TEC-0052

US Army Corps of Engineers
Topographic Engineering Center

Cooperative Autonomous Agents Testbed

S.Y. Harmon
D.M. Keirsey
D.W. Payton
S.C. Yang
M.D. Howard
J.A. Krozel
D. Y. Tsend

Hughes Research Laboratories
3011 Malibu Canyon Road
Malibu, CA 90265

November 1993

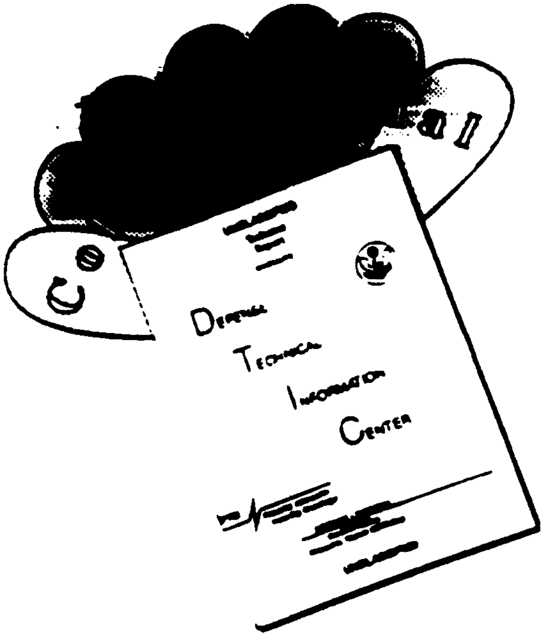
T E C

Approved for public release; distribution is unlimited.

Sponsored by:
Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington, VA 22201-1714

Monitored by:
U.S. Army Corps of Engineers
Topographic Engineering Center
7701 Telegraph Road
Alexandria, Virginia 22310-3864

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF COLOR PAGES WHICH DO NOT REPRODUCE LEGIBLY ON BLACK AND WHITE MICROFICHE.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
November 1993

3. REPORT TYPE AND DATES COVERED
Final Report Feb. 1989 - May 1993

4. TITLE AND SUBTITLE
Cooperative Autonomous Agents Testbed

5. FUNDING NUMBERS
DACA76-89-C-0002

6. AUTHOR(S)
S.Y. Harmon M.D. Howard D.M. Keirsey J.A. Krozel
D.W. Payton D.Y. Tseng S.C. Yang

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Hughes Research Laboratories
3011 Malibu Canyon Road
Malibu, CA 90265

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Advanced Research Projects Agency
3701 North Fairfax Drive, Arlington, VA 22201-1714

U.S. Army Topographic Engineering Center
7701 Telegraph Rd., Alexandria, VA 22315-3864

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
TEC-0052

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The goal of the Cooperative Autonomous Agents Testbed (CAAT) program is to develop new techniques by which multiple autonomous agents can interact intelligently and effectively in both cooperative or competitive modes of operation. Among the most significant accomplishments of this program are the development of two techniques that improve the effectiveness of decision making and that simplify the acquisition of tactics knowledge for computer generated forces used in large-scale distributed interactive simulations. We have developed a concurrent control paradigm that vastly improves decision making in the presence of multiple competing objectives. We have also developed case-based learning techniques to simplify the acquisition of tactics through direct interaction with subject experts. These techniques have been applied to automate simulations of M1 tank platoons as well as F-14 fighter aircraft.

14. SUBJECT TERMS
SAFOR, SIMNET, DIS, intelligent agents, cased-based reasoning, SOAR, behavior-based control, reactive planning, machine learning

15. NUMBER OF PAGES
80

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UNLIMITED

TABLE OF CONTENT

1. Executive Summary 1

2. Introduction 5

3. Technical Approach 11

Concurrent Control 11

Concurrent Control for SAFOR Implementation 12

Extensions To Other SAFOR Components 14

SAFOR Accomplishments 15

Automated Knowledge Acquisition 17

Behavior-Based Intelligent Agents 22

Selective Focus-Of-Attention 23

IFOR Accomplishments 26

4. Observations and Recommendations 29

Appendix

A. SAFOR Performance Analysis 31

B. Case-Based Tactics Acquisition 47

C. Experimental Analysis of Case-Based Tactics Acquisition 59

D. Description of Air Domain Behaviors 69

References

TABLE OF FIGURES

Figure **Page**

1. Plan view of route skirting a large building. 7
2. Platoon executing route march along large building 7
3. Command arbitration takes the range of acceptable commands from all active control laws and finds a single command that achieves the control objectives. 12
4. Composition of the Concurrent Control-Based Driver 13
5. Comparison between Concurrent Control and Finite State Machine methods. 14
6. Architecture for an integrated Concurrent Control-Based vehicle 15
7. The plan view geometry for an avoid/pursue scenario. 18
8. The Decision Space is a space spanned by the two state variables Angle Off (AO) and the Range (R). 18
9. The first case is stored based on features obtained from the Simulation Space. 19
10. Defining a second case in the Simulation Space will partition the Decision Space into two parts. 20
11. Using symmetry, a new configuration in Simulation Space can lead to the addition of two new cases. 20
12. Defining two more cases in the Simulation Space further refines the partitioning of the Decision Space. 21
13. As more cases are added, the shape of the regions in Decision Space will change accordingly, allowing for arbitrarily complex boundaries. 21
14. Case-Based matching controls which control laws are ON or OFF. 23
15. When a fighter sees the bogeys in the visual scene, markers track these agents. 25
16. The Simulation Space view shows how markers are used for the wingman, the support formation, a fired missile, the current target, and the current threat. 25
17. A scenario using a hypothesized marker "D" to represent the hypothesis that an ambushing vehicle waits behind a hill. 26
18. The above complex 2v4 tactical scenario involving cooperation between multiple vehicles has been successfully captured using our case-based acquisition tools 27

PREFACE

This research is sponsored by the Advanced Research Projects Agency (ARPA), 3701 North Fairfax Drive, Arlington, Virginia 22201-1714 and monitored by the U.S. Army Topographic Engineering Center (TEC), Alexandria, Virginia 22315-3864, under Contract DACA76-89-C-0002, by Hughes Research Laboratories, Malibu, CA 90265. The Contracting Officer's Representative at TEC is Mr. Kevin Mullane. The ARPA point of contact is CDR Dennis McBride, USN.

1. EXECUTIVE SUMMARY

The goal of the Cooperative Autonomous Agents Testbed (CAAT) program (contract #DACA-89-C-0002) is to develop new techniques by which multiple autonomous agents can interact intelligently and effectively in both cooperative or competitive modes of operation. The initial phase of the contract focused on simulation of the behavioral aspects of ground-based autonomous agents such as M1 tanks, while the second phase directed its effort on simulating autonomous F-14 aircraft behaviors in beyond-visual-range (BVR) engagements. The resulting technology from this program favorably impacted both the SIMNET SAFOR (Semi-Automated FORces) capabilities and IFOR (Intelligent Forces) performance in the DARPA WISSARD (What If Simulation System for Advanced Research and Development) program.

For SAFOR, a new control paradigm, concurrent control, was developed and implemented which virtually eliminated all collisions by the M1 entities, and produced realistic behaviors in complex mission scenarios. A thorough performance analysis was undertaken to quantify the realism of SAFOR exercises, and a series of performance metrics was derived to permit the quantitative evaluation of the modifications. For IFOR, the utilization of machine learning techniques (viz. case-based reasoning) allowed the rapid acquisition of tactics, thereby injecting a large variety of maneuver tactics into the simulated F-14 entities. As a result, diverse and credible air tactics were generated which could be selected during run time in BVR engagements.

Funding for the CAAT program was provided by both the Army Advanced Concept and Technology (ACT) Committee and DARPA, with the Army Topographic Engineering Center (TEC) as the contract monitoring agency.

Program Highlights: The CAAT program produced several significant technology advancements in providing autonomous agents with a high degree of perceived realism in executing military missions. Highlights of the CAAT program are summarized.

For SAFOR, the SIMNET SAF 6.6.1 version was used as the basis, and we significantly modified the control scheme by replacing the existing finite state machine approach with our concurrent control technique, resulting in the following accomplishments:

- The development of a concurrent control technique which permitted SAFOR entities to pursue simultaneous goals, and the use of an arbitration scheme to execute the best resultant action.

- The elimination of nearly all collisions with moving and stationary objects by the application of concurrent control to the SAFOR driver.

- The derivation and evaluation of performance metrics to quantify improvements in executing specific maneuvers. Twenty one exercises were executed and evaluated, involving large and small buildings, stationary and moving vehicles, individual and platoon size units, and parallel and intersecting routes.

- The extension of concurrent control techniques to the SAFOR gunner and commander in order to carry out complex exercises. The exercises which were performed included coordinated platoons attacking a prepared position, a company on a road march defending against an ambush, and a movement to contact maneuver.

For IFOR, our concentration was primarily directed at developing a prototype tool suitable for rapidly capturing air tactics knowledge, and demonstrating its usefulness in BVR engagements for up to 2v4 scenarios. Two primary objectives were to be achieved: In the short term, to demonstrate an interactive knowledge acquisition method for capturing a large variety of air tactics knowledge which can be dynamically retrieved and used in BVR air engagements; For the longer term, to provide a method of guiding SOAR agent development (for the WISSARD program) by exploring a broad variety of tactics which can help SOAR developers in focusing on the overall behavioral requirements of the SOAR agents. Several significant accomplishments were achieved in the IFOR development:

- The ability to rapidly acquire new air tactics knowledge by interacting with the domain expert and ModSAF (Modular SAF) simulator through the use of the IFOR knowledge acquisition tool; the new knowledge can be acquired within a few hours.

- The representation of air tactics as cases, thereby permitting the utilization of case-based reasoning and matching techniques to determine and select tactical maneuvers appropriate for IFOR engagement exercises; the tactics are selected and matched dynamically as the engagement evolves.

- The simulation of BVR engagements for up to 2v4 configurations utilizing the case-based matching method of air tactics selection and case-based learning techniques for acquiring new tactics; as a result of these simulated exercises, new air tactics knowledge was acquired, and was incorporated into the knowledge base as additional cases.

- The exploration of more than 80 cases of air tactics and their use in simulated BVR engagements; the simulation of 1v1, 2v2, and 2v4 engagement exercises which demonstrated realistic behavior for the IFOR agents.

- The initial utilization of the IFOR agent as a challenging opponent for the SOAR agent in 1v1 BVR engagements. The purpose was to demonstrate the effectiveness of this method for pinpointing deficiencies in the SOAR agent knowledge base, thereby making it possible to enhance and effect marked improvements in the SOAR knowledge base.

<u>**Recommendations for Further Research**</u>: Although computer generated forces (CGF) such as SAFOR and IFOR have been reasonably successful in simulating acceptable human behavior at the platform and weapons control level, further advances in CGF capabilities will not be possible until progress is made to automate the command and control functions. This is one of the primary hurdles that must be cleared in order to achieve force aggregation to the higher echelons of command. The automation of command and control, unfortunately, is one of the most difficult challenges facing CGF technology today. Successful automation of command functions involves many of the hardest problems in artificial intelligence, including situation assessment, planning, knowledge representation, and intelligent control.

Based on our experience in developing SAFOR and IFOR technology for the CAAT contract, we recommend the following issues to be addressed in future CGF developments:

- The design and implementation of a canonical CGF commander model which is valid at various levels of command structure. This model must allow the commander's processing tasks to be partitioned into symbolic reasoning and reactive response components. It also must reproduce the communications between superiors and subordinates by means of combat orders.

- The ability of the CGF commander to accept mission requirements issued by either a human operator or another CGF commander. This requires that a representation be selected that is common to both modes of operation, allowing CGF actions to be interpreted by human operators.

- A design which provides the flexibility to enable a human operator to access any level of command within the CGF structure in order to supplement CGF capabilities by directly controlling the CGF, when necessary.

- Specialized reasoning modules for such activities as interpreting mission objectives, performing situation assessment, interpreting tactics and doctrine, planning mission activities and executing commands. Wherever possible, the design should take advantage of the many well developed inferencing, interpretation, and reasoning techniques available from related disciplines.

- Representations of tactics and doctrine in easily accessible and modifiable data bases; such knowledge should not be hardcoded into the CGF programs.

Using this approach to develop the command and control framework of CGF, we believe that faithful representations of authentic command and control of CGFs can be achieved. Additionally, it will preserve design options so that future force structures and organization changes can be modeled without prohibitively expensive alterations.

2. INTRODUCTION

Background -- The original objective of the Cooperative Autonomous Agents Testbed (CAAT) program was to investigate the interactive behavior of multiple autonomous agents in both cooperative and competitive situations. In a previous ARPA Autonomous Land Vehicle program, we conceived and developed the behavior based concurrent control scheme for navigating an autonomous vehicle in cross country terrain [Olin and Tseng 91]. Because this technique proved to be critical in controlling a single autonomous agent, we wanted to extend it to the multi-agent domain in order to study the broader problems presented therein. These issues include the degree of interaction needed to carry out complex tasks, the level of knowledge that must be shared among agents, the representations of this knowledge, the tradeoffs between centralized and decentralized control, and the best methods for maintaining coordination.

During the first year of the CAAT program, the SAFOR needs of the Army SIMNET program became known to us. This presented an excellent opportunity to direct the general multi-agent research efforts to a specific and timely application. In doing so, all the ingredients of the original CAAT program goal were preserved while facing a provoking challenge to apply these new techniques to a real application. In consultation with the CAAT program manager, we decided to focus our program efforts on the SAFOR problem. This proved to be a good decision because the concurrent control technique eliminated nearly all of the collisions and erratic behavior of the existing SAFOR, and allowed complex scenarios to be performed. Because of this, we were able to undertake a thorough performance analysis to quantify the realism of SAFOR exercises, and use a number of metrics to quantitatively evaluate SAFOR performance. These metrics were used to determine the capabilities of the concurrent control technique, and also were used to quantitatively compare SAFOR capabilities which used different control schemes. Highlights of concurrent control SAFOR results and the improvements achieved in SAFOR performance are discussed below.

The behavior-based concurrent control paradigm, which proved to be a critical technique in demonstrating well behaved performance by SAFOR entities, was also extended to the air domain to control the behavior of F-14 IFOR simulations. To provide insight and to contribute to the WISSARD program, a second objective of this program was to develop a prototype tactics acquisition tool to gather the tactics knowledge required for controlling the IFORs in beyond visual range (BVR) engagements. To complement the current direction of the SOAR effort, viz. the pursuit of deep knowledge for a narrow range of scenarios centering on a single agent, we focused on the rapid capture of a broad range of tactics knowledge which could be applied to a variety of scenarios, such as 1v2, 2v2, 2v4, including the availability of missile resources. In this way, the future direction of SOAR development, as it tackles the more complex scenarios, can be guided by the wide range of results obtained in this program. In other words, the "breadth" vs. "depth" approach for tactics knowledge acquisition should help to effectively channel future SOAR development by pointing out the pertinent requirements, as well as pinpointing the deficiencies in the SOAR knowledge base. Highlights of IFOR results and the use of case based methods for tactics acquisition and engagement exercises are discussed below.

SAFOR -- The development of our concurrent control SAFOR was carried out using the SIMNET SAF 6.6.1 version of the code (the only version available at that time). We utilized SIMNET SAF 6.6.1 as the basis of our SAFOR development because of the large amount of work that already existed from the Army SIMNET program, and due to the limited scope and resources of the CAAT program. In principle, this effort involved the removal of the control portions of the existing SIMNET SAF code, and substitution of our concurrent control technique in its place. In reality, however, this proved to be a formidable task. There were missing modules, undefined subroutines, and numerous uncommented portions in the code. Through diligence and perseverance, we were able to understand the code to make the transition to concurrent control possible. The results in the subsequent SAFOR improvements were achieved using our concurrent control technique in the SIMNET SAF 6.6.1 code.

A major cause of the collisions and erratic behavior of the ground vehicles in SIMNET SAF was due to the use of a finite state machine model of control. In this scheme, the world state is evaluated at each time interval and an action is chosen to address the prevailing situation at that time. Because each chosen action focuses on a specific world state problem relevant at that immediate moment, there is no assurance the solution selected will not initiate, propagate, and amplify further instabilities -- even though it may be the correct solution for the immediate problem at hand. Multiple goals cannot be handled simultaneously by the platform entities, as is often required in complicated situations. In reacting to complex commands, the use of a finite state machine model produced such erratic behaviors as cyclic maneuvers, random motions, multiple collisions, inappropriate focus of attention, and abandoning members of a unit.

The basis for our concurrent control methodology is that a SAFOR entity should be permitted to pursue multiple simultaneous goals in the execution of a mission. Each goal represents a single task, implemented as a behavior, that attempts to optimize its own performance independent of the other goals. An arbitration scheme was developed to resolve competing or conflicting behaviors and transform the resultant combination into a single command that is transmitted to the SAFOR entity for execution. A detailed discussion of our concurrent control technique is presented in the Technical Approach section. Applying this control technique to the SAFOR driver, we were able to eliminate nearly all collisions by the SAFOR entity with moving and stationary objects.

The application of concurrent control also enabled the execution of complex scenarios with a high degree of realism. One such scenario required a SAFOR platoon in line formation to follow a route which skirted around a large building, Figure 1. The route was chosen so that two of the SAFOR vehicles would collide with the building if it maintained formation rigorously. Our control scheme directed these vehicles to break formation while maneuvering around the building, Figure 2, and then reform after the building had been cleared. This is an example of the arbiter shifting priority to adjust to dynamic requirements encountered during execution. A detailed discussion of concurrent control is given in Section 3, Technical Approach.

To test the validity and capability of concurrent control, an extensive analysis was undertaken to quantify the realism of SAFOR performance, and a series of metrics was derived to permit the quantitative evaluation of the modifications. Twenty one exercises were executed and evaluated, with concurrent control applied only to the SAFOR driver. These exercises were carefully designed to enable the analysis of the results and the isolation of the effects of different conditions presented to the SAFOR units. The taxonomy of exercises was partitioned into large and small buildings, moving and stationary vehicles, individual to company size units, and parallel and intersecting routes. When platoons were involved, both the column and line formations were tested. The metrics used to gain quantitative insight on SAFOR performance consisted of total vehicle collisions, total building collisions, mean route efficiency, mean vehicle speed, and the dispersions of latter two parameters. A detailed discussion of the exercises performed, the analysis, and the evaluation results is contained in Appendix A, where

Figure A1 shows the exercise taxonomy, Table A1 lists the specific characteristics of the calibrated exercises, and Table A3 summarizes the evaluation results.

