative maps, and may prove to be more practical than approaches that use smaller, more fragmented, knowledge structures.

Figure 7 illustrates the general principle behind the use of Circumstance Descriptors. They can be thought of as freely floating chunks of information that might be applicable to any given situation. (As implemented in "eaglet" each type of operation has the circumstances to be checked listed). Fits can be made based on rules for the general situation (e.g. force ratio) as well as the existence or non-existence of certain roles, or the filled (or not) status of those roles. When a match is found, the circumstance is recognized and donates new roles to the operation structure. The use of circumstance descriptors can apply both during the planning process, or in a reactive sense as a plan is executed. In the latter case, the actions taken in many cases amount to "fragmentary orders" which order a subordinate or subordinates to take a particular action to respond to the circumstance. Note also that circumstances are not necessarily all unfavorable; some may represent opportunities, such as the chance for an encounter of a friendly armored force with enemy rear echelon elements such as air defense sites.

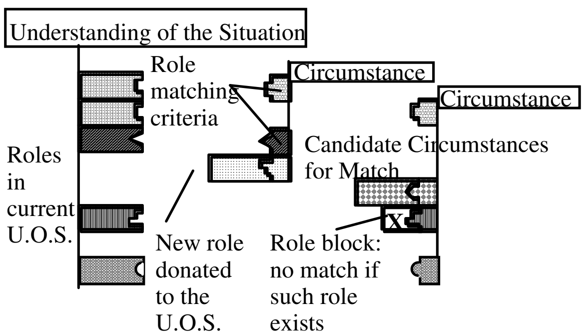

Figure 7: Circumstance Descriptors in General

In some sense Circumstance Descriptors are frames themselves: They have roles for the key actors that might be present in a unit's situation. In combination they constitute a circumstance calling for explicit recognition and action. Circumstance descriptors are particularly targeted toward integration of loose objects such as units or terrain features that may be important but do not fit any role in the existing frame that defines the units' understanding of the battlefield and its plan for carrying out its mission given that understanding. Figure 8 below illustrates abstractly how a circumstance descriptor "fits" existing roles and objects, in the process adding new roles.

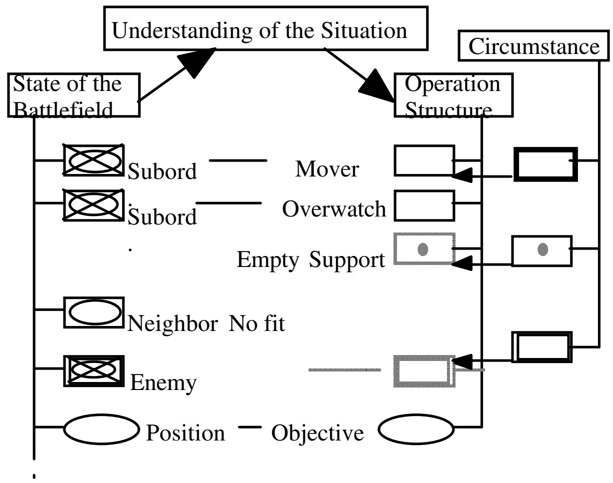

Figure 8: Circumstance Descriptor Use

In this case, the particular circumstance descriptor to be tried has three roles. Two are essential ones: "friendly subordinate" and "enemy unit." "Essential" means that the circumstance will not be recognized as applicable unless both can be filled. There is an interesting distinction between them. Note that in one case, the role is filled by a mapping from one of the roles already filled in the frame representing the operation. A "friendly subordinate" role will typically already exist. The "enemy unit" role is a role that only provisionally exists while a search is made to see if there is a known enemy unit that can fill it. It has no current corresponding role in the operation structure, but can be filled by the enemy unit that the decisionmaker knows about (it's been reported) but for which he has not yet found a corresponding role. If no enemy unit is found to fill it, neither the role nor circumstance are applicable. There would, of course, be rules giving suitability criteria for each of the roles. Not just any enemy unit would fit. It should be one that actually presents a threat that the circumstance descriptor is designed to recognize. In particular, the circumstance descriptor, like the template frame that defines the concept of the operation, must be oriented to the sector of the decisionmaking unit. We will assume that the enemy unit meets these criteria, in that it is in a location and of a type and capability which presents a threat to the moving unit.

