A HIERARCHICAL RULE-BASED ARCHITECTURE FOR IMPLEMENTING INTELLIGENT ADVERSARIES IN A SIMNET ENVIRONMENT

Avelino J. Gonzalez
Computer Engineering Dept
University of Central Florida

Dan Mullally
UCF - IST
Orlando, FL.

Gilbert Gonzalez
UCF and IST
Orlando, FL

ABSTRACT

The availability of intelligent adversaries in a training simulator environment can clearly enhance the training experience for students. However, implementation of this capability into simulators has been slow as well as difficult. The semi-automated forces presently available for SIMNET, although quite sophisticated, still represents a partial solution, as the name itself indicates.

Representation of tactical expertise in rules gives rise to the problem of encapsulating every possible scenario within simple rules. This could lead to the need for a very large number of rules which, not only would have to be developed, but would also have to be efficiently executed in a real-time environment. This represents an unacceptable situation.

Improvements could be made by grouping rules according to the mission being simulated, but the number of rules required would still be large, and there would be no benefit of reusing situational knowledge commonly required in different missions.

The approach described in this paper is to develop a hierarchical ordering of rules which, at the highest levels, can be used to recognize the general situation being faced by the adversary. Examples of these situations are when an adversary needs to remain hidden from the student, or when it is appropriate to attack the student. Recognition of this high level situation will activate a lower level set of rules which will attempt to implement the prescribed course of action within the context of the situation. These will, in turn, activate another set of rules which will carry out the low level implementation details of the action within the simulation software.

1.0 INTRODUCTION

It is clear to anyone that in a networked simulated training environment such as SIMNET, the endowment of intelligence to a simulated representation of enemy forces represents an advantage in terms of tactical training. Making the correct tactical decision depends not only on evaluating the "static" situation such as the terrain, the weather, and the mission, but also on what the enemy does. Therefore, it is advantageous that the enemy entity behave in an manner representative of that of an actual enemy's.

AI techniques address the issue of representing and modelling human intellectual behavior in specific circumstances. The best developed sub-field in AI is that of knowledge-based systems. Using deductive reasoning as well as other means, but without actually modelling the brain, knowledge-based systems attempt to simulate the heuristic decision-making process followed by people knowledgeable in a specific domain when faced with a problem in that domain.

Since the enemy forces can be presumed to be knowledgeable in the tasks for which they have been trained (i.e., tactical warfare), it is only natural that knowledge-based systems be considered a prime candidate for use in this task. Nevertheless, there exist some significant obstacles to the use of knowledge-based systems for the task at hand:

The first one of these is that the knowledge possessed by a battle entity (i.e., an attack helicopter, a foot soldier, a tank platoon etc.), contains a large measure of common sense as well as survival instinct. Knowledge-based systems are not particularly suited to representing common sense or instinctive behavior. Although a certain amount of these can be adequately represented, it is virtually impossible to represent all of the common sense knowledge accumulated by individuals throughout a lifetime, or their in-born instinct to survive in a dangerous situation.

Secondly, knowledge-based systems are best employed when a limited set of inputs (i.e., scenarios) initiate the exercise of a base of knowledge in order to choose which of a limited set of alternatives to implement. A battle situation, however, can present to the student a virtually endless range of scenarios to which he has to properly react. To prescribe a course of action for each of the possible scenarios which could be presented is not a realistic means of implementing a knowledge-based system. That would require an immense base of knowledge which would be nearly impossible to capture and manipulate effectively as well as efficiently.

Thirdly, knowledge-based systems are typically inefficient and rather slow. This has been partly due to the traditional use of symbolic languages by researchers in the technical community. Such languages as LISP or PROLOG are generally interpreted, memory-intensive, and built for flexibility rather than speed. The more recent trend, however, is to develop knowledge-based system tools in the conventional languages such as C or ADA, thus somewhat alleviating the problem. Nevertheless, knowledge-based systems remain a comparatively slow means of computing when compared to conventional algorithmic methods.

The research described in this paper first analyzes the knowledge to be used by the intelligent opponent in the SIMNET environment and proposes an architecture for implementing it.