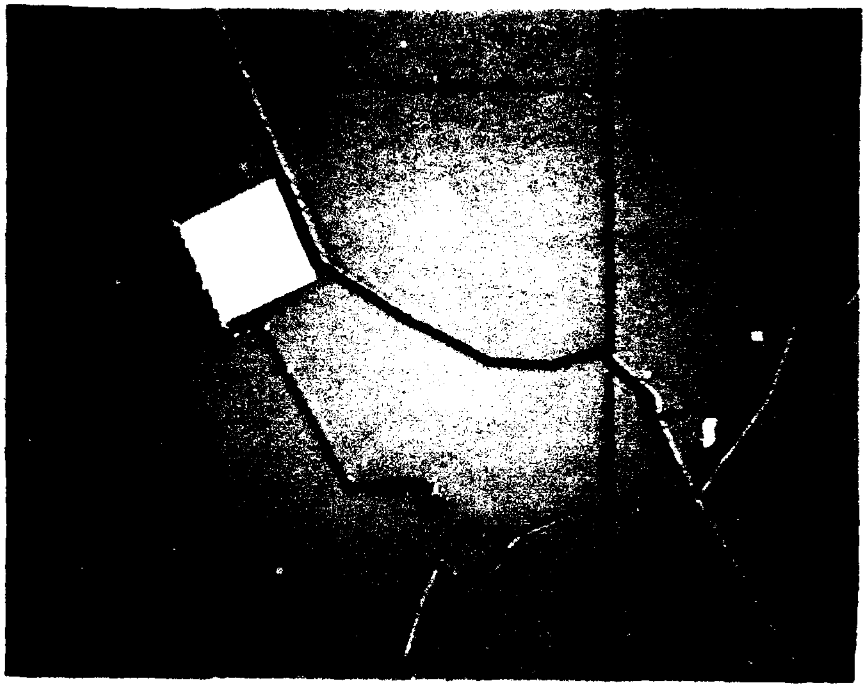


Figure 1. Plan view of route skirting around a large building.

Figure 2. Platoon executing route march around large building.

To further improve SAFOR capabilities, concurrent control techniques were extended to the SAFOR gunner and commander, with rudimentary communication between the SAFOR entities implemented. This enhancement permitted the exploration of significantly more complex maneuvers. The complex maneuvers carried out consisted of: bounding overwatch, coordinated platoons attacking a prepared position, movement to contact, and a company on a road march encountering an ambush. The last maneuver was the most complicated because it required the company to disperse into coordinated platoons upon initial ambush, each platoon engaging the ambush units, and, upon termination of the engagement, re-assembling the surviving units to continue the road march.

IFOR -- To accomplish the goal of rapidly acquiring tactics knowledge and to conduct BVR engagements for a wide range of scenarios, we chose to utilize case based reasoning techniques. Case based reasoning provides an excellent compromise between purely automated knowledge acquisition and knowledge engineering. Of course, the ideal method for capturing tactics would be to simply observe an expert engaged in realistic scenarios. Without sufficient background knowledge, however, there is no way to know how an expert might have altered his behavior had the details of the scenarios been even slightly different. This leads to the more conventional knowledge engineering alternative. Knowledge engineering involves a computer programmer attempting to understand the domain by talking with and observing an expert and then encoding that knowledge into a computer program. While effective, capturing the knowledge is a difficult and time consuming task. As a compromise between these two extremes, we have developed a methodology that permits an expert to input his knowledge directly through sets of example cases. The expert is able to interactively explore how scenario variations can influence the behavior of the intelligent forces, and is thereby able to iteratively taylor the intelligent forces to behave in a manner that reflects his own choices.

Our case-based tactics acquisition approach exploits the natural tendency for people to express their knowledge in terms of specific problems and examples. Looking at, and reacting to, a concrete situation, an expert can provide rules of thumb or suggest specific actions to be performed. Our system generates concrete situations for the expert through the use of intelligent force simulations. At any time during a simulation, the expert may suggest different actions and the intelligent forces will respond accordingly. The expert can then observe the effects of his suggestions and revise them as he sees fit. As this process goes on, the expert is actually generating a database of cases that can be used by the intelligent forces in future simulation exercies to emulate the expert's tactical responses.

The knowledge acquisition tool has been used to capture knowledge for a number of multi-agent scenarios. During tool development, the emphasis was placed on the less demanding 1v1 scenarios. Subsequently, to test the flexibility of the approach, more complex scenarios involving up to 2v4 engagements were acquired. The latter scenarios required cooperative IFOR agents to fly in formation, performing coordinated maneuvers, while in pursuit of mission objectives. In designing the agent architecture, the need to consider multiple vehicle scenarios was seen as essential to the realization of long term IFOR objectives.

An important long range goal of the IFOR/WISSARD program is that of making the SOAR agents capable of learning from pilots in action. To this end, it was proposed that test range data from the Tactical Air Combat Training System (TACTS) be used as the source of this information. Unfortunately, the knowledge contained in TACTS data is not in a form easily captured by SOAR. The reason is that the TACTS data contain implicit knowledge in the form of examples, while SOAR requires explicit knowledge. In the case-based method used in our IFOR development, our system is designed to use examples, as opposed to explicit knowledge. Thus, we can provide the needed bridge between raw TACTS data and SOAR. Rather than verbally annotating the TACTS data, our knowledge acquisition tool allows the expert to construct cases from the TACTS tapes. These cases could then be used to drive a reactive agent that can respond to a SOAR agent's actions as the SOAR agent learns.

Our accomplishments in support of the IFOR/WISSARD program are evident in the capabilities of the tactics acquisition tool developed to date. In its current form, the tool can be used to acquire tactics through interactions with the domain expert in conjunction with the ModSAF simulator. Using this approach, new tactics have been acquired within a matter of hours. The tool also allows an IFOR agent to exhibit in simulation the acquired tactical behavior. Experiments performed with both our Case-Based agents and SOAR agents have paved the way to improving the SOAR knowledge base, thereby effectively serving as a validation method for the completeness of the SOAR agent.

A detailed discussion of case-based tactics acquisition is presented in Section 3, Technical Approach and Appendix B, with an analysis of the engagement exercises and description of the air domain behaviors descibed in Appendices C and D.

Observations and recommendations for further research in the development of more capable SAFOR, especially for command and control of aggregated units in higher echelons, are discussed in Section 4, Observations and Recommendations.

3. TECHNICAL APPROACH

Among the most significant accomplishments of the CAAT program are the development of two techniques that improve the effectiveness of decision making and that simplify the acquisition of tactics knowledge for computer generated forces (viz. SAFOR and IFOR). We have developed a concurrent control paradigm that vastly improves decision making in the presence of multiple competing objectives. We have also developed case-based learning techniques to simplify the acquisition of tactics through direct interaction with subject experts. Our concurrent control techniques have been applied to both land and air domains. The tactics acquisition techniques have thus far only been applied to the air domain, but they are equally suitable to ground vehicles. This section first describes concurrent control, and then discusses how tactics acquisition methods are built on top of this fundamental control paradigm.

To achieve project goals, we have capitalized on past experience gained from the Army Intelligent Tactical Autonomous Control (ITAC) program, the DARPA Autonomous Land Vehicle (ALV) program, and the Navy Autonomous Control Logic (ACL) program. As an initial effort, concurrent control techniques were used to enhance the low level platform functions (i.e., driver). The choice was made to enhance the existing SAFOR code (viz. SIMNET SAF 6.1.1) in order to minimize development time and to optimize the time available to explore the characteristics of concurrent control. Later, under the WISSARD program, ModSAF code was modified in a similar fashion to allow concurrent control to be used in the air domain.

Concurrent Control

The concurrent control approach to intelligent systems has evolved from considerable experience with autonomous vehicle control and implementation [Payton, Rosenblatt, Kerisey 90]. A concurrent control system is constructed from several individual control loops, each of which is trying to achieve its single goal. As a result, the system emphasizes the independence of the control loops and the complete distribution of control among them. An essential feature of this approach is to allocate as much intelligence to the low level processes as possible. This implies, within the context of SAFOR, that concurrent control is applied to automated vehicle drivers and gunners. However, our current thinking suggests that concurrent control constructs may also be applicable to automated unit commanders ranging from the individual vehicle through to the Battalion level. At these levels, the control loops can accomplish such activities as unit coordination, communication, and resource allocation.

In concurrent control approaches [Payton 86][Payton, Rosenblatt, Keirsey 90][Payton et. al. 91], intelligent action is a manifestation of many simple processes operating simultaneously and coordinated through the context of their environment. Concurrent control provides a means to resolve actions motivated by competing goals by allowing each competing control unit to vote for alternative actions, and then select the most preferable action by combining these votes in an arbitration process as shown in Figure 3. Systems built on this principle exhibit a great deal of robustness because their actions are in direct response to immediate sensory input.

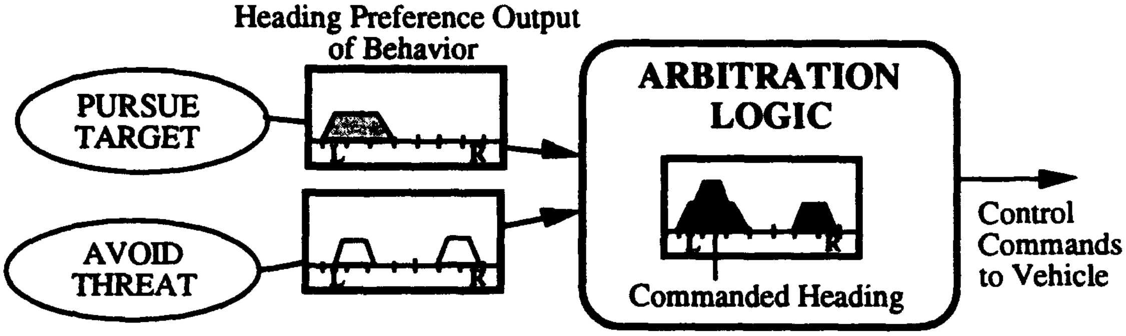


Figure 3. Command arbitration takes the range of acceptable commands from all active control laws and finds a single command that achieves the control objectives.

A concurrent control approach can produce coordinated multi-agent actions in the same way it produces meaningful action in a single agent. The use of simple local control strategies, coordinated through a common environment, can often yield organized collaborative interaction without explicit centralized control. Franklin, for example, has shown in simulation how three predators can be made to chase, encircle, and close in on a prey using only local information [Franklin and Harmon 87]. Each agent in this simulation knows only its relationship to other agents within a very limited range. A very simple set of decision rules is used by each agent to evoke actions in response to the actions of others nearby. Surprisingly, while the resulting actions appear very well orchestrated, no explicit communication between agents is required. More recently, Miller [Miller 90] has suggested a number of simple local strategies that could be used by a team of agents for exploration and sample recovery on Mars. Again, no explicit plan is used, and only minimal communication is required. Much of the work in Artificial Life [Langton 89] also exemplifies this approach. By giving each agent the same set of procedures for how to behave in response to the actions of others, a variety of interesting and useful group behaviors can emerge [Arai, et al 89].

Concurrent Control for SAFOR Implementation

The design of the concurrent control-based tank driver is illustrated in Figure 4. The control loops of the driver obtain information on the status of their respective goals from a vehicle world model which is maintained by the simulation system. This world model corresponds to the concept of virtual sensors [Payton 86].

As shown, three behaviors have been implemented in the driver control loops: avoid_obstacles, follow_route and keep_formation. Driver actions are derived from the input from these multiple behaviors, which are running simultaneously. The individual control loops vote for the speed and heading preferences which are needed to achieve their respective goals. An arbiter then decides which speed and steering commands to issue to the vehicle controller based upon the combination of these votes. The vehicle controller actually changes the vehicle control state through direct interaction with the simulator.

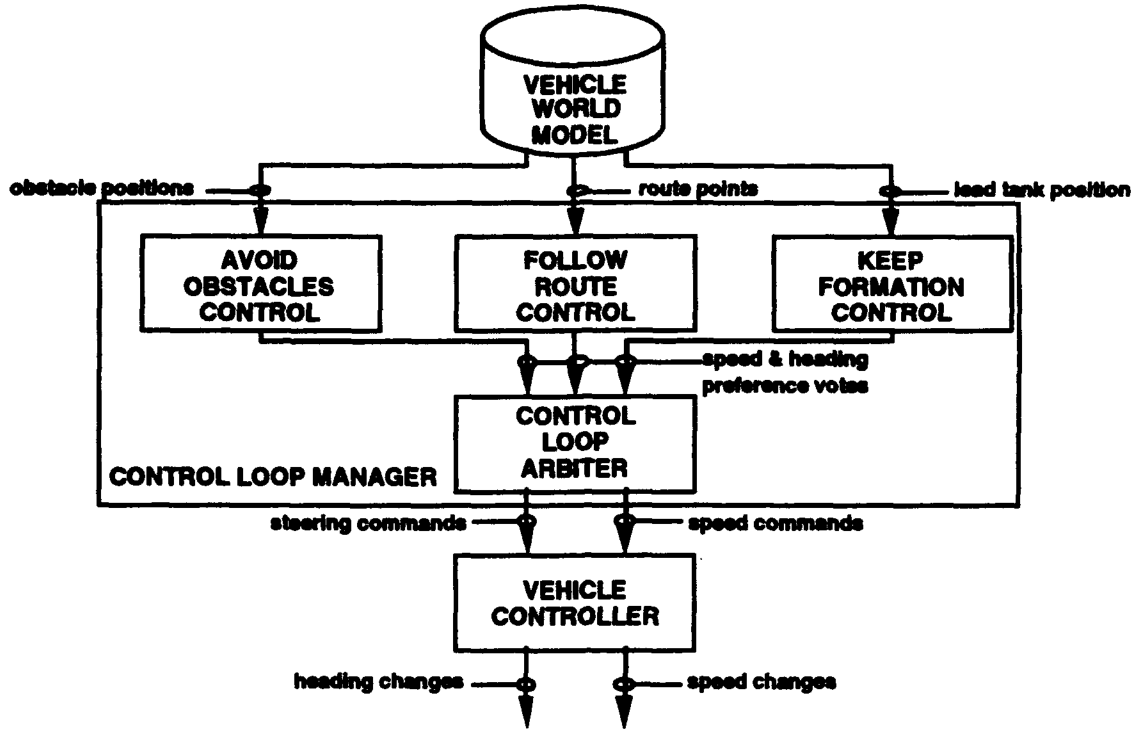


Figure 4. Composition of the Concurrent Control-Based Driver

A control loop manager determines which control loops are appropriate for the prevailing situation. This element provides a rule-based paradigm for the management of the concurrent control loops. As a result, various aspects of nonlinear control can be easily implemented. In addition, the complexity of a concurrent control system can be limited to a tractable level. This component will become more valuable as more control loops are added to the system. This approach enables the modularization of interacting control laws which has proven to be an invaluable implementation aid. Additionally, this approach facilitates the incremental enhancement and improvement of the driver software.

Figure 5 compares the alternative control schemes for SAFOR design. Both schemes organize the overall function as a set of control loops which are represented as C1, C2 and C3. However, the coordination of these separate loops differs significantly. In the finite state machine approach, only one control loop is exercised during any particular cycle through the simulation (i.e., at any one tick). The loop which is chosen is determined by the state of the environment at some instant in time. This choice ensures that only one set of actuator commands is issued at any one time. This design approach is most easily conceived but its debugging can be very difficult. The system can manifest bizarre effects because it can become trapped in inappropriate states or can be responding to nonexistent situations. In addition, the code can become a rat's nest of special cases because of the need to deal with a variety of particular situations.

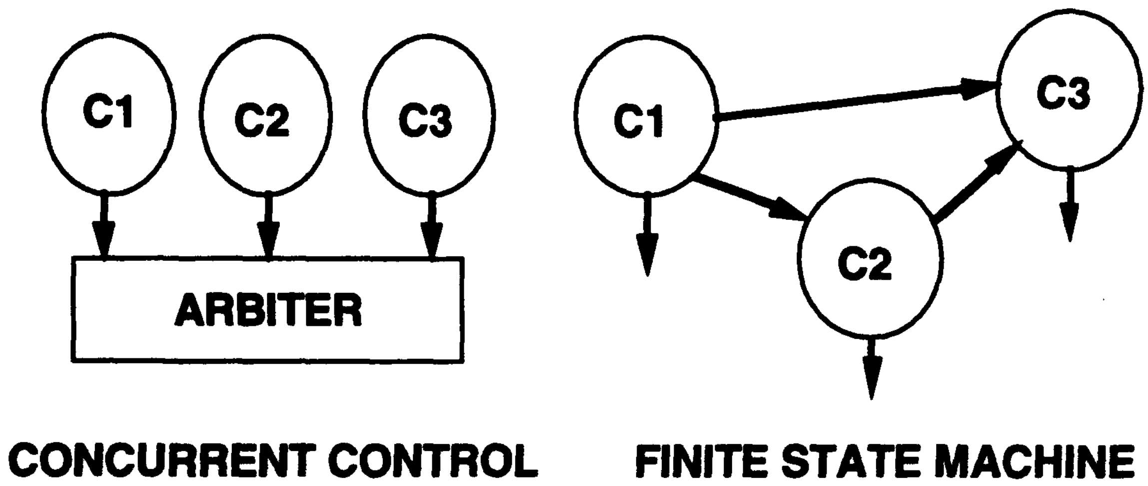


Figure 5. Comparison between Concurrent Control and Finite State Machine Methods

The concurrent control approach exercises all three control loops simultaneously in each cycle through the simulation. Each control loop generates a set of votes for its actuator commands and an arbiter decides which commands to issue once all the votes are received. As a result, the system is always responding to the entire pertinent situation. In addition, it is not possible to become trapped in an inappropriate control state since all the relevant control loops are active in any one cycle. The design of this class of systems is more difficult and less understood than those of finite state machines. However, complex behavior can be generated as a result of multiple interactions.

Extensions to Other SAFOR Components

As discussed, a concurrent control-based tank driver has been implemented for SAFOR. This architecture can be extended to both the SAFOR gunner and commander as well, Figure 6. These extensions can take advantage of much of the concurrent control software which has already been developed and debugged. The control loop manager and arbiter remain the same as that used for the driver, and only additional gunner and commander control loop software needs to be implemented. As with the driver, the gunner will get its sensor input from the vehicle world model. In this architecture, the commander functions somewhat differently from the other control loops, although it, too, can benefit from the components which are common amongst the driver and gunner. Essentially, the unit commander evaluates the status of its task according to the unit orders which have been received, and generates low level goals for the control loops of the gunner and driver. These goals are written into the vehicle world model and, thus, the commander is able to operate asynchronously from the gunner and driver. Although concurrent control techniques can be applied to the commander to achieve a certain degree of realistic behavior, more powerful reasoning techniques are needed to elevate its performance to a level sufficient for complex missions.

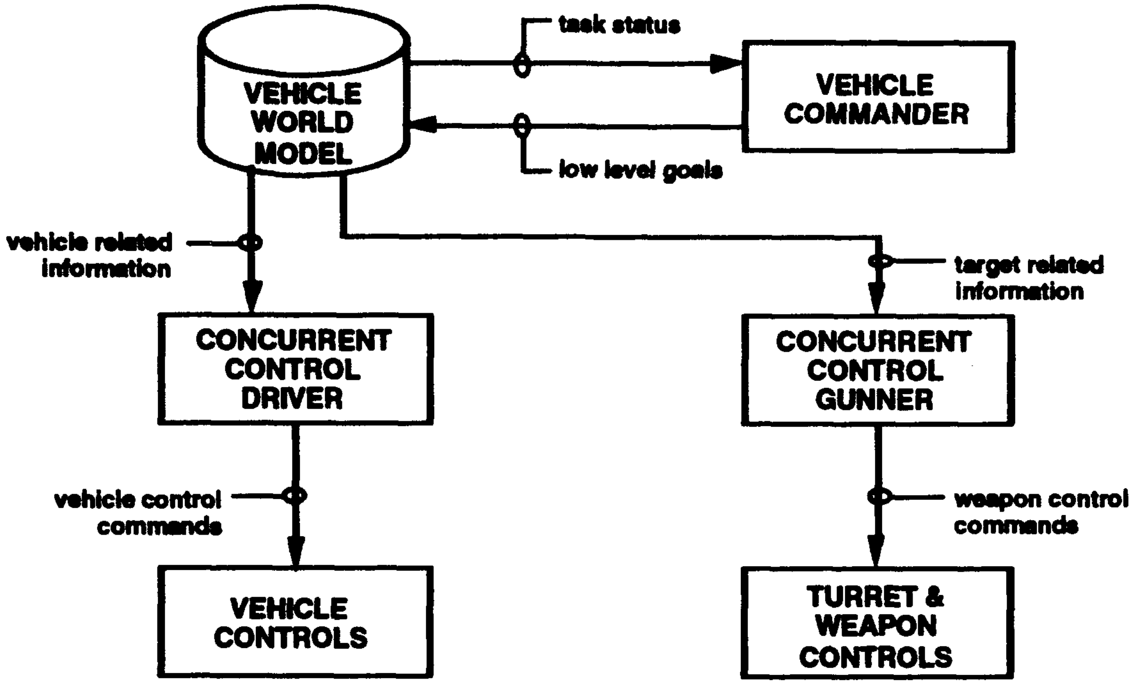


Figure 6. Architecture of an Integrated Concurrent Control-Based Vehicle

SAFOR Accomplishments

Our implementation of a concurrent control SAFOR was based upon enhancements to the existing SIMNET SAF 6.6.1, where an immense amount of effort and ingenuity had already been directed toward developing a realistic SAFOR. Considering the complexity of the problem, the SIMNET SAF performs remarkably well and has many notable attributes. By using this system as a baseline, we were able to focus on issues of behavior-based systems design and cooperating intelligent agents rather than having to duplicate the considerable effort that went into creating a sophisticated real-time simulation. Consequently, our efforts were focused on the modifications needed to produce more realistic SAFOR behavior and carry out more complex exercises. Furthermore, this baseline provides an ideal standard from which we can establish quantitative measures of improvement.