A third role in the example circumstance descriptor, that for an artillery unit, is "not essential." Even if this role remains unfilled, the circumstance is recognized. When it is recognized, the roles that do not exist in the operation structure are added to it. In this case, the only added role is the enemy unit role, since the artillery role already is present, though unfilled. By recognizing the circumstance, rules associated with it defining possible actions to be taken are also attached to the operation structure. One might be, "If the artillery role is filled, send a message to the unit filling the artillery role to fire at the unit filling the enemy unit role." Another rule, applicable if the artillery role is unfilled, might ask superior for air support, or allocation of fire either (directly) targeted on the enemy unit or (indirectly) to support the moving friendly unit.

Figures 9 and 10 illustrate some similar circumstances which are either desirable or undesirable, depending on the identity of the victim. In the case of the undesirable circumstance, a similar circumstance that is more generally applicable, shown in Figure 11, might also apply, with the result that a smoke role (for the desired cloud of obscurants) is added to the structure, and hopefully filled rather promptly.

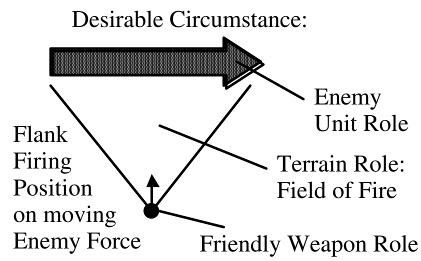

Figure 9: Desirable Flank Position Circumstance

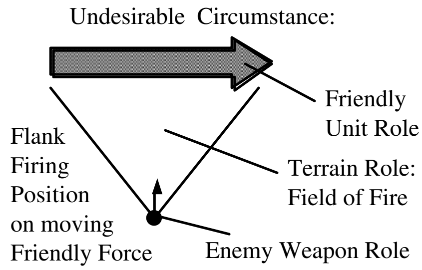

Figure 10: Undesirable Flank Position Circumstance

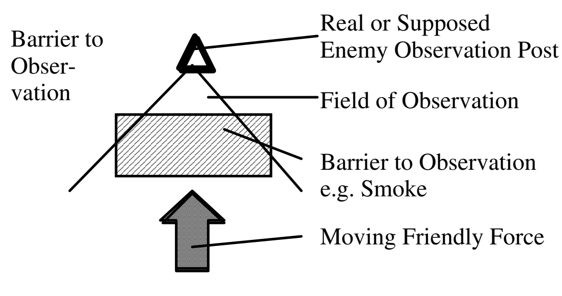

Figure 11: Enemy Observation Circumstance

More complex situations will arise, especially in planning, that can also benefit from the use of Circumstance Descriptors. Figure 12 shows a common problem: there is a chokepoint along a route that a subordinate (or perhaps the decisionmaking unit) must necessarily take. Recognition of this circumstance adds the chokepoint to the operation structure. This, then, may allow recognition of other, related, circumstances: Is there a likely enemy position with a field of fire covering the chokepoint? If so, and no enemy is detected there, then making sure that remains the case can become a specific tasking for intelligence assets. If an enemy is there, then air tasking or other means of suppression will be necessary. Engineering tasking may be needed for some chokepoint types. All of these would be circumstances and their remedies would be in the form of individual descriptors.

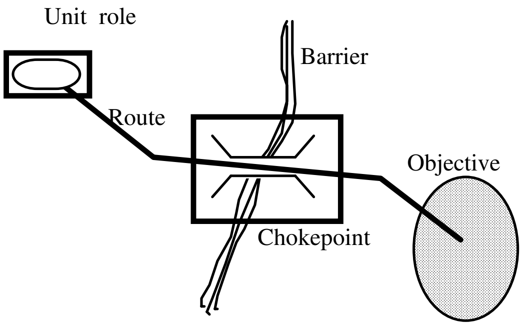

Figure 12: Chokepoint Circumstance

Circumstance Descriptors may serve in more complex situations as well. Figure 13 shows a case where a very unfavorable outcome can be projected if an enemy unit continues to where it encounters a point of vulnerability. If roles for a blocking position and a reserve can be filled successfully, and in this case temporal considerations are necessary, then the role results in a frag order to the reserve. A related situation which reflects almost the same condition as an opportunity is shown in Figure 14. Here the "vulnerable unit" serves as a decoy for an ambush.