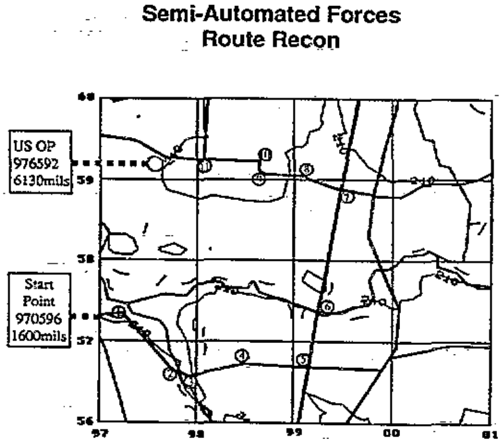
2.0 THE NATURE OF TACTICAL KNOWLEDGE

In order to devise an architecture to implement intelligent adversaries, it became important to understand his mode of thinking. It was decided to elicit such knowledge from an experienced tank commander, using the SIMNET environment as a demonstration testbed for that knowledge. The vehicle of interest was a single T-72 main battle tank (for the Warsaw Pact forces). The domain was limited to a reconnaissance mission along a road with the objective of searching for the enemy (which in this case was actually the blue forces). Upon reaching the final destination set out in the mission, the simulated vehicle was to take up a concealed point of observation and report all enemy activity until it was either attacked, discovered, or the main column of friendly forces reached its location.

2.1 Knowledge Elicitation Methodology

The methodology used was a combination of the traditional question-and-answer interview and observation. In the q&a phase of the process, the focus was on learning about military terminology and procedures, as well as some basic tactics. The capabilities of the weapon systems were also discussed as they might affect the tactics. This was carried out intermittently along with the observation phase.

The observation phase had two objectives: one was to observe the expert's tactical decision-making process, and two, to understand the capabilities of the SIMNET environment in which the simulated intelligent adversary would be located. This phase was composed of carrying out a simulated reconnaissance mission on a specifically-chosen piece of terrain in SIMNET in order to observe the expert's tactics. The terrain chosen was a relatively flat region around Ft. Knox, KY, which is represented in the SIMNET terrain data base. Figure 1 depicts the region used. An enemy tank (a "Blue force" tank in this case) was positioned near the destination of the mission in order to determine the range of visibility possible in SIMNET.



CP#1 973574
CP#2 97
CP#3
CP#4
CP#5
CP#6
CP#7
CP#8
CP#9
CP#10
CP#11

FIGURE 1

2.1.1 Observation About SIMNET

There were a number of significant observations made about SIMNET, which although not directly related to the objectives of this exercise, nevertheless provided a valuable insight into SIMNET and its effect on an intelligent opposing force simulation. Some of the more significant ones are as follows:

- Treelines, although very realistic from a distance, have no depth. Thus, they only provide concealment from the enemy when it is on the other side. However, it is difficult to position the tank to hide behind it and still be able to see from behind it. In order to do the latter, the tank has to protrude far enough through the treeline that it loses concealment.
- The tank crew is permanently limited to a closed-hatch condition, which does not allow dismounting in order to look over the top of a hillmass before exposing the entire tank. This limits the tactics that can be employed.
- Forest canopies are basically circular treelines. In other words, once inside the forest, the entire inside of it can be seen immediately. This is also not realistic, in that threats which could in reality could be hidden in the depth of the forest, can be seen immediately upon entering.
- It is somewhat difficult to ascertain, even from relatively close, whether a line is a road or a river. Additionally, bridges are defined only as the intersection of a road and a river. There is no superstructure or abutment to indicate a bridge from a distance.
- Lastly, there was considerable flicker in the horizon. This may be a symptom of the particular tank being used, and not a generic problem with SIMNET, but it could have been a reason why one of the authors incurred simulator sickness during the observation exercises.

2.1.2 Observations About Tactical Knowledge

Although the exercise on SIMNET provided a good initial base of knowledge, it's relative flatness and deforestation did not provide the appropriate environment for testing some of the tactics learned.

This led the investigators to create an imaginary terrain region on paper which included the type of features which would provide for more challenging tactical decisions. Figure 2 shows that terrain. The observation process now shifted from a SIMNET exercise to one of pencil-and-paper. This presented the advantage that terrain features could be altered in order to more deeply investigate the fine points of tactical maneuvering. Additionally, in order to understand the appearance of tanks in a more varied terrain as well as to understand the firing capabilities of the tanks in question, several exercises were held with the TOP-GUN gunner training simulator.