A rigorous technique was developed by which to evaluate the performance improvements of the SAFOR as it was modified. This technique consists of defining several meaningful measures of performance, developing a set of calibrated exercises, establishing the performance of the baseline version of the software, measuring the performance of the modified versions of the software and comparing the measures of performance between the baseline and the modified versions. SAFOR performance was evaluated in twenty exercises, each of which emphasized a different aspect of the ground vehicle driver's performance. Multiple trials of all of the exercises were executed, where feasible, to characterize the magnitude and nature of the stochastic components of performance. The raw data from these exercises was refined into six performance measures which quantified several aspects of realistic driver behavior: number of vehicle collisions, number of building collisions, route efficiency, route efficiency dispersion, mean vehicle speed and mean vehicle speed dispersion.

The quantitative evaluation of the SAFOR driver performance enabled quick identification of the effects of the design changes as they were made. This made it possible to close the design loop and continuously make the driver behavior more realistic. In addition, this systematic evaluation process made very rapid improvements possible since detailed knowledge of the effective direction toward improvement was always available to the designers.

Despite the advances in improving SAFOR performance through concurrent control which were made by this effort, its most important accomplishment has been the development of a rigorous and quantitative methodology for evaluating and comparing SAFOR performance. The six performance measures proved to be valuable indicators of realistic driver performance. These measures enabled the rigorous quantitative comparison with the baseline system to show unambiguously any improvements and degradations caused by the modifications. These measures also assisted the identification of situations which needed correction. This methodology makes the rigorous engineering of SAFOR modifications possible as opposed to relying upon strictly qualitative factors. The change from qualitative to quantitative evaluation makes the systematic assessment of the effects of incremental changes possible. In addition, the defensible comparison of different approaches to the same problem is also made possible. In general, this step moves the development of SAFOR away from an art and closer to a justifiable science. Movement in this direction is necessary to make large scale SAFOR a practical reality. This work has shown that a carefully designed evaluation methodology is an invaluable part of any SAFOR development effort.

Two enhanced versions of the SAFOR were evaluated using our concurrent control methodology. The results of this evaluation technique clearly indicate the improvements of the modified SAFOR over the baseline version. A detailed discussion is contained in Appendix A. In the first enhanced version, all vehicle and building collisions were eliminated and random vehicle movements were reduced by an average of approximately 80%. These improvements were achieved without significantly sacrificing the performance of the baseline version in terms of route efficiency. However, the mean vehicle speed of the first enhanced SAFOR was degraded in almost every exercise by an average of approximately 25%. Analysis of the data indicated that this problem could be overcome with the addition of a ships-passing rule for avoiding oncoming vehicles; the cause was due to the persistent attempt to avoid a collision at the expense of ever slower speeds as the vehicles approached each other. This modification was made and the second enhanced SAFOR version was created and evaluated. This version maintained the absence of building collisions and it reduced vehicle collisions by 99% over the performance of the baseline version. It maintained the same reduction in random vehicle motion as the first version while experiencing a speed reduction of only 2% over the baseline version. A tabulation of the statistical results comparing the enhanced SAFOR versions to the baseline SAFOR is shown in Table A3 of Appendix A.

In summary, the utilization of concurrent control techniques to manipulate SAFOR performance accomplished several goals by:

- demonstrating the effectiveness of concurrent control techniques for significantly improved SAFOR performance

- developing a systematic technique for SAFOR evaluation through the use of rigorous performance metrics

- demonstrating the importance of systematic evaluation in SAFOR design and implementation.

The results achieved not only improved the performance capability of SAFOR vehicles, but also establishes a rigorous and quantitative method of comparing future SAFOR developments. The metrics used in this program, and additional metrics to be generated, will provide a well documented framework in which to evaluate SAFOR improvements.

Automated Tactics Acquisition

Knowledge acquisition is one of the most difficult problems in building intelligent forces (IFOR). The goal is to instill the IFOR with all the knowledge that experts have in performing a task. However, an expert usually will have difficulty expounding relevant and comprehensive knowledge if he is not given a specific problem to solve. We have addressed this problem by partially automating this process with computer-based tools for extracting the tactical expertise from a subject expert and focusing on specific tactical scenarios. Our approach to knowledge acquisition is to display tactical situations to the expert as they occur in computer simulation. The expert then interactively determines the proper tactical decisions that the IFOR should make. While watching the simulation progress, if the expert observes a situation in the simulation that should trigger a new action, the expert stops the simulation and enters the new tactical decision into the knowledge base of the appropriate IFOR.

The knowledge is represented within a case-based system [Hammond 89]. The geometric and tactical information is presented to the expert in the *Simulation Space*. The Simulation Space, as illustrated in Figure 7, is defined as the standard Euclidean space in which objects appear as they might appear in the real world. The state variables describing the motion and geometry of the IFOR at a particular instant in the simulation defines a *case*. Given a particular case observed in the Simulation Space, an expert associates a set of desired actions to that case; these desired actions are referred to as the behavior *response(s)*. The knowledge acquisition procedure displays sequences of cases in the Simulation Space and records the associated behavior responses suggested by the expert for each case. The collection of all cases for a given IFOR is stored in a case database for that IFOR. This constitutes the knowledge base for the IFOR.

The state variables used to characterize each case can be viewed as the orthogonal axes of a multi-dimensional space we call the *Decision Space*. A simplified 2-dimensional Decision Space is shown in Figure 8. From this point of view, a case, with unique values for each state variable, is defined as a point in the Decision Space. Any specific configuration of vehicles in the Simulation Space can be mapped to a specific point in Decision Space. However, a point in Decision Space may map to a variety of configurations in Simulation Space. Through careful selection of the variables that define the axes of the Decision Space, this space may be designed to be invariant to rotation and translation of the configurations in Simulation Space. Therefore, a single point in Decision Space maps to all possible rotations and translations of a particular configuration of vehicles. This allows cases to be independent of the choice of reference frame used for the Simulation Space.

We can use cases to control the behavior of an IFOR by finding the stored case that is most similar to the current situation. Since the current situation is simply a point in Decision Space, our objective is to recall the stored case that is closest to this point. Once the closest case is found, the behaviors specified by that case are applied to the IFOR.

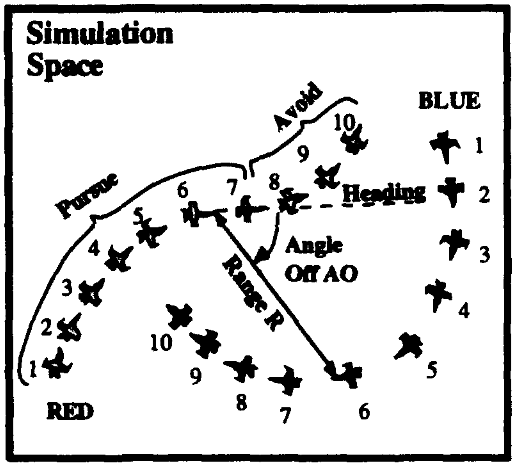


Figure 7. The plan view geometry for an avoid/pursue scenario. The two state variables critical in the decision making of the Red Fighter are the Angle Off (AO) and the Range (R), which are identified here at simulation step 6.

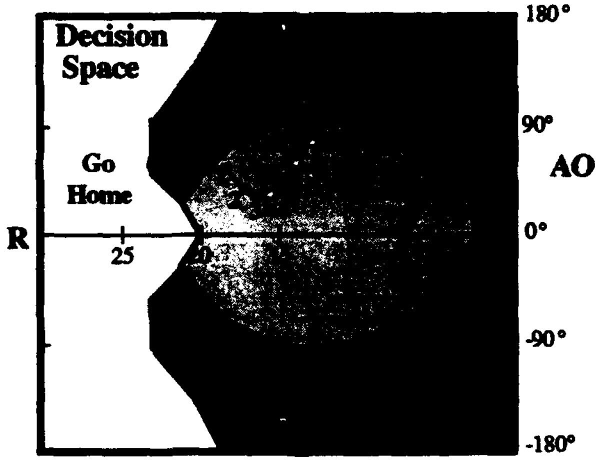


Figure 8. The Decision Space is a space spanned by the two state variables Angle Off (AO) and the Range (R). For any point in the Decision Space, an action of Avoid, Pursue, or Go Home is acquired from the tactics expert.

To illustrate the process of knowledge acquisition, consider a constant altitude avoid/pursue scenario. Figure 7 shows a plan view of the geometry in Simulation Space. The knowledge we wish to acquire depends solely on the relative orientations, distances, and speeds of the two vehicles. For the sake of simplicity, assume that only two state variables are needed for decision making, the Angle Off (AO) and the Range (R) of the Blue Bogey with respect to the Red Fighter. This results in a 2-dimensional Decision Space as shown in Figure 8, where one axis is Angle Off (AO) and the other axis is Range (R).

For any point (R, AO) in the Decision Space, we will want to use cases to specify a corresponding behavior for the Red Fighter. For this example, assume we have a choice of only three behaviors: Avoid the bogey, Pursue the bogey, or Go Home. Ideally, the Decision Space will be partitioned as shown in Figure 8 in order to produce the behavior exhibited in Figure 7. However, to obtain this partitioning, it will be necessary to capture an appropriate set of cases.

When the expert begins the process of knowledge acquisition, there are no cases at all. The exercise begins with the expert specifying the first case. In the initial configuration seen by the expert in Simulation Space, the Blue Bogey is too far from the Red Fighter. Consequently, the expert indicates that the correct behavior is "go home." This establishes the first case c1. As shown in Figure 9, the geometry in Simulation Space shows a range (R) of 25 Nautical Miles (NM) and an angle off (AO) of 0 degrees. This corresponds to the point c1 in Decision Space. If the expert were to allow the Red Fighter to be controlled by the case-based system at this time, the plane's response would always be "go home" because c1 is the only case available to match.

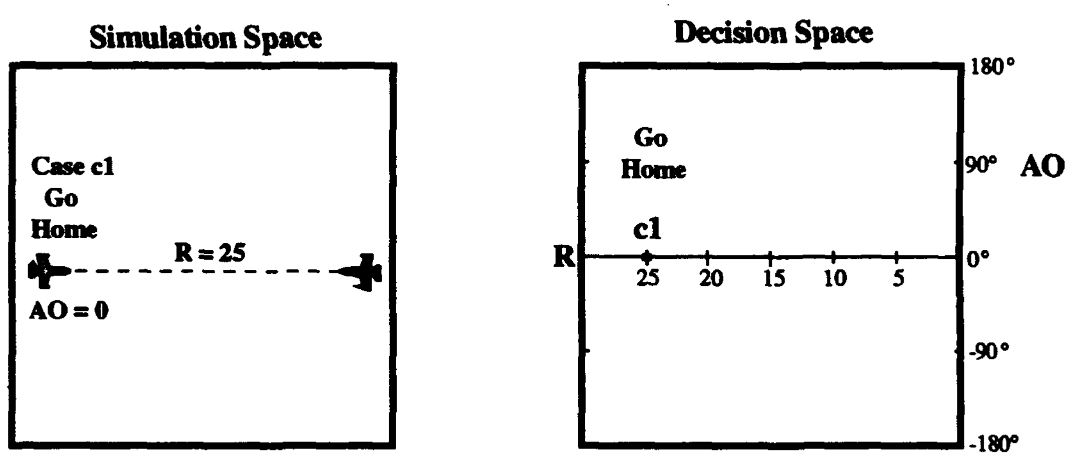
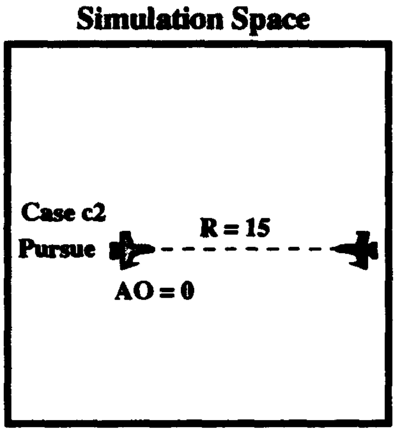


Figure 9. The first case is stored based on features obtained from the Simulation Space. This case corresponds to one point in the Decision Space.

As the simulation is allowed to progress, the Blue Bogey will get closer to the Red Fighter in Simulation Space. When the two planes get close enough to one another, the expert will decide that the Red Fighter should "pursue." The expert will then enter a new case into the case database for the Red Fighter. The new case, c2, uses the new values of R = 15NM and AO = 0, as obtained from Simulation Space (Figure 10).

As shown in Figure 10, the two cases c1 and c2 divide the Decision Space into two regions. In the region containing case c1, the IFOR will be commanded to "go home," and in the region containing c2, the IFOR will be commanded to "pursue." The dividing line between these two regions is the set of all points that are equidistant to the two cases. This line is called a Voronoi edge, and the partitioning of the Decision Space may be described by a Voronoi Diagram [Preparata and Shamos 85]. A Voronoi Diagram for a set of N point cases in the Decision Space plane is a partitioning of the plane into N polygonal regions, with one region associated with each case. Each point within a

Voronoi Region is closer to the case point for that region than it is to any other case point in the Decision Space.



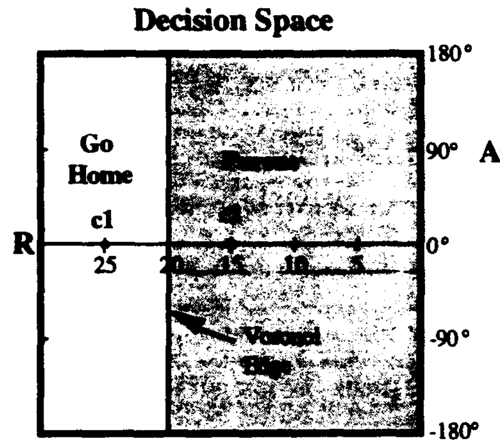
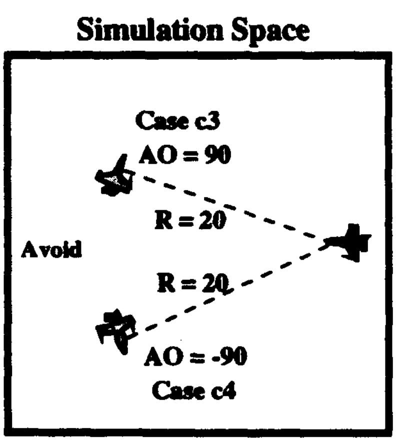


Figure 10. Defining a second case in the Simulation Space will partition the Decision Space into two parts. When the state of the Red Fighter is closer to case c1, then the Red Fighter will go home, and when closer to case c2, the Red Fighter will pursue.

Using his control over the simulation, the expert may now explore alternative configurations of the Blue and Red planes. By appropriate maneuvering of the Blue Bogey, the expert is able to create a situation in which the Red Fighter should do an avoidance maneuver. This leads to the addition of two more cases to the case database of the Red Fighter. As illustrated in Figure 11, the symmetric cases c3 and c4 define states in which the Red Fighter should "avoid" the Blue Bogey. The addition of these two cases results in a new Voronoi partitioning of the Decision Space. Note that these new cases significantly alter the regions for "go home" and "pursue" that were established in figure 10. The expert will now need to add more cases to correct for some of these changes.



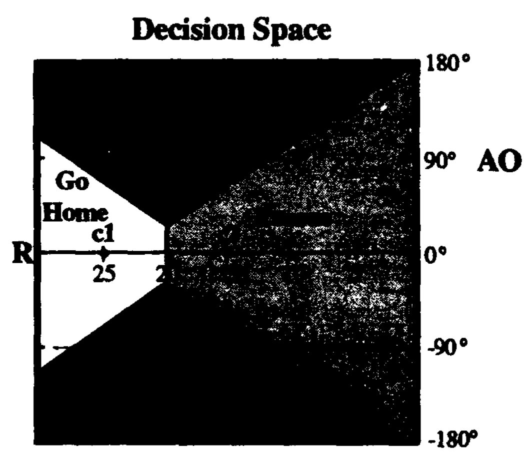


Figure 11. Using symmetry, a new configuration in Simulation Space can lead to the addition of two new cases. These cases define two new regions in the Decision Space that identify configurations in which the Red Fighter should "avoid" the Blue Bogey.

The process of adding more cases to the case database of the Red Fighter is repeated until the expert has achieved the desired partitioning of the Decision Space. Figures 12 and 13 illustrate several new cases that might be added. In Figure 12, the expert defines an additional pair of cases for the Red Fighter to "go home" when it would otherwise have performed an "avoid" maneuver. This corrects for the changes made to the "go home" region of Decision Space when the "avoid" cases c3 and c4 were added previously. In Figure 13, two cases are used to extend the "avoid" region of Decision Space into an area that would otherwise have directed the IFOR to "pursue." As more cases are added, the decision boundaries between the "avoid," "pursue," and "go home" alternatives can become arbitrarily complex. With a sufficient number of cases, the Decision Space will ultimately approach the partitioning shown in Figure 8.

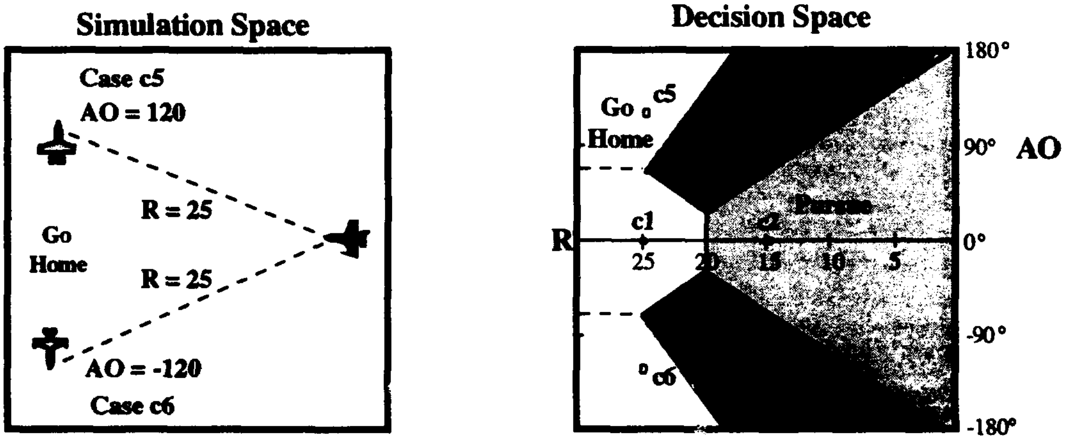


Figure 12. Defining two more cases in the Simulation Space further refines the partitioning of the Decision Space.