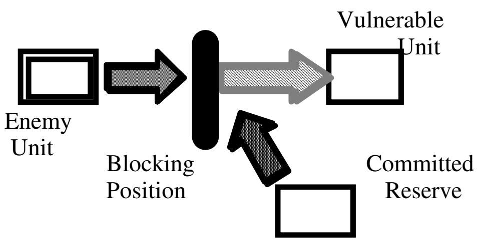

Figure 13: Interception Circumstance

The variety of Circumstances that may be envisioned are numerous. How much this construct can contribute to improved modeling of command control remains an open issue, as the work done to date has primarily addressed the question of how and whether it can be implemented in a force on force simulation.

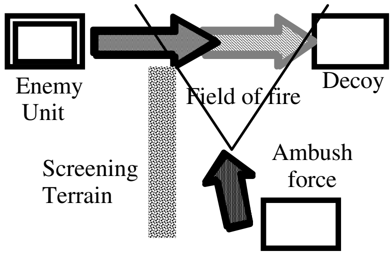

Figure 14: Ambush Circumstance

5 THE PROTOTYPE

Circumstance descriptors have been implemented within a version of the "eaglet" simulation. This simulation was designed to include many of the essential features of "Eagle", but is much simpler. It has been used in studies of Multitrajectory Simulation, and was convenient for the purpose of prototyping Circumstance Descriptors because it already had a simple frame based planner used to generate scenarios. In the variation of "eaglet" used to test Circumstance Descriptors, there is a terrain representation based on links and nodes, with a small set of features including roads, ridges, heights, urban areas, and woods. The roles used for the template based planner have been adapted for use in circumstance descriptors, and expanded in scope to include terrain features. (The terrain features have not, however, been backfitted yet into combat and acquisition routines at this time.) The "eaglet" simulation was written in C++.

One attractive reason for using "eaglet" is that the multitrajectory support also makes it fairly easy to use the simulation recursively, so that a decisionmaking element can create a simulation object with multiple trajectories in order to project the situation into the future based on different decision options it might take. This will allow Circumstance Descriptors to have a temporal component, in that rules will be able to query projected future states of the battlefield, not just the current one. Also, a simulation that can use itself recursively implicitly includes adversarial planning, in that the projections include the possible reactions of the enemy. A full discussion of recursive use of simulation is beyond the scope of this paper, and has not yet been applied together with the use of Circumstance Descriptors.

Figure 15 is a screen shot showing an execution of the simplest of the Circumstance Descriptors, that of Figure 9, in which the unit has a flank shot on an enemy. The enemy unit (#1) fills "role 1" (shown by the oval) for Unit 2 on the ridge. The second role is the field of fire, shown in light gray. Note that a ridge line (diagonal lower left to top) blocks the line of sight to the upper left corner. Had Unit #1 been a bit farther to the left (outside the field of fire region), its role might still have been filled, as would the field of fire