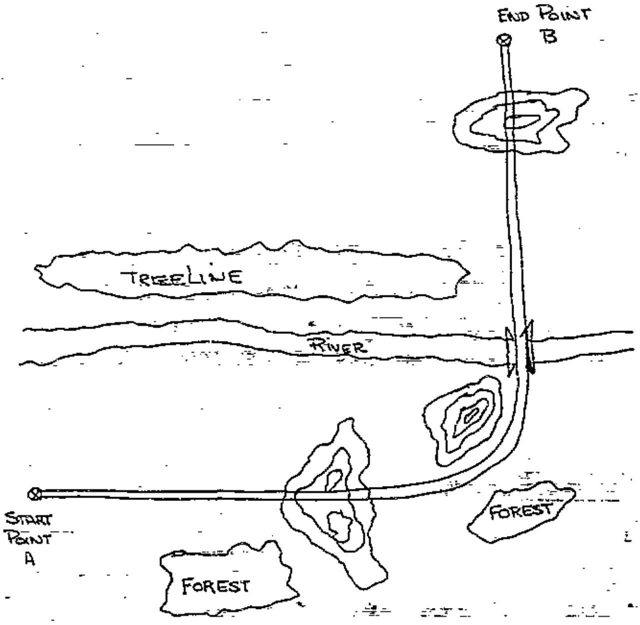


FIGURE 2

This process resulted in the following bits of knowledge:

- A reconnaissance mission requires tactical movement so as not to be discovered, and/or attacked by enemy forces which may be in the area. This is in contrast to administrative movement, which implies travelling somewhat more leisurely down a road, without the expectation of enemy contact.
- Tactical movement is composed of tactical path planning, a sub-task which requires the selection of the next immediate destination that will serve as a steppingstone to the final destination, and which will best provide protection from vulnerability to enemy sighting or fire.

2.2 Tactical Path Planning

Tactical path planning, as a sub-task to tactical movement, will be chosen based on:

- Mission: The type of mission will dictate whether tactical movement or administrative movement is required; whether the enemy should be engaged or whether retreat is preferred. In a reconnaissance mission, the objective is to see and not be seen. Therefore, engaging the enemy is prohibited unless acting in self-defense. *Reconnaissance-by-force,* on the other hand, attempts to find the enemy by presenting a target and drawing fire in order to engage the enemy and mislead him into thinking that it is part of the main force. Other missions which were not investigated are *attack, hasty attack,* and *retreat* as well as others.

- Terrain: The terrain features which are being included as part of the investigation are hillmasses, treelines, forests, rivers, roads and bridges. Each of these have different features as they affect tactical path planning. For example, hillmasses provide cover from direct ground fire. This is the most desirable terrain in which to be for protection against ground-based enemy. Treelines provide concealment from ground-based elements, but provides no cover since fire can typically penetrate the treeline. Forests can provide cover as well as concealment, depending on their size and orientation. Generally, they are not to be entered unless they are sparse. Rivers typically represent obstacles to movement. In this investigation, they are all assumed to be unfordable, and therefore, crossed only over bridges.

- Enemy Presence: The key issues are the direction of enemy threat, and the likelihood of having contact with the enemy. The levels of possible contact are *definite, likely,* and *unlikely*. The actual presence of the enemy is represented as a simple boolean yes or no.

2.3 Sequence of Events in Tactical Path Planning

In the process of trying to move tactically through an unfriendly territory in a reconnaissance mission, a tank crew commander will typically go through the following sequence of events in making a tactical path plan:

- Situational awareness, composed of:
  - terrain appreciation
  - weather
  - damage assessment (if any)
  - remaining stores (fuel, ammo,etc)
  - enemy assessment
- Course of action selection (tactical knowledge)
- Course of action implementation

Situation awareness knowledge is that which is used by the commander to review the situation and recognize the important points which he will use in determining the optimal course of action. Some of the things which this includes are recognition of potential danger (i.e., an ambush), being under fire, completion of mission, obstacles to completion of mission, avenues of approach, vulnerability to enemy sighting and fire, recognize concealment, cover, chokepoints, canalization, going too far out of the way and many other features about the terrain.

Tactical knowledge is then what the commander's experience indicates should be done in order to carry out the overall objective of the mission. This should include knowledge on tactical movement, conduct of fire, reaction to threat, mission execution, multi-vehicle tactics, escape or hasty retreat, and attacking, among others.

The course of action implementation should include physically moving the simulated entity from one place to another, avoiding minor obstacles such as a tree, realistic acceleration and deceleration, stop and turn the vehicle, fire, etc.

A further discussion of these events as they would affect the implementation of a semi-automated force will be included in the next section.