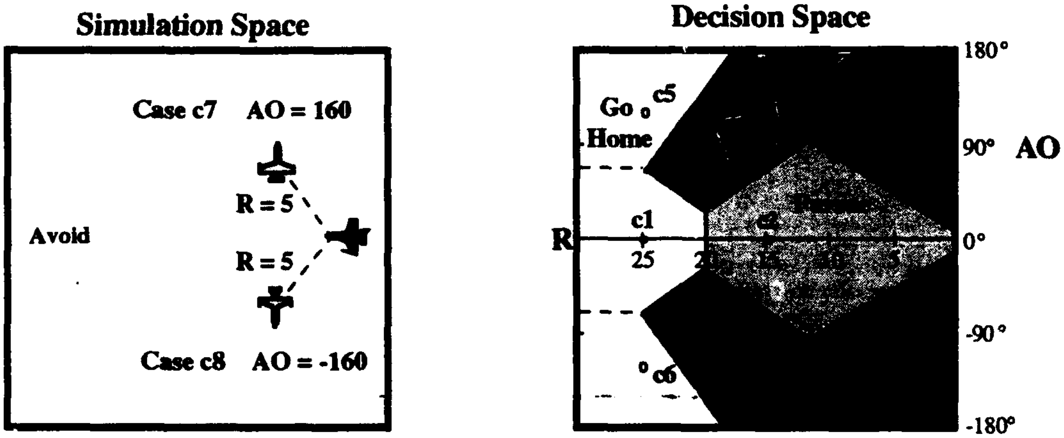


Figure 13. As more cases are added, the shape of the regions in Decision Space will change accordingly, allowing for arbitrarily complex decision boundaries.

After an initial set of cases has been acquired, a simulation of the Red Fighter can be run against the Blue Bogey to investigate the current decision logic modeled in the case database of the Red Fighter. If the simulation indicates a fault in logic or if transitions between cases are not occurring as desired, the simulation can be halted, and a new case can be added to the case database using the geometry in the simulation. This iterative method for including new cases from repeated simulations is a feasible method of fine tuning the partitioning of the Decision Space.

This method of semi-automated knowledge acquisition is also applicable to Decision Spaces of higher dimensionality. In fact, our case-based IFOR has successfully used up to 36 states, giving the Decision Space a dimensionality of 36. Although we cannot view this Decision Space, this is not a problem for knowledge acquisition because all user interaction occurs through the two-dimensional Simulation Space.

Behavior-Based Intelligent Agents.

In the implementation of our knowledge acquisition system, a concurrent control approach is used to create the fundamental repertoire of behaviors that may be selected through case-based reasoning. Our concurrent control foundation allows the expert to choose generalized actions such as "Pursue Target" or "Avoid Threat" as the desired response to a given situation or case. Concurrent control allows a simple action specification such as "Pursue Target" to result in a rich set of possible maneuver responses. This is due to the fact that an agent's responses are expressed in terms that are relative to the agent's surroundings. For instance, the way an agent pursues a target will depend on the actions of the target. This has two important benefits. First, it provides the expert with a level of specification that is meaningful despite noticeable variations in the agent's surroundings. Second, it simplifies the Decision Space into larger, more homogeneous regions, so that fewer cases are needed to specify a complex tactical maneuver.

Cases control actions by selecting the on/off state and the relative priorities of component control laws within our concurrent control architecture. Figure 14 illustrates this architecture. The stored case that most closely matches the current state of the tactics geometry turns the appropriate behaviors on or off. In many situations, control laws are set to an intermediate on state, giving them only partial influence over the ultimate control decision. This is done by assigning a weight between zero and one for the output of each control law.

The control laws that are on at any given time operate concurrently, issuing commands simultaneously. Since only one heading, velocity, and altitude command can actually be sent to control an aircraft, our arbitration logic is used to perform command fusion [Rosenblatt and Payton 89]. The arbitration process occurs independently for heading, velocity, and altitude commands. A significant point to note is that knowledge acquisition is raised to a level of specifying tactical maneuvers and actions, rather than low level stick commands.

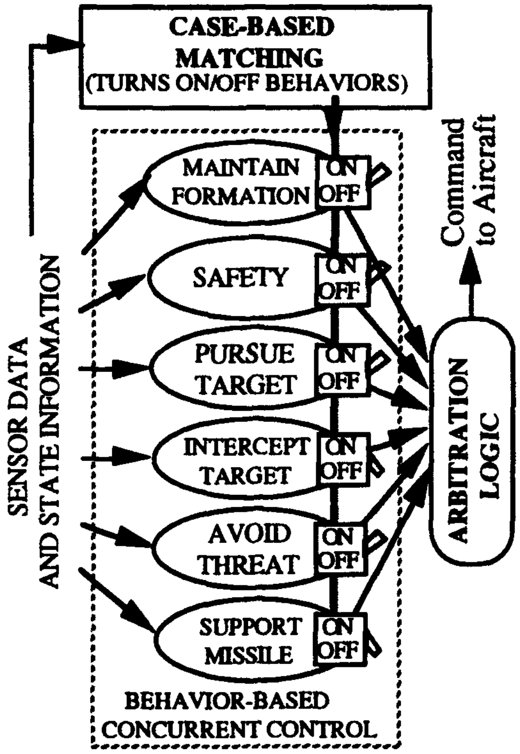


Figure 14. Case-Based Matching controls which control laws are ON or OFF. All the ON control laws concurrently issue commands to the arbitration logic unit.

In our case-based system, behaviors have been created for several fundamental air-domain actions. Specifically, the following behaviors were implemented: Attain Heading, Attain Speed, Attain Altitude, Maintain Formation, Safety, Pursue Target, Intercept Target, Avoid Threat, Avoid Collision, Maintain CAP Maneuver, Fire Missile, and Support Missile. Various combinations of these behaviors provide the necessary controls for a wide variety of interesting tactical maneuvers.

Selective Focus-Of-Attention

In complex multi-agent scenarios, the number of possible configurations of vehicles can become far too complex to allow for meaningful selection of cases or for consistent behavior execution. It is therefore necessary to organize an intelligent agent's perception of the world in terms of a more consistent and stable set of features. For example, by devising a means for determining the best target for a vehicle to pursue at any given time, a "Pursue Target" behavior can be implemented without our having to consider all possible configurations of enemy vehicles. Similarly, if cases are selected according to metrics such as the relative orientation between a vehicle and its "best target," then the addition of new vehicles to a scenario does not significantly complicate the Decision Space used for knowledge acquisition.

The use of stable features such as those described above provides a means for modeling aspects of selective focus-of-attention that are common to many human decision-making tasks. Once labels such as "best target" or "nearest threat" are assigned to specific objects in an agent's environment, the agent will attend to these objects explicitly. The labels used identify objects in accord with the critical roles these objects play in the IFOR's attack or survival responses. By repeatedly applying user-defined rules to assign labels, the IFOR is able to respond to changes in its environment. As the tactical situation evolves, yielding new threats or new target opportunities, different objects may be assigned these labels. This corresponds to a shift in attention as new information becomes available.

In our approach, we consider the labels themselves, such as "best target" or "nearest threat," to be objects. Using the concept of markers, as described by Agre and Chapman, our system labels an object by placing a marker on it [Agre and Chapman 87] [Agre 88]. Each marker has associated software that allows it to track the object, as well as to characterize properties of the object, such as its estimated range, average speed, and heading. Thus, when an IFOR finds the vehicle that is its best target, it places its "best target" marker on that vehicle. The marker will then follow the movements of that vehicle until another vehicle is selected as the best target. At this time, the marker will be moved from one vehicle to the other.

Markers need not be associated only with observable objects. It is possible to use "hypothesized markers" for objects that are predicted but not seen by the IFOR [Payton and Dolan 91]. If a given situation suggests that an opponent may be hidden from view, or disguised, it is possible for our agent to hypothesize the presence of that opponent. Therefore, if the current situation suggests that a vehicle is hidden behind an obstacle, or outside of the current field of view, the agent can still assign a marker to the predicted location of that vehicle and thereby behave as if that vehicle were actually observable.

In the air domain, as shown in Figure 15, markers are used within a retinocentric reference frame. Markers are initially associated with visible objects such as neighboring target and threat vehicles. As visible vehicles move relative to the observer, the markers are moved accordingly. Over time, the markers measure average speed and heading of the objects they track so that if an observed vehicle moves out of an agent's field of view, that vehicle's position can still be estimated. This way, even if the real target disappears from view, the "best target" marker will continue to estimate that target's position until it can be re-acquired.

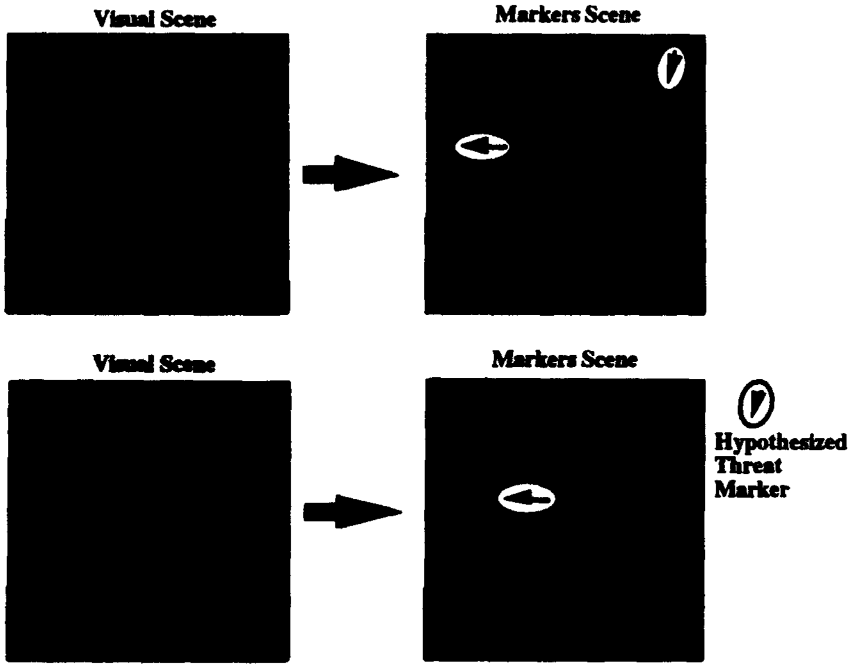


Figure 15. When a fighter sees the bogeys in the visual scene, markers track these agents. If the marked agent leaves the visual scene, then the marker becomes a hypothesized marker until the bogey is seen again or until enough time elapses that the bogey is considered out of the current field of interest.

In multi-agent scenarios, we apply the same principles, but use more markers. In the example shown in Figure 16, an agent not only uses markers for threats and targets, but also for its own missile, its wingman, and its support formation. In many cases, two markers can refer to the same sensed object. For instance, the current target can also be the current threat.

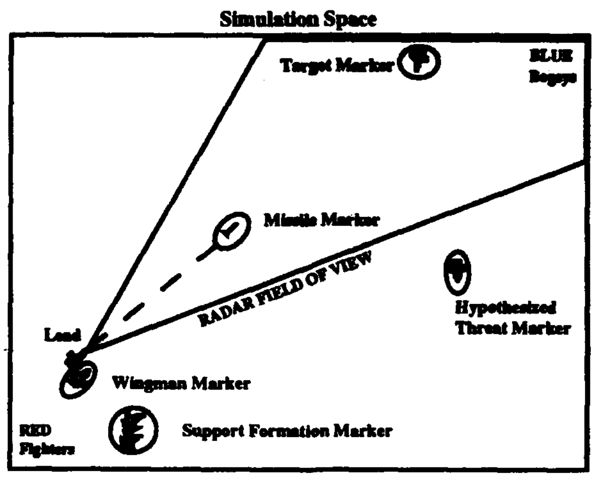


Figure 16. This Simulation Space view shows how markers are used for the wingman, the support formation, a fired missile, the current target, and the current threat.

In the ground domain, markers can be used in a similar fashion to track targets, identify threats, and to maintain proper coordination with members of the same company or platoon. As shown in Figure 17, an observed formation of three enemy vehicles may lead an agent to predict the presence of a hidden vehicle. The agent viewing this scene does not really know whether or not there is a vehicle hiding behind the hill. The formation of the observed vehicles, however, may suggest that this possibility exists. By conceptually placing an "ambushing vehicle" marker behind the hill, our agent will be able to behave with appropriate caution.

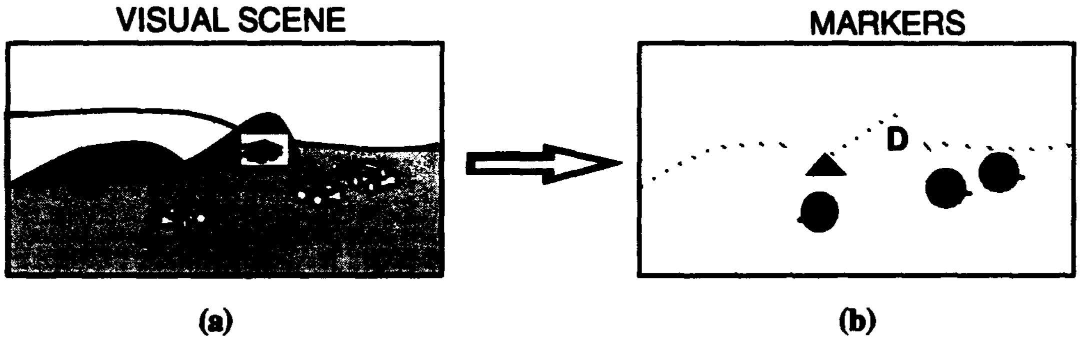


Figure 17. A scenario using a hypothesized marker "D" to represent the hypothesis that an ambushing vehicle waits behind a hill.

IFOR Accomplishments

Our accomplishments in support of the IFOR/WISSARD effort are evident in the capabilities of the tactics acquisition tool to date. The tool currently can be used to acquire tactics through interaction with a tactics expert during the course of a controlled simulation exercise. Complex new tactics can be acquired in a matter of hours. It also allows an agent to exhibit in simulation the acquired tactical behavior. This has allowed us to use our tool to help refine the knowledge bases used in other non case-based agents such as the SOAR agents developed for WISSARD. We do this by providing a consistent yet reactive opponent for the SOAR agents to fly against. By watching how the SOAR agent fails in various trials, we are able to quickly identify deficiencies in the SOAR agent's knowledge base. When these deficiencies are corrected, we are then able to validate the correction with the same consistent responses. We have evaluated our tool according to two criteria. The first was the ability to capture complex tactics rapidly; the second was the ability to use our tool to help improve the knowledge base of a SOAR agent.

Our knowledge acquisition tool has been used to capture tactics for a variety of multi-agent scenarios. During tool development, an emphasis was given to 1v1 scenarios. However, to test the flexibility of our approach, we have also captured tactics for several 2v4 scenarios. These scenarios required various forms of cooperation between agents. For example, as illustrated in Figure 18, our tool has successfully captured the coordinated tactics needed to enable two pairs of defensive planes to pursue an incoming bogey and to execute alternative attack and evasion maneuvers in response to the bogey's actions. This has been the most complex scenario implemented, and requires coordination between both pairs of planes as well as between each leader and his wingman.

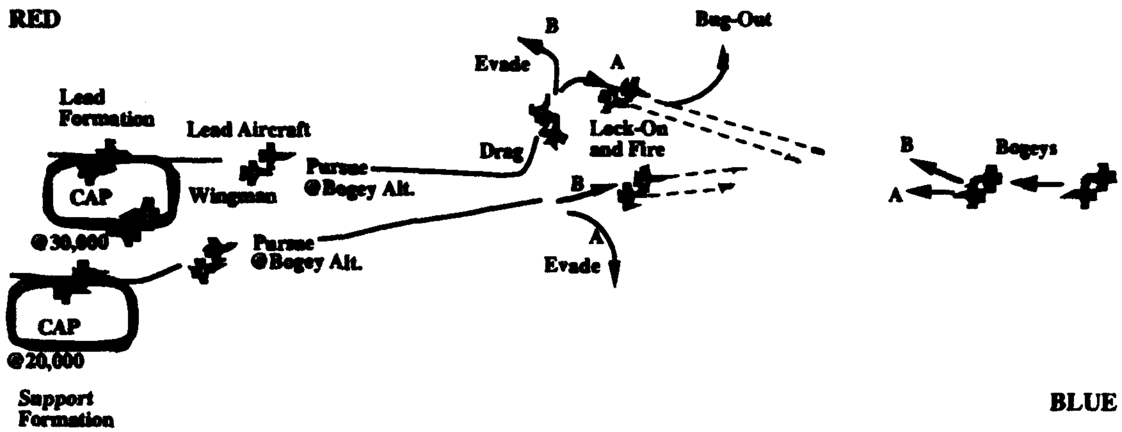


Figure 18. The above complex 2v4 tactical scenario involving cooperation between multiple vehicles has been successfully captured using our case-based acquisition tools.

4. OBSERVATIONS AND RECOMMENDATIONS

Although computer generated forces (CGF) such as SAFOR and IFOR have been reasonably successful in simulating acceptable human behavior at the platform and weapons control level, further advances in CGF capabilities will not be possible until progress is made to automate the command and control functions. This is one of the primary hurdles that must be cleared in order to achieve force aggregation to the higher echelons of command. The automation of command and control, unfortunately, is one of the most difficult challenges which face CGF technology today. Successful automation of command functions involves many of the hardest problems in artificial intelligence, including situation assessment, planning, knowledge representation, and intelligent control.

During the performance of the CAAT program, and based on our participation in other intelligent agent development, we have made several observations on the current trends and status of CGF development:

- At present, the existing simulations are at the platform level (e.g. M1, M2, artillery, fixed wing, rotary wing, missile, etc.). Although there is on-going work at producing higher levels of CGF, the only definitive design and implementation has been the Hughes C^2 SAFOR for controlling units up to the Company level.
- Platform level entities are providing acceptable performance in executing moderately complex missions, with ModSAF currently being the most advanced simulation framework for this purpose.
- The most ambitious CGF programs, ModSAF and BDS-D CGF, are still in the early stages of development; many of the capabilities are still at the platform system requirements level.
- The ability to execute complex missions is not well developed, primarily due to the limited deep-reasoning capability of the entities and the lack of automated command and control capability.
- The need for CGF command and control is critical in evolving to the higher echelons of battlefield units, and in solving the force aggregation problem in simulation.
- Current simulation programs are not addressing the overall joint services' needs in the simulation systems.

Based on our experience in developing SAFOR and IFOR technology for the CAAT contract, we recommend the following issues to be addressed in future CGF developments:

- The design and implementation of a canonical CGF commander model which is valid at various levels of command structure. The model must allow the commander's processing tasks to be partitioning into symbolic reasoning and reactive response components. It also must support the communications between superiors and subordinates by means of battle orders.

- The ability of the CGF commander to accept mission requirements issued by either a human operator or another CGF commander. This requires that a representation be selected that is common to both modes of operation, to allow for the interpretations of CGF actions in the language understood by human operators.