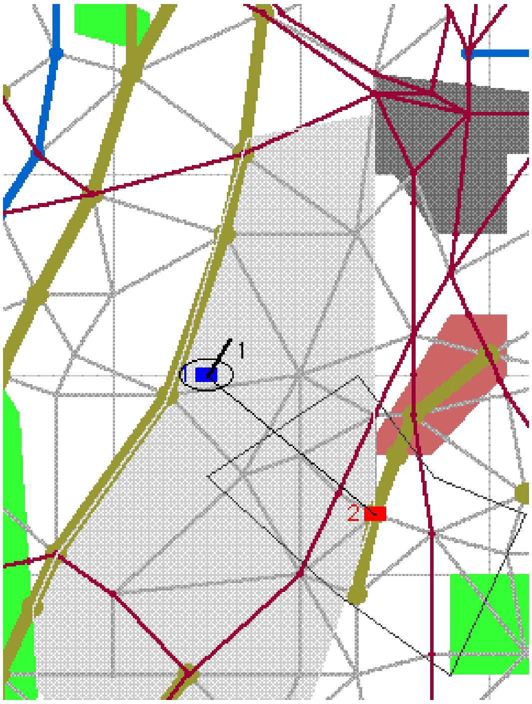

Figure 15: Flank Opportunity Circumstance Recognized

role. But the rule requiring the unit role to be within the field of fire role would have failed, and the circumstance consequently not recognized. Since all of the required conditions are present, the Circumstance is recognized, and appropriate action (for example, opening fire) can be taken.

The example seen in Figure 16 is more complex. It features an application of a Barrier Gap circumstance during planning. The Blue units occupy a defense position along a ridge line, with the fields of fire (for the two forward units) being roles that are shown in light gray. One role (#6, for an enemy unit forward of the position) is unoccupied. This "barrier gap" circumstance has identified an enemy unit (Role #10, in the lower left corner) which is on a trajectory which will intersect the barrier (representing the unit's main line of resistance) at a point where there is no field of fire. Recognition of this circumstance will lead to assignment of a new role, for an additional defensive position, and ultimately to the assignment of a unit to fill that role. (At this time "eaglet" is not developed enough to perform the assignment of the reserve, or a detachment of another unit, for that purpose.)

Figure 17 shows the use of a simplified version of the "Ambush" circumstance illustrated in Figure 14. This case differs from the others, in that the units are operating independently (without a common superior) and are doing so with simplified versions of plans called tasks having no permanent role objects. Nevertheless, the circumstances can be used. Unit 2 (near the bottom) recognizes a contingency in which unit 20 (top left, enemy) is on a path to approach Unit 2 (Top right). A field of fire is found that Unit 20 will have to cross, that gives unit 1 a flank shot.

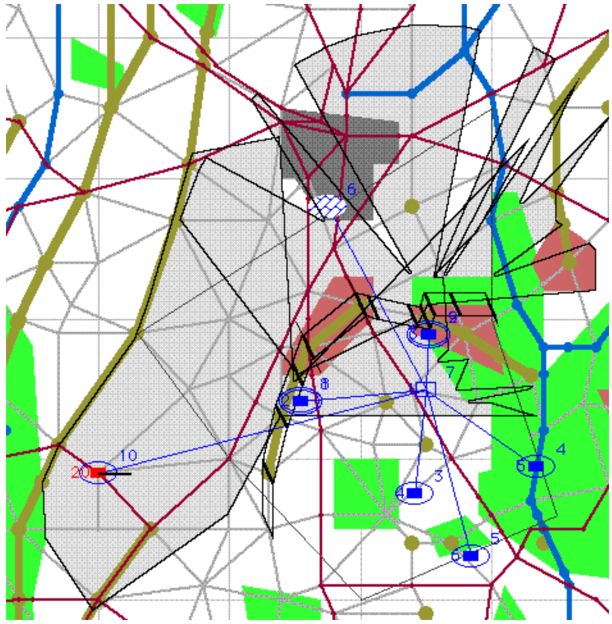

Figure 16: Barrier Gap Circumstance During Planning

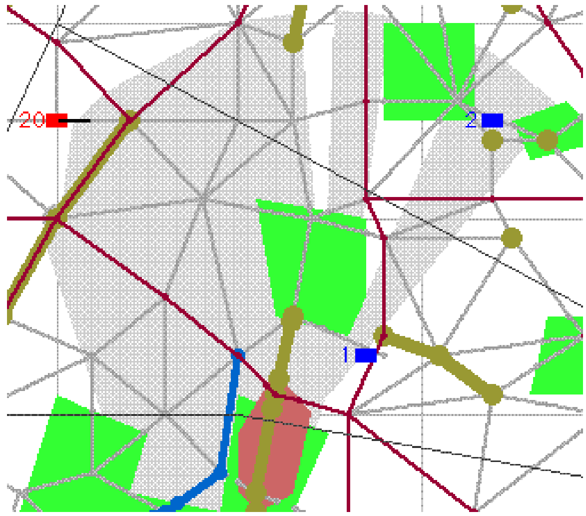

Figure 17: Ambush Circumstance for Unit with no Plan

(This simplified version of the ambush circumstance omitted the barrier role shown in Figure 14. Consequently, the firing position selected was on the height giving a wide field of fire. Improvements to the logic will be needed to specify a generally less preferable, but desirable in this case, narrow field of fire that exposes the ambushing force less.)