2.4 Rules of Tactical Path Planning

Rules of tactical movement are an example of the tactical knowledge discussed in the previous section. These were developed as applicable to a route reconnaissance mission, with an objective of getting from a starting point A to a final destination Z, establishing a point of contact at the latter, and reporting any enemy activity detected. The following are some rules which were extracted from the domain expert during the investigation. A later section describes the actual implementation of these rules.

1) Seek the closest cover from present position that leads closer to final destination.

2) Seek concealment when close-by cover in the general direction of the final destination is not available.

3) Move rapidly when cover and concealment are not available.

4) Select next partial destination before leaving a covered or concealed location.

5) Minimize time of vulnerability to enemy fire or sighting.

6) If time is of the essence, sacrifice cover and concealment for speed of movement.

7) Unmask potential locations of enemy before proceeding to them if possible.

8) Unmasked terrain becomes masked once again if unobserved for longer than thirty minutes.

9) Go to nearest cover when there exists concealment which is only slightly closer than said cover, and both are in the general direction of the destination.

10) Do not proceed to terrain elements which are further than 150 meters away from the roadway.

11) If enemy contact has been made and the enemy has seen you, seek cover as soon as possible, report their position and change mission to reconnaissance by force.

12) If enemy contact has been made and he has not seen you, then seek concealment, report sighting, maintain contact, and remain out of sight.

3.0 AN ARCHITECTURE FOR THE IMPLEMENTATION OF AN INTELLIGENT ADVERSARY IN A SIMNET

In a semi-automated force environment, a mission determination task could be called ultra-high level knowledge and it could be represented by a human, depending on the type of tactics to be taught, and the terrain to be employed. Although this could also conceivably be done with a knowledge-based system, it was clearly beyond the scope of this investigation.

Tactical knowledge, on the other hand, represents the essence of what a tank commander does in a tactical path planning task. This is a highly heuristic task, which is most clearly applicable to knowledge-based systems.

The situation awareness task is intermediate in nature because tactical path planning requires an assessment of the situation before a course of action can be chosen. It is probably best represented by a combination of knowledge-based and algorithmic techniques. The algorithmic portion would be composed of mathematical functions which can describe the lay of the land, the location of significant terrain features, and the distances between them and the tank. The heuristic parts would be the classification of the various features into a context usable by the higher level knowledge, for example, that a hillmass provides cover, but that it is too far away to be of immediate consideration.

The implementation of the course of action is the lowest level routine described here. It represents the actual carrying out of the selected course of action by the tactical knowledge module. This would best be carried out algorithmically since its function is rather procedural in nature. Figure 3 is a graphical description of the above levels.

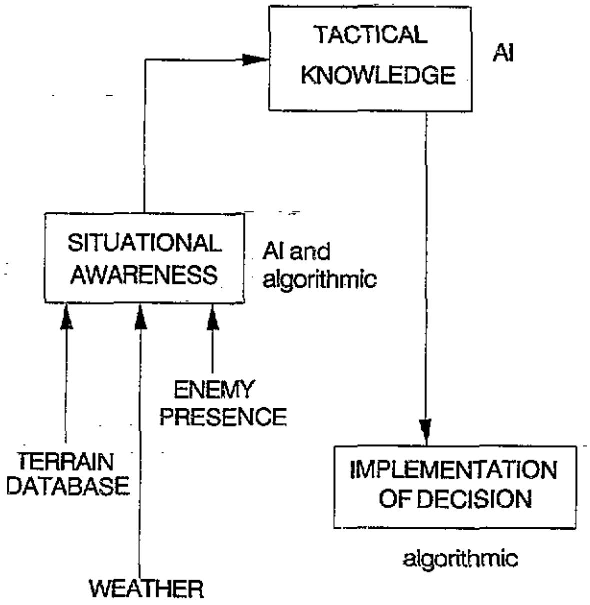


Figure 3 - High Level Knowledge Representation

3.1 Knowledge Representation Paradigm and Generalized Techniques

It quickly became clear to the investigators that the knowledge possessed by the expert was generally that of an If-Then rules. This was strongly hinted in the course of previous sections.

Although rules are a powerful means of representing and exercising knowledge, the need for frames became obvious just as quickly. Frames would allow for a richer representation environment where all the data calculated by mathematical functions, or new inferences made by the inference engine would be stored. The frames' capability for demons would be highly useful in calculating distances to various terrain features as well as determining the rate of usage of fuel, and the remaining store of ammunitions. Inheritance, on the other hand, could facilitate the representation of knowledge about a tank or other battlefield entity.