- A design which provides the flexibility to enable a human operator to access any level of command within the CGF structure in order to supplement CGF capabilities by directly controlling the CGF, when necessary.

- Specialized reasoning modules for such activities as interpreting mission objectives, performing situation assessment, interpreting tactics and doctrine, and executing commands. Wherever possible, the design should take advantage of the many well developed inferencing, interpretation, and reasoning techniques available from related disciplines.

- Representations of tactics and doctrine in easily accessible and modifiable data bases; Such knowledge should not be encoded into the CGF programs.

Using this approach to develop the command and control framework of CGF, we believe that faithful representations of authentic command and control CGFs can be achieved. Additionally, it will preserve design options so that future force structure and organization changes can be modeled without massive alterations to the CGF software.

Appendix A: SAFOR PERFORMANCE ANALYSIS
Evaluation of Concurrent Control

While many opportunities exist for improving current SAFOR technology, our effort has concentrated on making the behavior of SAFOR more realistic. Realism can be simply qualitative in nature but that choice makes it an elusive goal. Instead, we have concentrated on developing quantitative metrics for evaluating SAFOR realism. Using these metrics, we have endeavored to compare the effectiveness of our new concurrent control techniques relative to the finite state machine methods that are predominantly in use today.

We expended a considerable amount of effort to develop a systematic methodology by which to quantify and evaluate the realism of SAFOR performance. This methodology was then used to assess the success of the modifications which were made to the SAFOR driver software. In this methodology, the modified version of the SAFOR was compared with a baseline version.

The baseline version was created by porting the SAFOR software, which was delivered with SIMNET Version 6.0 to the Government, to a Sun 4. This software was then debugged to remove any obvious logical and programming errors and inconsistencies. The modified version of SAFOR consists of the baseline SAFOR with the concurrent control-based driver, as described above, replacing the driver part of the ground vehicle component. The comparison was performed by executing several standardized exercises with both versions and collecting data from each. This data was then reduced to several meaningful performance measures which permitted evaluation of the effects of the modifications.

Carefully Designed Calibrated Exercise Set

Twenty exercises were carefully designed to enable the analysis of the results and the isolation of the effects of different conditions as much as possible. Each exercise involved traveling a designated route under the various conditions. The details of these conditions are described in Table A1. The essential differences between these exercises are shown in Figure A1.

In Figure A1, the numbers in square brackets, [ ], indicate the exercise number. In exercises where platoon-sized units were involved and where appropriate, two different formations were used, column and line (designated in Figure A1 by the suffix "R" for roadmarch and "A" for assault, respectively). The unit sizes for these exercises ranged from single tanks to a single company of vehicles.

The first partition of the exercises is between those which involve interactions with buildings and those which involve interactions between vehicles. The small building is approximately the size of a single tank whereas the large building is larger than the distance between two vehicles in formation. In exercises with the large building, two different routes were tested, one which passed through the building and another which followed the perimeter of the building very closely.

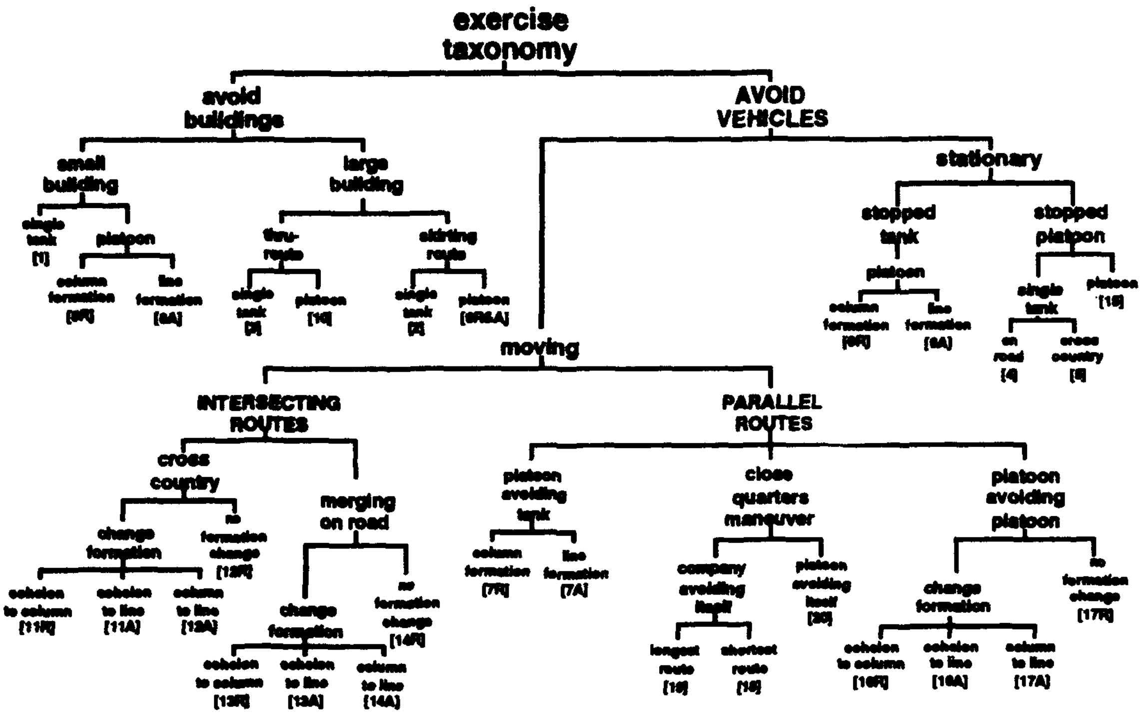


Figure A1. Taxonomy of SAFOR Evaluation Exercises

Exercises were constructed which explored interactions between both stationary and moving vehicles. The stationary vehicles included a single tank and a platoon of four tanks. Two types of routes were considered for moving vehicles, intersecting and parallel. Intersecting routes involved both cross country intersections and merging on roads. Parallel routes included both vehicles approaching one another (e.g., as on a road) and moving in the same direction in coordinated fashion (e.g., as in formation).
In these exercises, both blue and red force used the same combat instruction set (CIS). The roadmarch CIS used the column formation with maximum speed of 25 km/hr. The assault CIS used the line formation with maximum speed of 25 km/hr. This set of exercises includes all aspects of simple driving and emphasizes the nature of the three concurrent control loops which are described in the main body of this report.

Establishment of Baseline Performance

The baseline performance was first established to form the performance reference against which all successive modifications would be compared. This step is absolutely necessary to develop the measure by which the effects of all changes can be accurately gauged. Developing the baseline performance gave immediate experience with the exercises and the process of collecting performance data. Several changes to the exercise set were made as a result of this effort. In addition, many problems in instrumenting the software for accurate data collection were solved. As a result, building the baseline performance data set not only generated a numerical standard with which to compare the performance of the modified SAFOR but also provided the procedures by which to collect the data.

Table A1. Characteristics of the Calibrated Exercises

Exercise Number	Formation Used	Exercise Description	Situation Studied
1	column & line	a single blue tank follows a cross country route which passes through a small building	single tank avoiding a small obstacle
2	column & line	a single blue tank follows a cross country route which goes around a large building	single tank skirting a large obstacle
3	column	a blue tank follows a cross country road which passes through a large building	single tank avoiding a large obstacle
4	column & line	a blue platoon is stopped on a road in column formation; a red tank follows a road route which passes between two blue tanks	single tank avoiding static vehicles head-to-head on a road
5	column & line	a blue platoon is stopped in column formation & a red tank follows a cross country route which passes through the blue platoon	single tank avoiding static vehicles facing a different direction on a cross country route
6	column & line	a blue platoon starts column formation & follows a road route which passes a single red tank which is stopped on the road	platoon avoiding a stopped vehicle while maintaining formation
7	column & line	a blue platoon starts in column formation & follows a road route while a red tank follows a road route in the opposite direction; the blue platoon & red tank meet each other somewhere on the road	platoon maintaining formation while avoiding a single tank moving on the same route in opposite directions
8	column & line	a blue platoon starts in echelon right formation & follows a cross country route which passes through a small building	platoon changing formation then avoiding a small obstacle while maintaining formation
9	column & line	a blue platoon starts in echelon right formation & follows a cross country route around a large building	platoon changing formation then skirting a large obstacle while maintaining a formation
10	column	a blue platoon starts in echelon right formation & follows a cross country route which passes through a large building	platoon avoiding a large obstacle while maintaining a formation & following a route

Table A1. Characteristics of the Calibrated Exercises (continued)

Exercise Number	Formation Used	Exercise Description	Situation Studied
11	column & line	a blue platoon starts in echelon left formation & follows a cross country route while a red platoon starts in echelon right formation & follows a cross country route in the opposite direction; these platoons meet each other somewhere on the route	two platoons changing formation then avoiding each other while following intersecting cross country routes & maintaining formation
12	column & line	a blue platoon starts in column formation & follows a cross country route while a red platoon starts in line formation & follows a cross country route in the opposite direction; these platoons meet somewhere on the route	two platoons avoiding each other while maintaining formation & following intersecting cross country routes
13	column & line	a blue platoon starts in echelon left formation & follows a road route while a red platoon starts in echelon right formation & follows another road route which merges with the blue road; these two platoons merge with each other somewhere on the road while traveling in the same direction	two platoons changing formation then merging on the same road while maintaining formation
14	column & line	a blue platoon starts in column formation & follows a road route while the red platoon starts in line formation & follows another road route which merges with the blue road; these two platoons merge with each other somewhere on the road while traveling in the same direction;	two platoons merging on the same road while maintaining formation
15	column & line	a red platoon starts in line formation & follows a road route which passes a stopped blue platoon	platoon maintaining formation while avoiding a stopped platoon on a road

Table A1. Characteristics of the Calibrated Exercises (continued)

Exercise Number	Formation Used	Exercise Description	Situation Studied
16	column & line	a blue platoon starts in echelon left formation & follows a road route while a red platoon starts in echelon right formation & follows the same road route in the opposite direction; these two platoons meet each other in opposite direction somewhere on the road	two platoons changing formation then passing each other in opposite directions while following a road & maintaining formation
17	column & line	a blue platoon starts in road formation & follows a road route while a red platoon starts in echelon right formation & follows the same road route in the opposite direction; these two platoons meet each other in opposite direction somewhere on the road	two platoons passing each other in opposite directions while following a road & maintaining formation
18	column	a blue company starts in line formation & follows the cross country route	a company maintaining formation & following a cross country route
19	column	a blue company starts in a line formation & follows the cross country route which begins at the end of the starting company formation; this route requires the company to follow a serpentine route which passes itself	a company maintaining formation, following a cross country route while avoiding other vehicles in the company
20	line	a blue platoon starts in line formation & follows a cross country route with sharp right angle turns and smooth right angle turns	a platoon executing maneuver with close quarters turns while maintaining formation

The process of determining the baseline performance allowed us to taylor and refine the design of the standard exercises. Once the exercises were established then the baseline code was exercised again to create the final reference. These results formed the performance baseline. Qualitative observations were also made while exercising the baseline. These observations indicated a number of areas where improvement was necessary and provided some guidance for the design of the modifications. These observations are presented in Table A2.

Table A2. Qualitative Observations of the Baseline Performance in the Calibrated Exercises

Exercise Number	Formation	Qualitative Observations
1	N/A	none
2	N/A	the red tank hit the large building when it followed the skirting route
3	N/A	the red tank drove through instead of around the large building although it successfully avoided the small building in Exercise 1
4	N/A	the red tank detected 3 blue tanks; only two of them were parked on the road; the third one was parked next to the road; the red tank did pass a route point between the two blue tanks parked on the road
5	N/A	the red tank detected only one blue tank which was parked in the middle of the cross country route
6	both	none
7	both	none
8	column	when the leading tank collided with the small building all the following tanks stopped or turned until the leading tank disengaged from the small building
	line	the leading tank followed the cross country route & drove through the small building after several collisions
9	column	some of the tanks hit the building when following the skirting route
	line	two tanks tried to drive through the large building; when the leading tank drove inside the building, the following tank turned 180 degrees for several minutes before it also drove into the large building
10	column	all the blue tanks drove into the large building
11	column	two tanks collided with each other when the blue platoon changed from echelon left to column formation; those collisions caused the other two tanks to turn wildly until the colliding two tanks successfully disengaged
	line	the specified route required a 90 degree turn; when the blue platoon passed through that turning point two of the four tanks lagged & either moved out of formation position for a time or collided with each other
12	line	2 tanks collided with each other when the blue platoon changed formation from column to line formation; when the blue platoon passed through a 90 degree turn two of the four tanks lagged & either moved out of formation position for a while or collided with each other
13	column	at the beginning, a lot of collision avoidance action occurred between the blue tanks

Table A2. Qualitative Observations of the Baseline Performance in the Calibrated Exercises

Exercise Number	Formation	Qualitative Observations
	line	at end of this exercise both platoons approached the small building simultaneously & some of the tanks turned wildly before stopping
14	both	the red tanks sometimes collided with each other
	line	when changing from road to line formation, a red tank collided with other tank. During the road following, in one case, a blue tank collided with a red tank and never disengaged from the collision, the followers of the both tanks also stopped
15	column	when the leading red tank collided with the leading blue tank which was parked in the middle of the road the rest of the blue tanks turned and ran
	line	when the leading red tank collided with the leading blue tank, all the tanks went crazy for sometime until those two tanks successfully disengaged
16	line	when the red platoon made a left turn (approximately 90 degrees) along the road route, one of the red tanks was out of formation for a time
17	both	none
18	column	many many collisions occurred; in addition, several tanks turned randomly & moved out of formation
19	column	chaotic behavior similar to the results of Exercise 18
20	line	some blue tanks lagged behind, headed in the wrong direction & collided with other tanks; if the SAF vehicle has its own intelligence then it can modify the route or change the formation before and after the sharp turn; without a flexible intelligent module, the SAF vehicle will never be able to act effectively in this exercise

Quantitative Performance Measures

Both versions of the SAFOR program were instrumented to record several different types of raw data. The raw data included number of vehicle collisions for each vehicle, number of building collisions for each vehicle, total distance traveled during the exercise by each vehicle, total time required to traverse the designated route and total length of the designated route. Whenever possible, each exercise was run ten times by both the baseline and modified programs to characterize the nature of random influences. Statistical variation was observed due to the influence of the operating system (i.e., UNIX) and to such effects as round-off errors. The multiple trials were initially formulated in the exercise design to identify the significance of performance differences which were observed. However, they also served another role which is discussed below.

The raw data were used to compute six performance parameters: number of vehicle collisions, number of building collisions, route efficiency, route efficiency dispersion, mean vehicle speed and mean speed dispersion, where route efficiency, RE, is defined by

$$RE = PL / DT$$

with PL = length of the designated path and
DT = total distance traveled by the vehicle

and where parameter dispersion, PD, is defined by

$$PD = PS / PM$$

with PS = standard deviation of the parameter P and
PM = magnitude of the parameter P (e.g., speed).

These parameters were chosen as an initial step toward quantitatively representing the qualitative notion of realism of SAFOR behavior. The number of collisions with vehicles and buildings is clearly important since under normal circumstances a human driver would doubtfully purposely collide with anything.

Route efficiency measures the size the deviations from the designated route. A good example of the role of route efficiency in assessing the realism of driver behavior is that of a route through the large building. This route demands a deviation from the designated path because following the route would require at least one collision with the building which is unacceptable. However, a good deviation is one which minimizes the extra length in an executable path. A human driver would not drive to near contact with the building and then deviate around it but would rather begin to deviate as soon as he realized that the route extended through the building. An ideal route in this case would begin a deviation around the building at infinity. Thus, route efficiency measures the way in which a driver minimizes the impact of an impossible but avoidable situation.

Like collisions, vehicle speed is clearly a good measure of driver performance since the goal of a driver is to maintain the designated speed as closely as possible. Further, this parameter also measures how the performance of the software is degraded by the modifications.

Initially, the dispersions were computed simply to quantify the significance of any comparisons. However, analysis of the results indicated that route efficiency and mean speed dispersions also measured the randomness of vehicle behavior. A driver's behavior should appear purposeful. Thus, random wandering is, in general, an unacceptable manifestation which degrades realism so these dispersions should be minimized. This is equivalent to saying that given the simple circumstances which are portrayed in the exercises, a good automated driver would execute precisely the same route at precisely the same speed every time regardless of the effects of such random influences as the underlying operating system performance.

Although additional performance parameters may be needed to more fully assess the realism of driver behavior, these parameters form a reasonable starting set. These parameters were analyzed to assess the differences between the baseline and modified versions of the SAFOR program. The results of this comparison are discussed below.

Experimental Arrangement

Figure A2 illustrates the hardware configuration of the experimental arrangement which was used for the evaluation. The SAFOR commander's workstation enables a human to construct, monitor and control SAFOR exercises. This capability is implemented on a Symbolics LISP machine. The SAFOR simulation was implemented on a Sun 4 Sparcstation with Version 4.1 of the Unix operating system and the SunView environment. When debugging support was needed, then the Emacs gdb debugger was used. The SAFOR commander's workstation communicates with the simulation platform through an Ethernet with User Datagram Protocols (UDP). The terrain database which was used was of the Ft. Hunter-Liggett area.

While the commander's workstation provides a map view display of the exercises, it is updated relatively slowly. Thus, 3D visualization hardware from Technology Systems Inc. was added to this configuration to improve the SAFOR monitoring capability. The SAFOR simulation platform communicates with the 3D graphics engine through an Ethernet with SIMNET protocols. Future modifications will enable the simulation engine to interact with larger scale implementations through DIS protocols. This enhancement will also enable the interaction of the SAFOR with manned simulators.

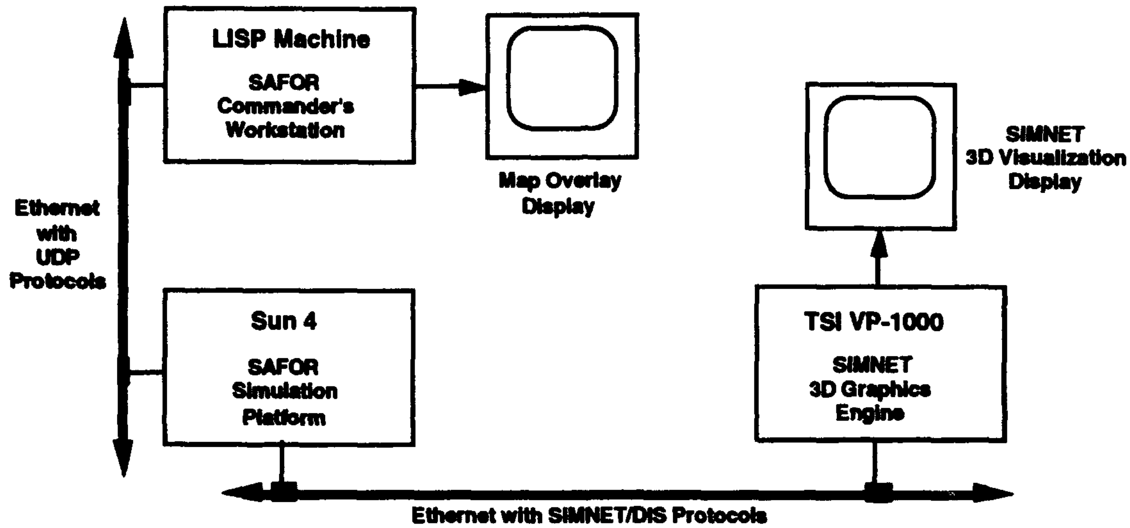


Figure A2. Demonstration SAFOR Hardware Configuration

The vagaries of the Unix operating system and the attachment of the Sun 4 to a laboratory wide network resulted in slight variations in the values of the raw data which were collected in the same exercise. As a result, each exercise was repeated several times to characterize the magnitude of these variations. This enabled the collection of additional performance statistics.