Since units with tasks rather than plans do not retain role objects, the roles are non-persistent; they are forgotten immediately after the decisionmaking phase of the unit's operation cycle. This means that less data and less elaborate information structures are needed. But it also is computationally expensive. (Because the objects are not persistent parts of a plan, some extraordinary code was needed to capture the field of fire in Figure 17, which is why the usual role symbols seen earlier are absent.)

The knowledge base is currently limited to a relatively few circumstance descriptors such as those shown, and the context in which they have been tested is limited to a version of the simulation which does not yet have terrain for combat and acquisition processes or a multitrajectory capability for plans and circumstance structures. This test and diagnostic version has been useful for development and testing, but is not yet able to do comprehensive runs.

The fact that prototyping has reached this stage is quite significant. The number of issues that lie between the overall abstraction described earlier in the paper, and the specifics needed to generate operational code, are considerable. The most complicated problems surround methods of reference: naming issues. A prototype scanner and parser that will process the rules and bring greater unity and consistency to this problem has not yet been integrated as of this writing.

As with most prototypes, this one has allowed numerous issues to surface which imply that much of the software organization might have been considerably better if it had been done otherwise. Such is the nature of prototyping. The software abstractions were deliberately kept minimal so that they would not get in the way of the modeling abstractions. Even so, some of the C++ class hierarchy actually used turned out to be somewhat counterproductive, and would be done differently given a fresh start.

The work done does not begin to fully exploit the possibilities that might be developed with Circumstance Descriptors. For example, the "Understanding of the Situation" as described is built around the structure of the unit's own operation plan. This can be thought of as a very "self-centered" viewpoint. If in addition the representations of other friendly units' operation plans are included, particularly the parent unit's and neighboring units', that opens the possibility of recognizing circumstances that are of import to those organizations. Should that happen, and the decisionmaking unit have adequate resources, it may be able to then recognize a circumstance to take action to help the other organization. This would be a step toward implementing an ability to act on the intent of the superior's order. In like manner, if a hypothesized enemy operation order can be constructed, points of attack on that operation might be recognized, rather than just attacks on enemy forces.

ACKNOWLEDGMENTS

The work described here is based on earlier research at Martin Marietta (now Lockheed-Martin) Advanced Technology Laboratory, particularly the contributions of Dr. Donald Krecker. This project also depends on research 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, particularly Mr. Gerry Cooper, Col. Andrew Loerch, and Dr. Robert Alexander.

REFERENCES

U.S. War Department. 1990. *Handbook on German military forces*. Louisiana State University Press, Baton Rouge, Louisiana. (Reprint of 1945 document.)

Headquarters, Department of the Army. 26 January 1998. FM 71-1 *Tank and mechanized infantry company team.* (Accessible via <www.army.mil>)

Gilmer, John B. Jr. 1984. The role of artificial intelligence in simulation of command control. *In Proceedings of the American Defense Preparedness Association Artificial Intelligence Symposium.* 181-224. American Defense Preparedness Association, McLean, Virginia.

Gilmer, John B. Jr. 1986. Translation of military operation orders: a natural language problem, In *Proceedings, Army Research Office Workshop on Future Direction in Artificial Intelligence*. 39-47.

Gilmer, John B. Jr. 1986b. Parallelism issues in the CORBAN C21 representation. *In Proceedings, Expert Systems in Government Conference*. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

Gilmer, John B. Jr., and D. K. Krecker. 1995. Modeling situational awareness for command decision making. Presentation at the workshop High Fidelity Modeling and Simulation in the DIS Environment.

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

BIOGRAPHY

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 has a Ph.D. in EE from VPI and currently teaches Electrical and Computer Engineering at Wilkes University.