It is important, however, that an innovative technique known as *hierarchical rule classification* be implemented. Without this, representation of the problem through rules would be infeasible due to the numerous individual scenarios which would have to be identified and the rules to respond to each of these written explicitly.

3.2 Hierarchical Rule Classification

The concept of this technique is that behavior can be generalized such that rules can be written that would have applicability in a number of different scenarios. This could circumvent the need to write a rule for each possible different scenario, which would be clearly unworkable. The rules written in section 2.4 are general in nature, so that their implementation would be consistent with this approach.

The significance of this is that various sub-levels of knowledge would be required in order to classify the situation and provide inputs so that the generalized rules could use the information to make a determination. The knowledge used to carry out the latter would also be generalized except to a lesser degree, but it would also need lower levels of knowledge to support its task. This hierarchical relationship of more generalized, higher-level knowledge supported by more specific lower levels of knowledge is referred to as hierarchical knowledge representation, and represents an innovative technique to use in the solution of the intelligent adversary knowledge representation and execution.

For example, in the context of a route recon mission that is the subject of this investigation, a highest level rule would say that

> If the mission is route recon, and enemy contact
> is likely,
> Then tactical movement is recommended.

Another rule which would perform a similar function at the same level would be:

> If the mission is route recon and enemy contact
> is unlikely,
> Then administrative movement is recommended.

Assuming that tactical movement is recommended, the presence of that fact would activate another chunk of knowledge which would be used to assess the situation. It would have to interface with data present in some attached databases such as the terrain database, as well as use some of the givens of the problem such as weather conditions, the state of the stores, the damage incurred, etc. This chunk would in fact be the situation awareness module and serve to support the tactical knowledge chunk which would be the next one to be activated. For example,

> If the distance to treeline A from the present
> location is the shortest to the tank's present destination of all other treelines,
> Then treeline A is the closest treeline to tank.

Similarly,

> If the distance to hillmass B from the present
> location is the shortest to the tank's present destination of all the other hillmasses,
> Then hillmass B is the closet hillmass to tank.

Examples of other rules which support the tactical knowledge are:

> If hillmass B is closer to the final destination
> than the tanks present location
> Then hillmass B is in the correct general direction of the mission goal.

> If a terrain element is a hillmass,
> Then it provides cover.

> If the terrain element is a river,
> Then it provides an obstacle to movement.

> If the terrain element is a treeline,
> Then it provides concealment.

The tactical knowledge would then use this supporting information as follows:

> If the recommendation is tactical movement, and
> terrain element X is the closest cover, and X is in the right direction,
> Then proceed to terrain element X.

After the tactical knowledge chunk makes a decision as to where to move to next, then a lower level function would implement that decision. But assuming that there is contact with the enemy prior to the movement being made, and that fact is reflected by being under enemy fire, then another general rule would activate and it would look as follows:

> If under enemy fire,
> Then retreat to safety.

Another chunk that interprets a safe retreat would be activated next which would survey the situation again in a different light and recommend a new location to go. The block diagram in Figure 4 shows the hierarchical nature of the rules shown above.

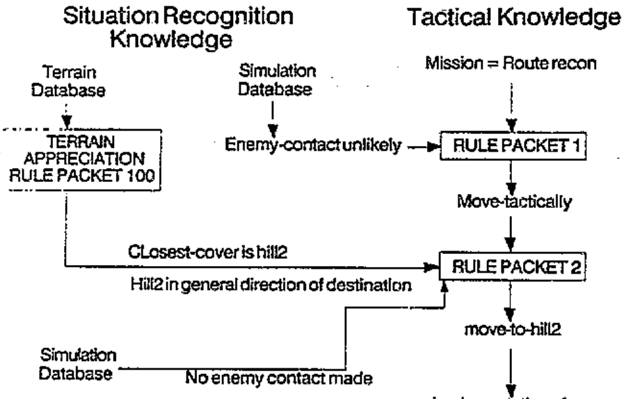


Figure 4 - Hierarchical Classification

4.0 PROTOTYPE DEVELOPMENT

The last stage of this investigation was to put into practice all the techniques discussed above as a prototype which would demonstrate their feasibility.

The objective of the prototype was to have one opposing force tank autonomously perform a route recon mission as described in the previous chapter.

The first decision faced was to decide what tool to employ in its design.