PERFORMANCE EVALUATION RESULTS

The exercise data are summarized in Table A3 which combines the results from all of the vehicles from all of the exercises together. This table also compares the performance of two different versions of the modified SAFOR. The first version implemented simple driver modifications. The second version included modifications to decrease the speed degradation, primarily through the implementation of a ships-passing rule when encountering oncoming vehicles. The number of collisions is summed over all of the exercises for all of the vehicles. The remaining parameters are mean averages for all of the vehicles in all of the exercises. The median values for the route efficiency and mean speed dispersions are also provided. These values together with the corresponding means crudely characterize the shape of the statistical distributions of their parent parameters. These values are provided for both the baseline and modified versions of the SAFOR program. Additionally, the percentage improvement is given. Where this value is negative, the modified software showed a degradation.

Table A3. Summary of SAFOR Evaluation Results

PARAMETER	BASELINE PERFORMANCE	MODIFIED PERFORMANCE (v0.1 / v1.0)	PERCENTAGE IMPROVEMENT (v0.1 / v1.0)
TOTAL VEHICLE COLLISIONS	935	0 / 12	100 % / 99 %
BUILDING COLLISIONS	211636	0 / 0	100 % / 100 %
MEAN ROUTE EFFICIENCY	77 %	82 % / 82 %	5 % / 5%
MEAN PERCENTAGE OF ROUTE EFFICIENCY DISPERSION	7 %	0.7 % / 0.4 %	90 % / 93 %
MEDIAN PERCENTAGE OF ROUTE EFFICIENCY DISPERSION	49 %	3 % / 2 %	94 % / 96 %
MEAN VEHICLE SPEED	6.3 m/s	4.7 m/s / 6.2 m/s	-25 % / -2 %
MEAN PERCENTAGE OF VEHICLE SPEED DISPERSION	14 %	4 % / 2 %	71 % / 86 %
MEDIAN PERCENTAGE OF VEHICLE SPEED DISPERSION	79 %	12 % / 14 %	85 % / 81 %

Figures A3 through A10 illustrate the performance of the baseline and modified version 1.0 (i.e., the second modified version) SAFORs for the exercises which used column formation. This modified SAFOR includes the ships-passing rule to accommodate passing oncoming vehicles. These figures depict the route efficiency, route efficiency changes, route efficiency dispersion, route efficiency dispersion changes, mean vehicle speed, mean vehicle speed changes, mean vehicle speed dispersion and mean vehicle speed dispersions changes. The results for the exercises in which the vehicles used the line formation were omitted for the sake of brevity although they show similar effects. The column formation exercises where chosen as examples because they were more complete representations of all of the exercises. Only Exercise 20 was not executed with the vehicles in column formation. Additionally, the results of the comparison between the baseline and the first modified SAFOR were omitted.

Those exercises for which multiple trials were infeasible for the baseline version are indicated. As a result, no statistics could be collected on the variation of the results and performance changes between baseline and modified SAFOR could not be meaningfully computed.

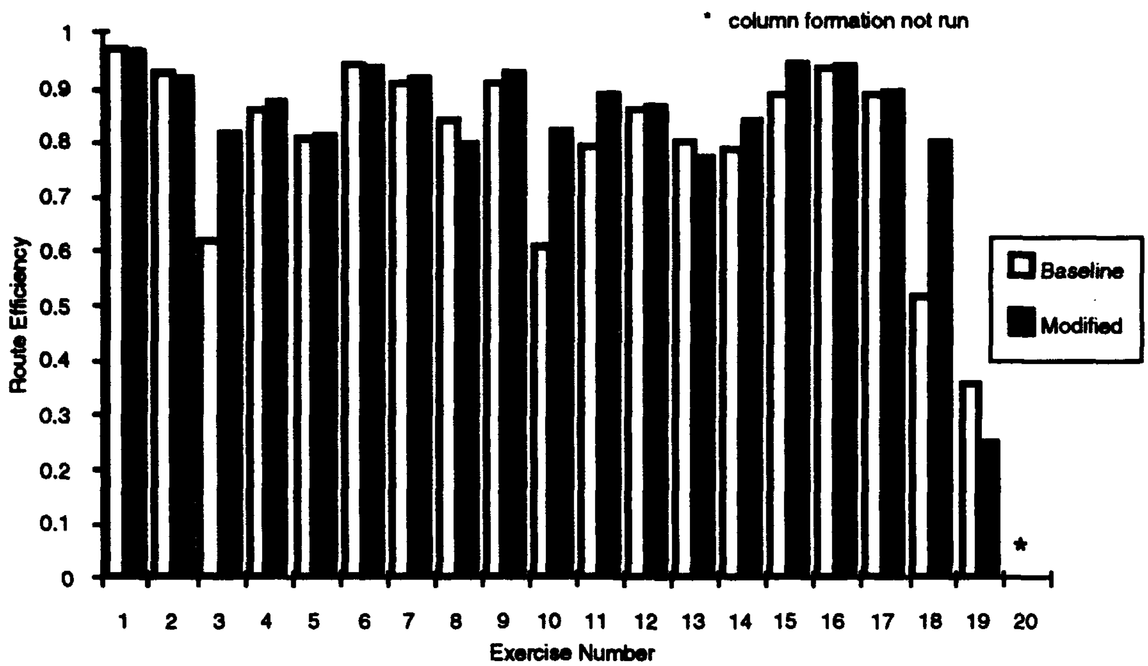


Figure A3. Route Efficiency Comparison for the Baseline and Modified Version 1.0 SAFORs in Column Formation

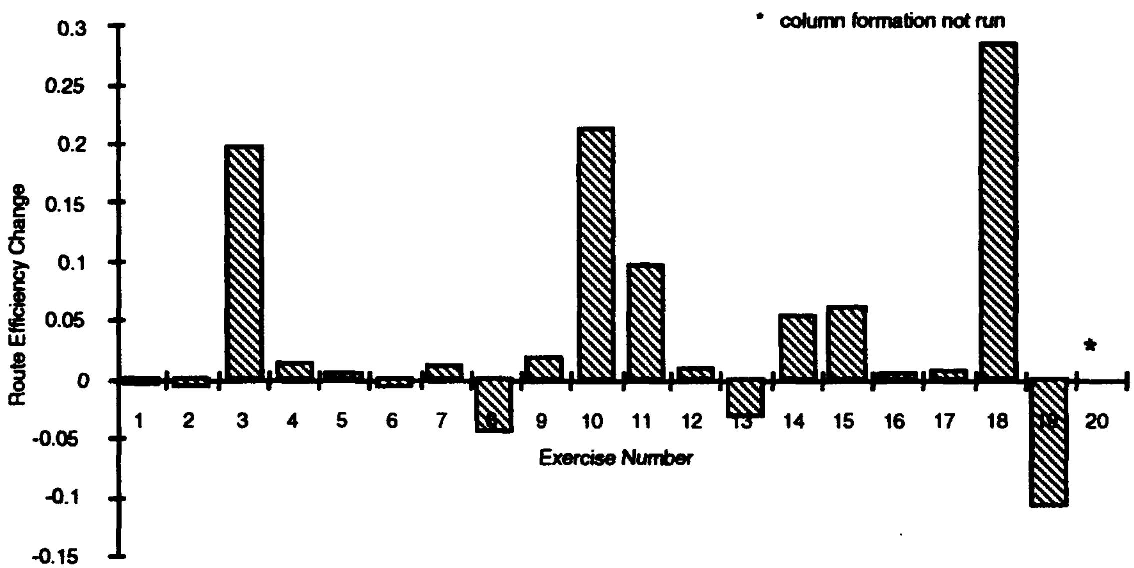


Figure A4. Route Efficiency Change from the Baseline to the Modified Version 1.0 SAFORs in Column Formation

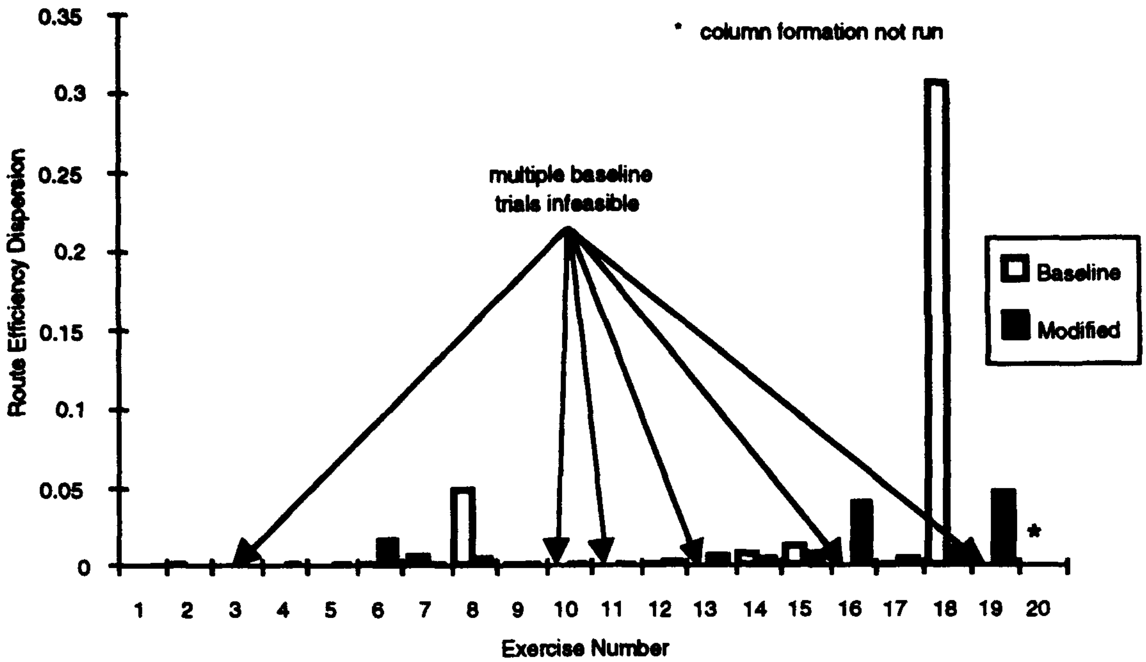


Figure A5. Route Efficiency Dispersion Comparison for the Baseline and Modified Version 1.0 SAFORs in Column Formation

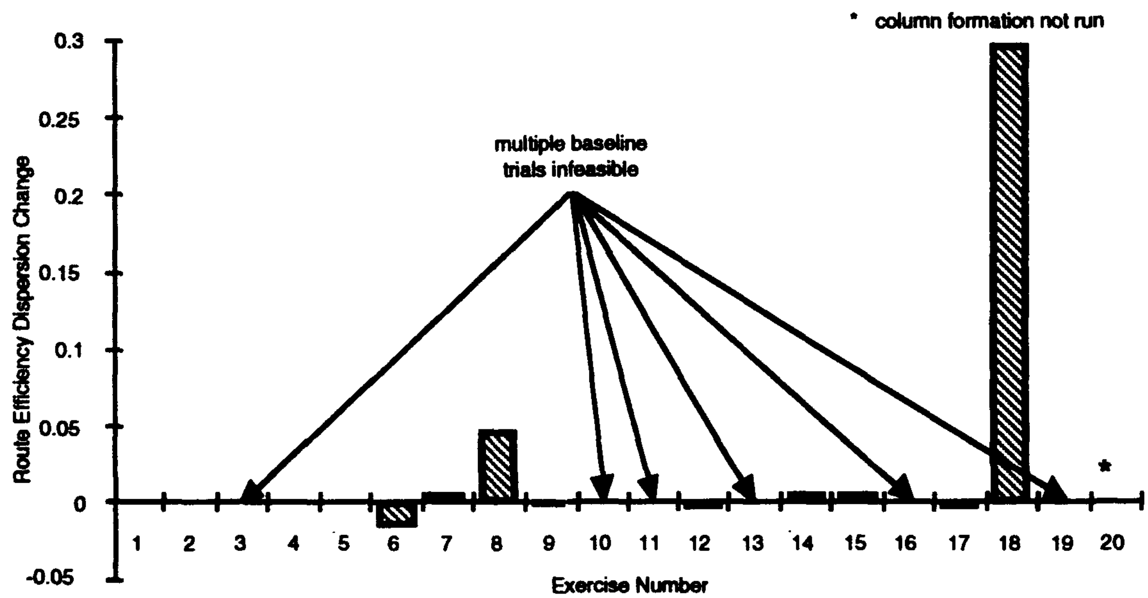


Figure A6. Route Efficiency Dispersion Change from the Baseline to the Modified Version 1.0 SAFORs in Column Formation

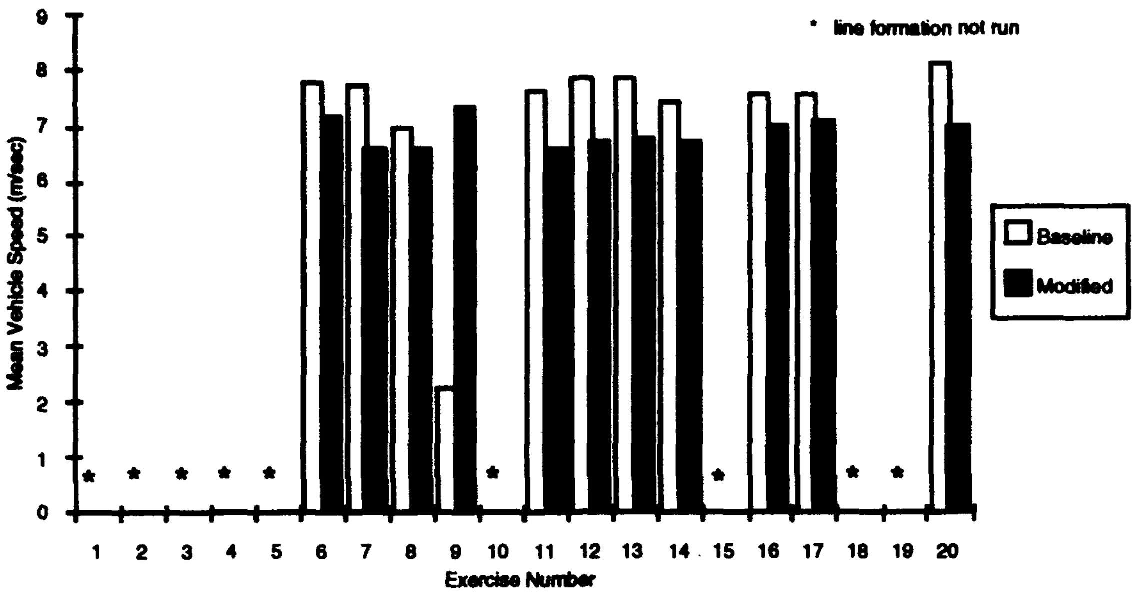


Figure A7. Mean Vehicle Speed Comparison for the Baseline and Modified Version 1.0 SAFORs in Line Formation

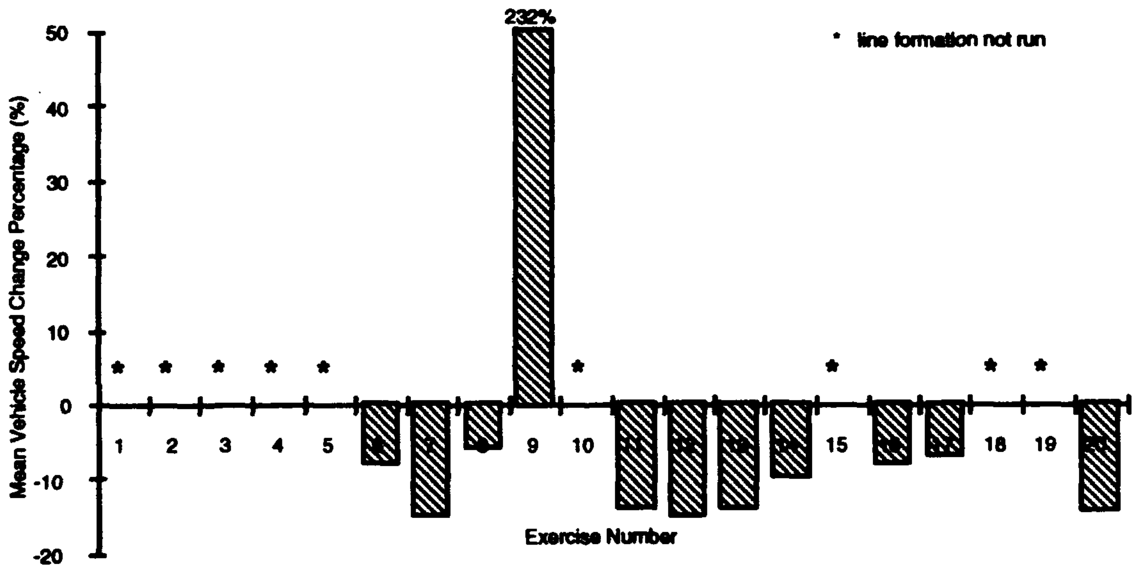


Figure A8. Mean Vehicle Speed Change from the Baseline to the Modified Version 1.0 SAFORs in Line Formation

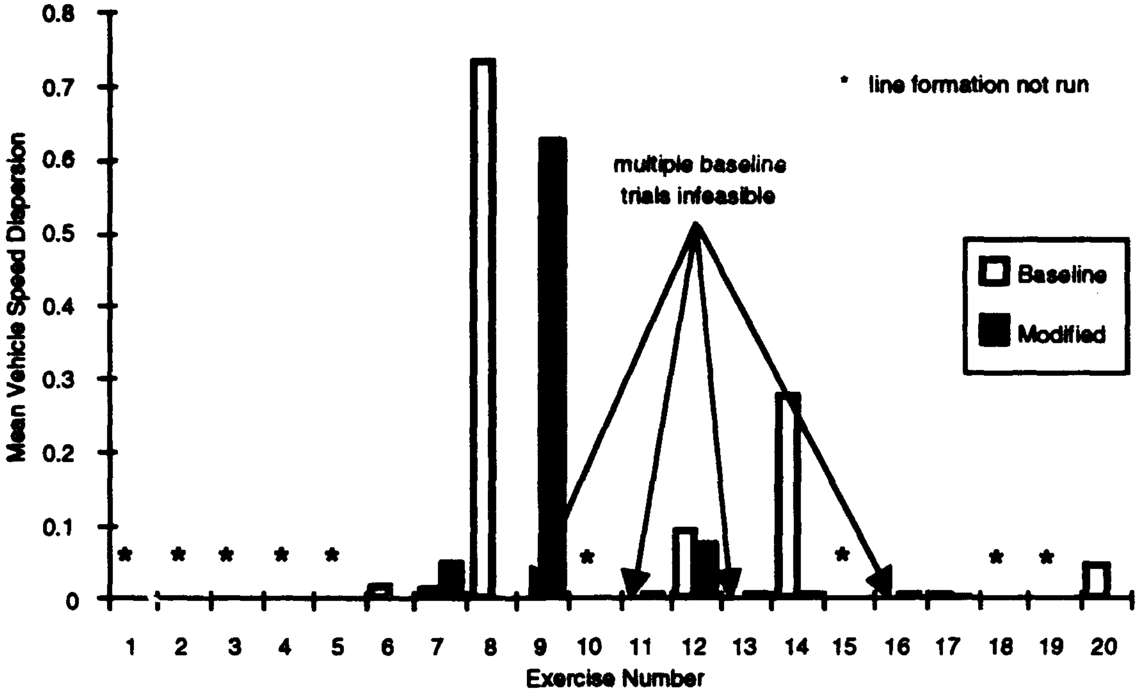


Figure A9. Mean Vehicle Speed Dispersion Comparison for the Baseline and Modified Version 1.0 SAFORs in Line Formation

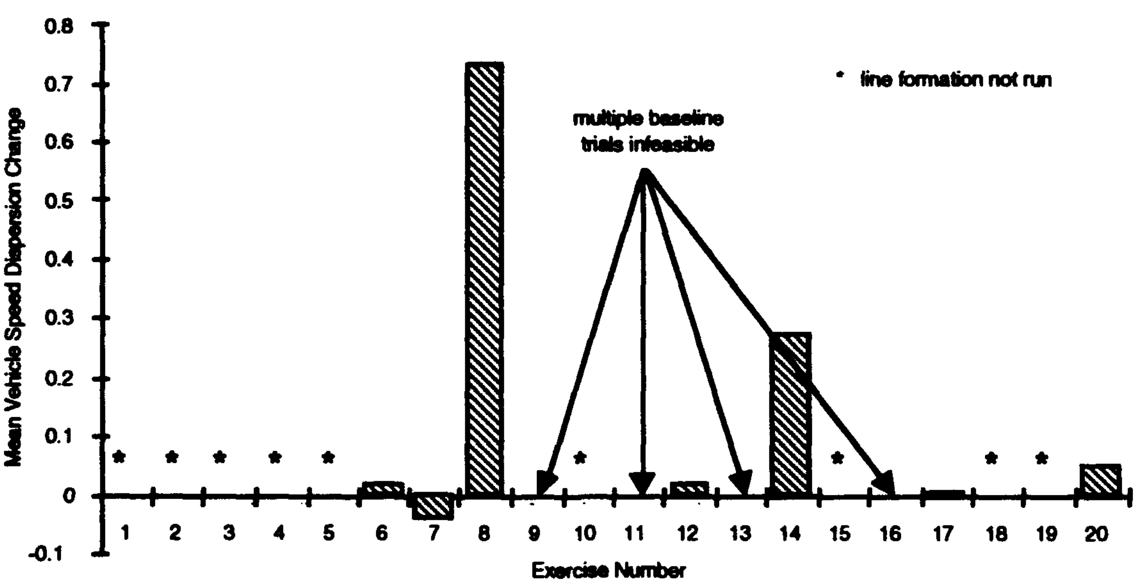


Figure A10. Mean Vehicle Speed Dispersion Change from the Baseline to the Modified Version 1.0 SAFORs in Line Formation

Four fundamental differences in the performance between the two SAFOR versions are clear. These are discussed below.

Building and Vehicle Collisions: First, the goal of eliminating all collisions was achieved. In the first version of the modified SAFOR, neither building collisions nor vehicle collisions occurred throughout all 200 executions of the exercises. The bulk of the baseline's building collisions were concentrated in exercises with routes involving the large building. Most of the baseline's vehicle collisions were concentrated in exercises which required changes of formation and formation changes of direction. Examination of these exercises with the 3D visualization system revealed that the vehicles appeared to move randomly. On the other hand, the modified SAFOR showed relatively purposeful motion in all of these exercises. Those exercises which could not be executed repeatedly with the baseline SAFOR involved situations where one or more vehicles became locked with either a building or another vehicle and could not successfully disengage. These exercises were terminated before completion by a timeout condition and were not re-executed.

When the software was again modified to increase the vehicle speed the number of vehicle collisions for the modified SAFOR increased a very small amount. These collisions resulted during close quarters maneuvering primarily because the Sun 4 did not have enough computational throughput to support the simulation and the vehicles traveled beyond their preview distance during the tick. Thus, they collided with vehicles which had not been observed in the previous simulation tick. It is anticipated that a more powerful computer would again reduce the number of vehicle collisions to zero for the modified SAFOR while maintaining the speed improvement which has been attained.

Route Efficiency: Second, the mean route efficiency was improved slightly (but still statistically significant). The largest improvements in route efficiency were in exercises which involved interactions with the large building. In addition, a significant improvement was obtained in the exercise which required a company of vehicles to change direction. The only exercise in which the modified SAFOR suffered a route efficiency degradation was one where a company needed to make a large path deviation to assume the designated line. The demand for coordinated movement within the company makes efficient routes impossible for most of the vehicles. In addition, this exercise was terminated for the baseline version before it accomplished the designated mission so direct comparison is misleading.

The modified version 1.0 (i.e., the second version) showed very similar route efficiency performance as the earlier version. Thus, it is fair to assume that the modifications did not degrade this parameter at all.

Mean Vehicle Speed: Finally, the performance improvements of the first version of the modified SAFOR were attained at the cost of decreased speed. This SAFOR tended toward slower speeds in almost every exercise with the exception of those in which the baseline vehicles became locked with the large building (i.e., Exercises 3 and 10). The average speed degradation was approximately 25% although in some exercises this was over 50%. The modified version showed the most degraded speed in situations where the vehicles were approaching one another. This condition was confirmed through observations of the exercises with the 3D visualization system. These observations showed the vehicles approach one another ever more slowly, stop and hunt for a path around each other. This is caused by a conservative obstacle avoider which always tries to reduce vehicle speed proportionally to closing speed or time to potential collision. At first thought, this seems a reasonable design but it clearly has aspects which manifest unacceptable behavior.

This problem was corrected by implementing a ships-passing avoidance rule where closing vehicles always turn in the same relative direction away from one another to avoid a collision. This solution also remedied those situations where mean speed dispersion was degraded. The performance of this version of the modified SAFOR validated the correctness of this modification and emphasized the utility of the systematic approach to SAFOR performance evaluation which was pursued.

Measures of Randomness: Third, both the route efficiency and mean speed dispersions were considerably reduced in both versions of the modified SAFOR. However, comparison between the baseline and modified software was hampered by the inability to collect statistics on six of the nineteen exercises which used the column formation (three of ten exercises which used the line formation could not be completed as well). Nevertheless, a large improvement in route efficiency dispersion is obvious in Exercise 18 and large improvements in mean speed dispersion are apparent in Exercises 8 and 18. In general, route efficiency dispersion was decreased where statistically significant. However, mean speed dispersion was actually increased in Exercises 12 and 14. This resulted from random movements when approaching vehicles attempted to pass one another. The overall decreased route efficiency and mean speed dispersions imply that the randomness of the vehicle motion was reduced. This implication was later confirmed through observations of the exercises with the 3
D visualization system.

Operator Intervention: One additional result which is not obvious from either Table A3 or Figures A3 through A10 is that apparently all situations where the SAFOR vehicles become deadlocked have been eliminated. This is another important improvement for which no performance measure was formally defined. Perhaps, this suggests the need for a measure of the number of times SAFOR operator intervention is required to correct some unacceptable behavior.

Appendix B: CASE-BASED TACTICS ACQUISITION

One of the major problems in building computer generated intelligent forces concerns knowledge acquisition. The challenge is to instill the intelligent forces with sufficient expert knowledge that they may be able to immitate the behavior of a true expert. Ideally, this knowledge could be captured simply by observing the actions of an expert engaged in realistic scenarios. Without sufficient background knowledge, however, there is no way to know how an expert might have altered his behavior had the details of the scenarios been even slightly different. This leads to the more conventional knowledge engineering alternative. Knowledge engineering involves a computer programmer attempting to understand the domain by talking with and observing an expert and then encoding that knowledge into a computer program. While effective, capturing the knowledge in this manner is a difficult and time consuming task. As a compromise between these two extremes, we have developed a methodology that permits an expert to input his knowledge directly through sets of example cases. The expert is able to interactively explore how scenario variations can influence the behavior of the intelligent forces, and is thereby able to iteratively taylor the intelligent forces to behave in a manner that reflects his own choices.

Our case-based tactics acquisition approach exploits the natural tendency for people to express their knowledge in terms of specific problems and examples. Looking at and reacting to a concrete situation, an expert can provide rules of thumb or suggest specific actions to be performed. Our system generates concrete situations for the expert through the use of intelligent force simulations. At any time during a simulation, the expert may suggest different actions and the intelligent forces will respond accordingly. The expert can then observe the effects of his suggestions and revise them as he sees fit. As this process goes on, the expert is actually generating a database of cases that can be used by the intelligent forces in future simulation exercies to emulate the expert's tactical responses.

In some situations, the expert may prefer to describe a specific tactical configuration explicitly rather than have to wait for it to show up in simulation. For this, our tool is designed to support a form of electronic chalk board. In this mode, the expert can graphically indicate the positions and the initial states of friendly and enemy forces. The expert can then identify behaviors for each of the intelligent force entities in the scenario. This is similar to the way tactics experts currently use an ordinary chalk board to describe a tactical scenario. Rather than moving through the scenario sequentially, the expert provides snapshots of the scenario at points in time where critical decisions must be made. The freedom to describe the scenario in a non-sequential manner allows an expert to deal with the most salient features first, before delving into the details.

In the remainder of this appendix, we first provide an overview of the knowledge acquisition process and then we describe, in detail, the underlying mechanisms that make this form of knowledge acquisition possible. Our overview of the process is presented from the perspective of a tactics expert attempting to describe a complex tactic through a series of examples or cases. The section on the mechanisms for knowledge acquisition explains how stored cases such as these may be used in future simulations to control an intelligent force entity so that it emulates the behavior of the expert.

The Automated Knowledge Acquisition Process

To illustrate the process of tactics acquisition made possible by our case-based acquisition tool, we present an example of how an expert might go about entering the knowledge for a complex tactical scenario for the air domain. Specifically, we are interested in a 2v4 beyond-visual-range scenario in which two Blue planes are attacking an established defensive position held by four Red planes. This is illustrated in Figure B1. Our goal is to capture tactical knowledge for the Red planes so that the Red planes can be used as intelligent automated opponents in future training exercises with the Blue planes controlled from manned simulators. During the tactics acquisiton process, the tactics expert will have control of both the Red and the Blue planes. This will allow the expert to explore the entire range of alternative actions available to each side.

Knowledge acquisition begins by establishing the initial conditions for the simulation. The tactics expert creates two Red formations, with two planes in each formation, and one Blue formation. Initially, the simulation is run with the Red Lead Formation and Support Formation flying Combat Air Patrols (CAPs). As specified by the tactics expert, the Lead Plane of the Lead Formation is to perform a CAP maneuver at 30,000 ft., and the Wingman is to fly in formation with the Lead. The Lead Plane of the Support Formation is to fly a CAP maneuver at 20,000 ft., and the Wingman is to fly in formation with the Lead. The Blue formation is set to approach Red from the East, at an altitude of 25,000 feet.

With these initial conditions established, the expert re-enables the simulation and watches the planes move. The Red planes fly in the standard race track pattern of the CAP as the Blue Bogeys approach. Eventually, the expert is notified that the Blue Bogeys are within detection range of the Red planes. Still, the Red planes continue to fly in their CAP formations until the expert decides they should change. At the appropriate time, the expert decides that the Red Lead Formation should break out of the CAP and pursue the Bogeys at the altitude of the Bogeys. To cause this change, the expert halts the simulation and specifies the new behavior for the Red planes. At this time, a case is entered into the Case Database for the Lead Plane. When the simulation is resumed, the Lead Formation now leaves the CAP and begins to head toward the Bogeys.

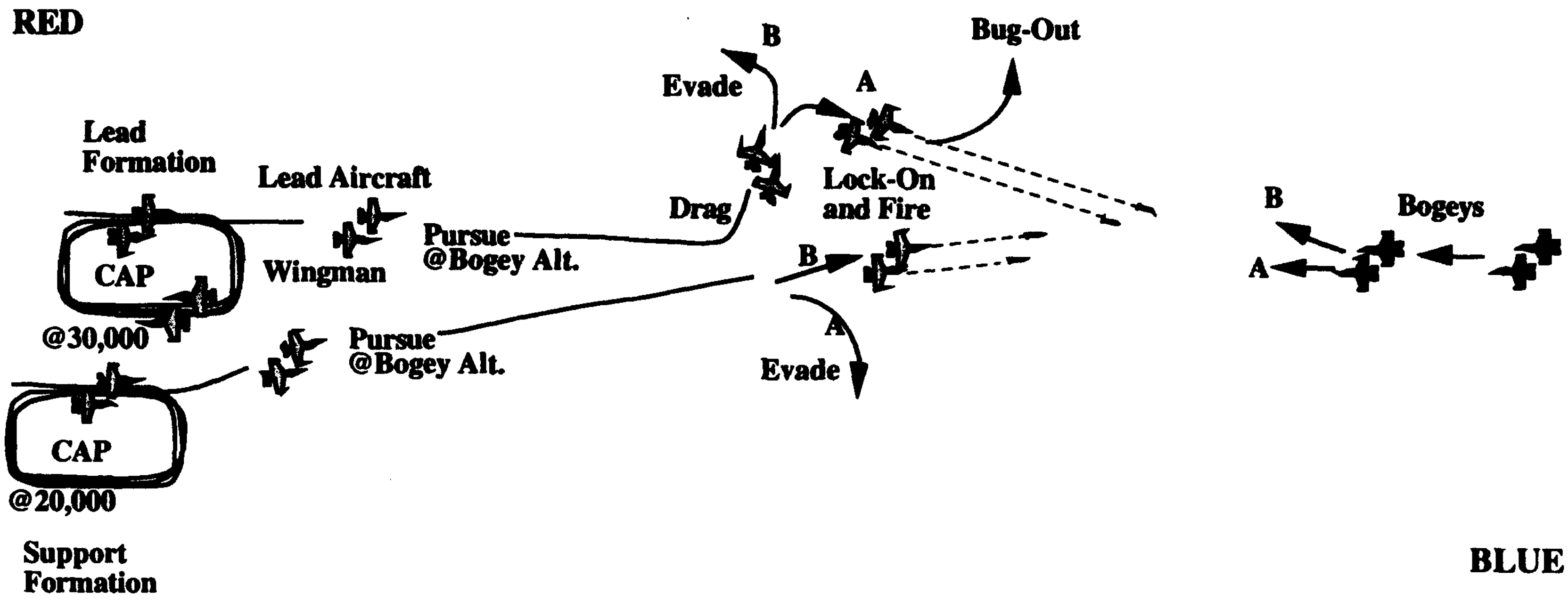


Figure B1. An example 2v4 beyond-visual-range tactical scenario in which two Blue planes are attacking four Red planes in a defensive position. The expert's objective is to provide the knowledge needed by the Red planes to execute the alternative tactics presented in this scenario.

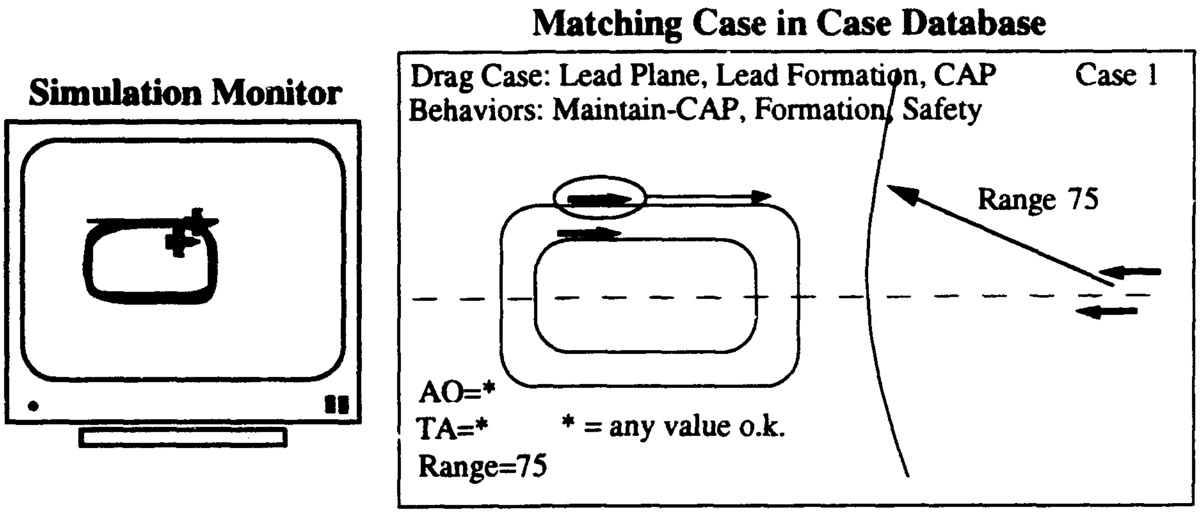


Figure B2. The tactics expert initially wants the Lead Formation plane to fly in a CAP flight pattern. On the right is shown a pictoral represenation of the case that is input to the database to control the CAP. On the left is the Simulation Monitor that the tactics expert views to confirm that the tactic is proceeding as planned.

As shown on the left side of Figure B2, the tactics expert observes the flight trajectories of all planes on the Simulation Monitor. To the right, the Figure shows a pictoral representation of the initial case that keeps the planes in their CAP formation. While the expert does not have access to this pictoral representation, he does specify the case through his interaction with the pictoral view seen on the Simulation Monitor. The primary elements of any case are the geometric attributes that identify when the case is appropriate and the list of behaviors that the plane matching this case should perform. The case shown in Figure B2 indicates that the "Lead Plane" of the "Lead Formation" should be "Maintaining a CAP," flying in "Formation" with the Wingman, and observing "Safety" constraints while flying in the formation. This behavior is specified by the tactics expert when the initial conditions for the simulation are established. The tactics expert then notes that some of the attributes describing the geometry are not important for the situation. For instance, the "Angle Off" (AO) and the "Target Aspect" (TA) can have any value. Within the case database, the unspecified parameters are expressed with an asterisk (*) to show that these variables can attain any value without changing the applicability of this case.

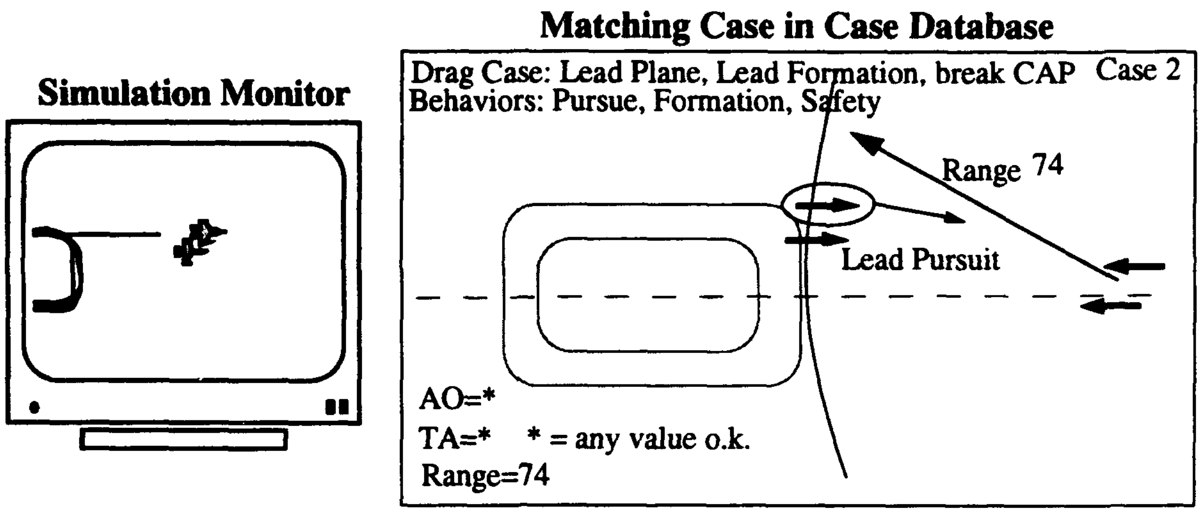


Figure B3. After observing in the Simulation Monitor that two Bogeys are approaching, the tactics expert decides that the Red Planes should break the CAP and pursue the Bogeys. The new case in the database that corresponds to this action is shown on the right. The tactics expert verifies the actions while observing the Simulation Monitor on the left.