4.1 Knowledge-based System Tool Selection

Upon consideration of various commercially-available tools which could be used to develop knowledge-based systems, it was decided that CLIPS represented the best compromise between sophistication, performance and price. Developed by NASA Johnson Space Center, CLIPS is a PC-based, rule-based shell which employs the Rete algorithm in a pattern-matching environment. A distant relative of OPS-5 and ART, it is quite powerful, and relatively easy to use. Additionally, due to the fact that it is written in the C language, its interfacing with the C-based SIMNET system testbed being developed at IST would be infinitely simplified. The only weakness seen was that version 4.3 does not support frames at the present. However, it is understood by the authors that the upcoming version 4.4 will have that capability.

4.2 Initial Prototype Description

In order to familiarize the research team with the CLIPS tool as well as be able to better understand the knowledge being extracted, a simple initial prototype was designed which would conceivably act as an on-bard assistant to an inexperienced tank commander in a route recon mission. This prototype is a stand-alone, and it acquires information by asking the commander about the mission and the nearby terrain features. Although simple in nature and admittedly unrealistic in its assumptions of commander attention, it nevertheless performed its intended function well and provided a starting point for the further development of the final prototype.

4.3 Final Prototype Architecture Recommendation

The final prototype was designed to be implemented directly into the SAFOR testbed being developed at IST. It would require no interaction with the tank commander and would direct other routines to move the tank to a new location. It would be usable under any terrain or set of conditions that would be applicable to a route recon mission. It would interface indirectly with the SIMNET terrain database so as to be as realistic as possible for its intended SIMNET environment. Although not completed at the conclusion of the author's summer assignment, its design will be described below as a recommendation for further research in the topic of intelligent adversaries in a simulated training environment. Please note that due to the lack of time to complete its implementation, the design described is by necessity, a very high level one with little detail having been defined.

The basic architectural design of the advanced prototype would be in a object-oriented environment (C++), where the tank(s) in the simulation would be an instantiation of a tank object with such attributes as its present location, straight-line distance to the final destination, number of rounds remaining, speed capabilities, river fording capability, amount of fuel left, damage assessment, and various others. The tank object would be able to call methods which would assist it in its tactical path planning. These methods would be embedded CLIPS packages which could, in turn, access other auxiliary C functions in order to carry out its task. Figure 5 shows a graphical description of this concept.

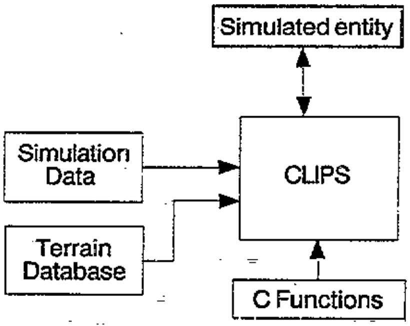


Figure 5 - General Architecture of Advanced Prototype

Many questions and details need to be filled in in this design. The largest remaining question at this point is the memory and processing time requirements of this type of arrangement on the limited resources of the simulation computer. It is hoped that the implementation of the high level prototype design described above would help answer these questions.

4.4 Future Research

Besides completing the advanced prototype described in section 3.3 above, there are other areas which merit investigation into its applicability to the problem at hand.

For example, an object-oriented extension to C such as C++ could conceivably simplify as well as enhance the some of the functionality of the testbed. Object-oriented extensions are ideally suited to simulations of independent objects which behave differently from other objects in the same environment. It also allows for easy modification and additions at a later date. And, although not yet proven, it is likely that the use of C++ will not introduce any additional overhead into the computations.

Other alternatives which could be employed if CLIPS proved ineffective in this application would be:

- a blackboard architecture system such as the Generic Blackboard. This has some definite drawbacks, but also great potential.
- a distributed processing environment.

Additionally, the knowledge for missions other than route reconnaissance would need to be extracted. Although not likely, this could reveal a significant complication in the endowing of intelligence to opposing forces.

5.0 SUMMARY AND CONCLUSIONS

In summary, it can be safely said that AI techniques such as knowledge-based systems are quite applicable to the problem of intelligent adversaries. In fact, it is believed by the authors that truly intelligent behavior in a simulation object cannot be practically achieved in any other way. The research performed under this grant was successful in achieving the following:

- analysis of the problem
- extraction of a limited portion of the tactical knowledge
- preparation of a scenario to use in testing the prototypCCe
- selection of a knowledge-based tool
- selection of a knowledge representation paradigm
- development of a simple prototype
- preliminary design of an advanced prototype system