When it can be seen that the Bogeys have been identified and are at the appropriate range to initiate a pursuit, the tactics expert halts the simulation and adds a new case to the database. The new case (illustrated in Figure B3) directs the Lead Plane of the Lead Formation to break the CAP maneuver and pursue the Bogeys. The tactics expert specifies that the Wingman is to fly in formation. Although not shown in the figure, all the parameters associated with the plane's actions (pursuit type, desired altitude, formation parameters, etc.) are determined by the tactics expert and input to the case.

With the addition of this new case to the case database, the tactics expert either resumes the simulation from the current state or restarts the simulation from the initial state. By restarting the simulation, the expert can allow the cases to direct the vehicle actions. This permits the transition between old cases and newly added cases to be examined. Since adding a new case to the case database might affect the past history of the simulation, restarting the simulation is an important part of the overall knowledge acquisition process. In this knowledge acquisition session, however, the tactics expert elects to continue the simulation from the current state. The newest case added to the case database is the case that directs the Lead Plane, and the tactics expert wants to continue guiding this plane through the simulation until the next decision point is encountered.

After allowing the Lead Formation planes to travel an appropriate distance, the expert halts the simulation again and enters a similar case to the one shown in Figure B3 for the Support Formation planes. This case directs the Support Formation planes to stop their CAP maneuver and to begin to follow the Lead Formation planes.

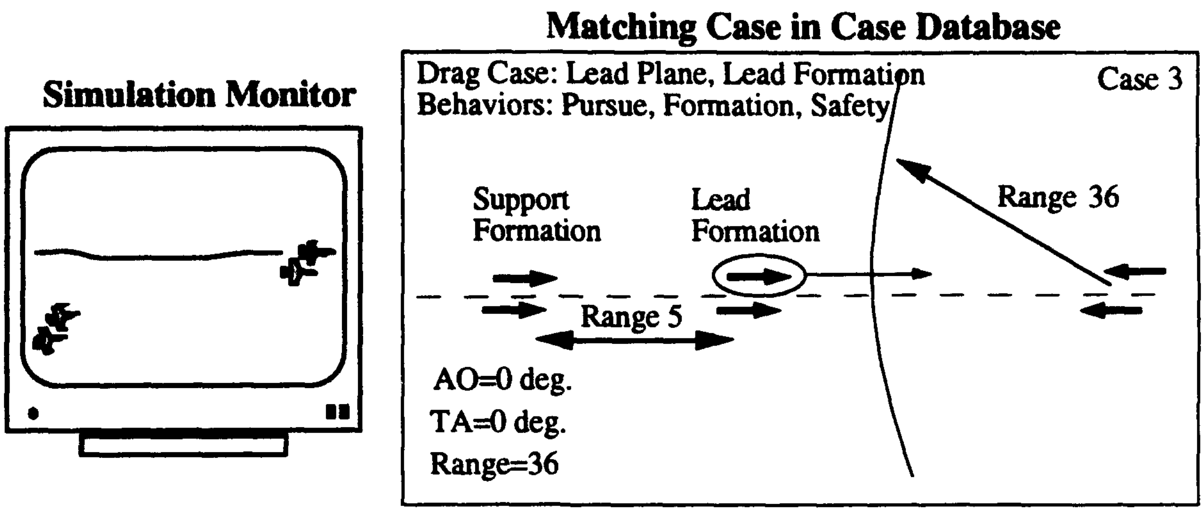


Figure B4. The tactics expert observes in the Simulation Monitor that the Lead Formation is getting in place for the offensive portion of the tactic. At this point, the simulation is stopped so that the coordination of the Support Formation position can be specified. The case that the tactics expert stipulates for this configuration is shown on the right. When this case controls the simulation, the tactics expert observes the coordinated behavior on the Simulation Monitor as shown on the left.

Next, the tactics expert observes the Simulation Monitor, and the decision point for the next transition in cases is identified. The tactics expert halts the simulation so that the coordination of the Support Formation with the Lead Formation can be specified. Figure B4 illustrates the next case added to the case database. In this situation, the Support Formation must be at a particular location relative to the Lead Formation. This coordinates the location of the Support Formation for the tactic.

The cases in the case databases of the Red forces are generalizations or "Textbook Cases" for guiding decisions in the simulation. As seen in the figure, the vehicular movement that occurs in the simulation need not correspond exactly to the case in the database. Therefore, the precise details of a case are not particularly important. This leads to two alternatives for entering cases. The tactics expert can either enter a case simply by recording a configuration he encounters during the simulation, or the expert may construct a case from a "textbook" example.

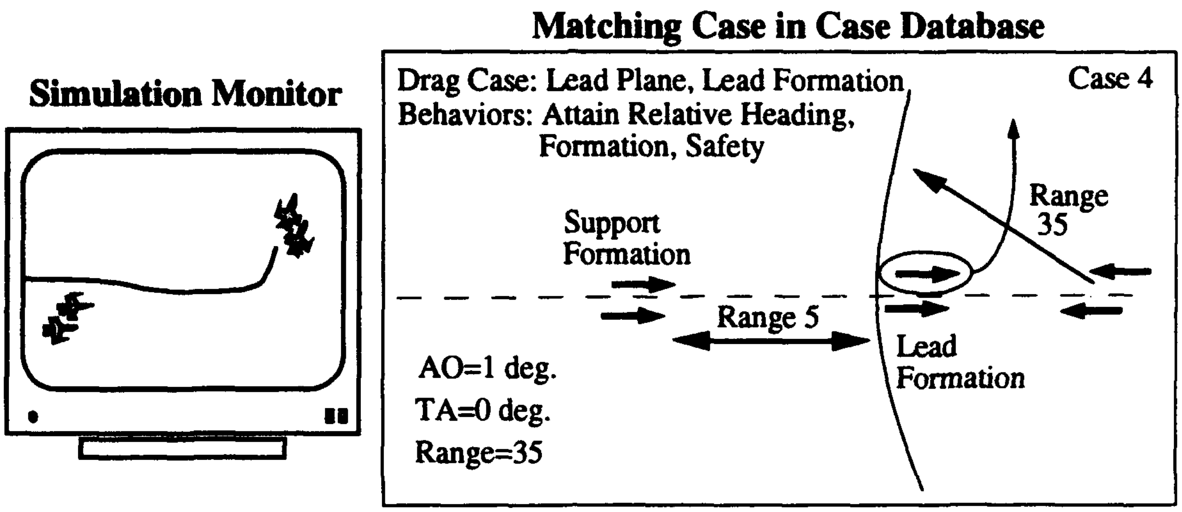


Figure B5. When the tactics expert decides that a drag maneuver should be initiated, the simulation is stopped, the case is added as shown on the right, and the simulation is continued. The tactics expert then observes the drag maneuver shown on the Simulation Monitor and continues the simulation until the next decision point.

The next case is added when the tactics expert observes from the Simulation Monitor that the range is appropriate for a drag maneuver. Figure 6 illustrates this maneuver. In this maneuver, the Lead Plane flies a trajectory with a heading perpendicular to that of the Bogeys. Note that the Lead Plane's heading is specified relative to the heading of the Bogeys. In this way, if the Bogeys are to change their heading, the maneuver will remain valid. Within case matching, both relative and absolute headings may be used by the tactics expert to specify the actions of the airplanes.

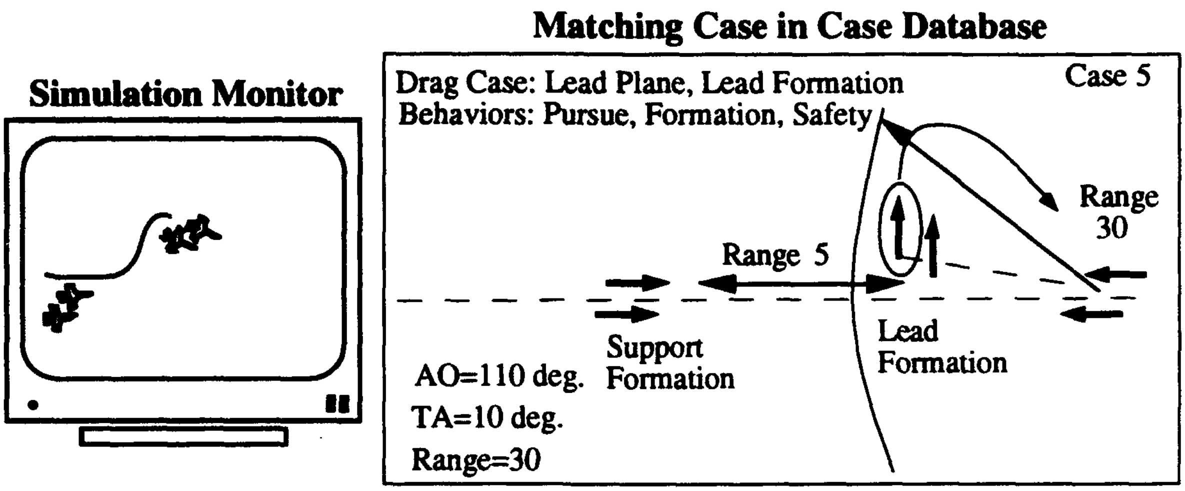


Figure B6. The tactics expert decides that the lateral separation is sufficient to continue the next defensive action. As shown on the right, the case that the expert specifies requires that the Support Formation be in place for a contingency action while the Lead Plane transitions into a "Pursue" behavior. The tactics expert observes the case being played out on the Simulation Monitor.

When the Lead Formation finishes the Drag maneuver, the tactics expert inputs two cases. In the first case (Figure B6), the Bogeys do not react to the Drag, and the Lead Formation carries on with the offensive action. However, as a contingency plan, the tactics expert also adds the case (Figure B7) where the Bogeys pursue the Lead Formation after the Drag maneuver. Here, the Lead Formation should evade the possible pursuit; as a contingency, the Support Formation is in place for a defensive strike against the Bogeys. On the Simulation Monitor, the Bogeys do not pursue the Lead Formation, thus, case 5 matches and controls the Lead Plane, and case 6 resides in the database, but does not currently get matched.

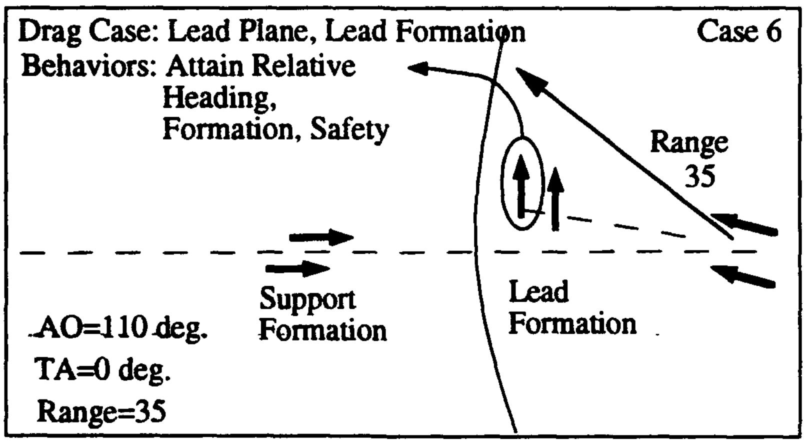


Figure B7. A case is added to the case database to provide an alternative response should the Bogeys choose to pursue the Lead Formation.

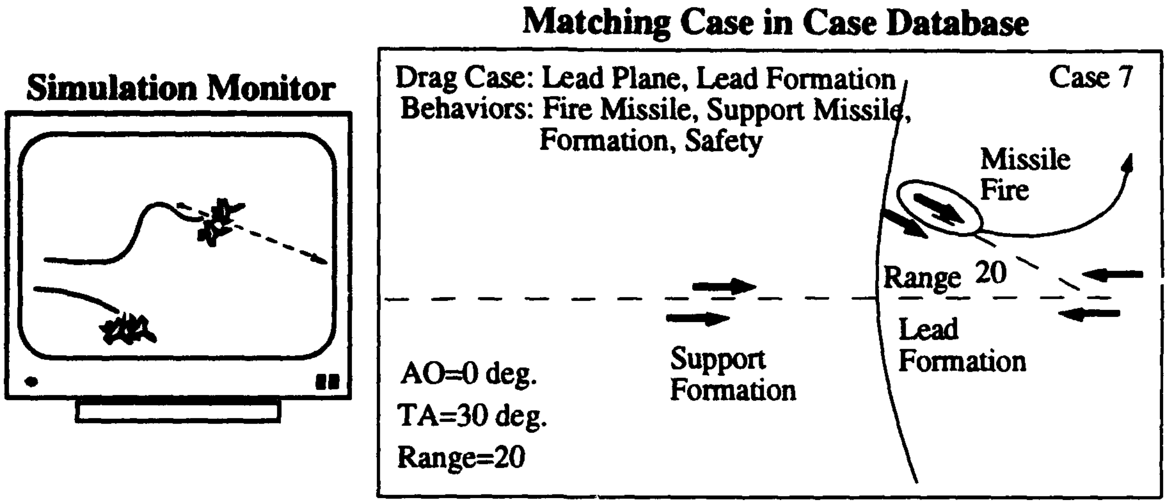


Figure B8. The tactics expert observes in the Simulation Monitor the configuration is appropriate to fire a missile. The case is input in the database and controls the appropriate action.

When the tactics expert determines from observing the Simulation Monitor that the geometry is appropriate to fire a missile, the simulation is stopped, and a missile firing case is added to the database. Figure B8 illustrates the case added to the database. This is the final decision point for the tactic that the expert outlined in the initial chalkboard scenario description.

Finally, the tactics expert reviews the tactical knowledge acquired in the case database. After all these cases have been added to the database, the tactics expert runs the simulation from several initial conditions to observe the Red force's actions under the control of the database. At this time the transitions between cases can be observed, modifications to the case database can be made, new cases can be added, the geometry may be altered, etc. Indeed, the cases that the tactics expert has already input to the system should be correct for the configuration that the tactics expert observed. There should be little need to alter a case once it is specified. Instead, if the simulation does not proceed as the tactics expert intends, the tactics expert should stop the simulation and specify the correct action at that decision point. For instance, when the simulation initial conditions are varied, the tactics expert may add new cases to correct any situations where inappropriate behaviors are evident. The repeated use of simulation and the updating of new cases should incrementally build the knowledge base of the Red Cased-Based Fighters to a competency level that the tactics expert observes to be acceptable.

In Figure B9, the cases acquired in this knowledge acquisition session are illustrated. All the cases shown apply only to the Lead Plane of the Lead Formation. These cases are separate from the cases applicable to the Support Plane of the Lead Formation, and the two Supporting Formation planes. As shown in Figure B10, a case database is built up for each plane involved in a multi-agent scenario. In this multi-agent scenario, the location of the Support Formation relative to the Lead Formation is important for the appropriate contingency actions to occur. The cases that require these formations to be in positions relative to each other should be coordinated in both the case database for the Lead Formation and the case database for the Support Formation.

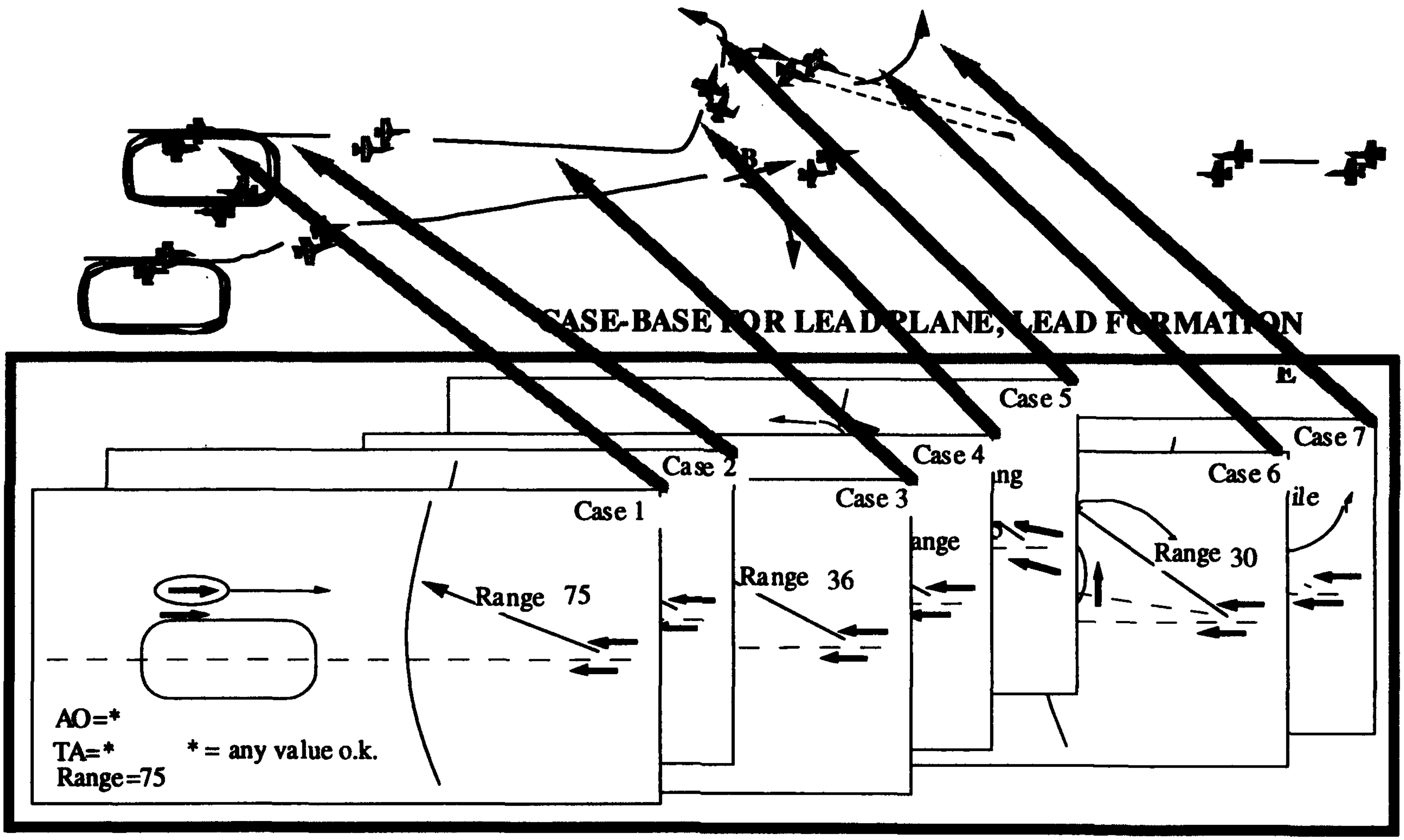


Figure B9. At any time during a simulation, a case in the case-base for the Lead Plane of the Lead Formation will control this plane through its role in the acquired tactic.

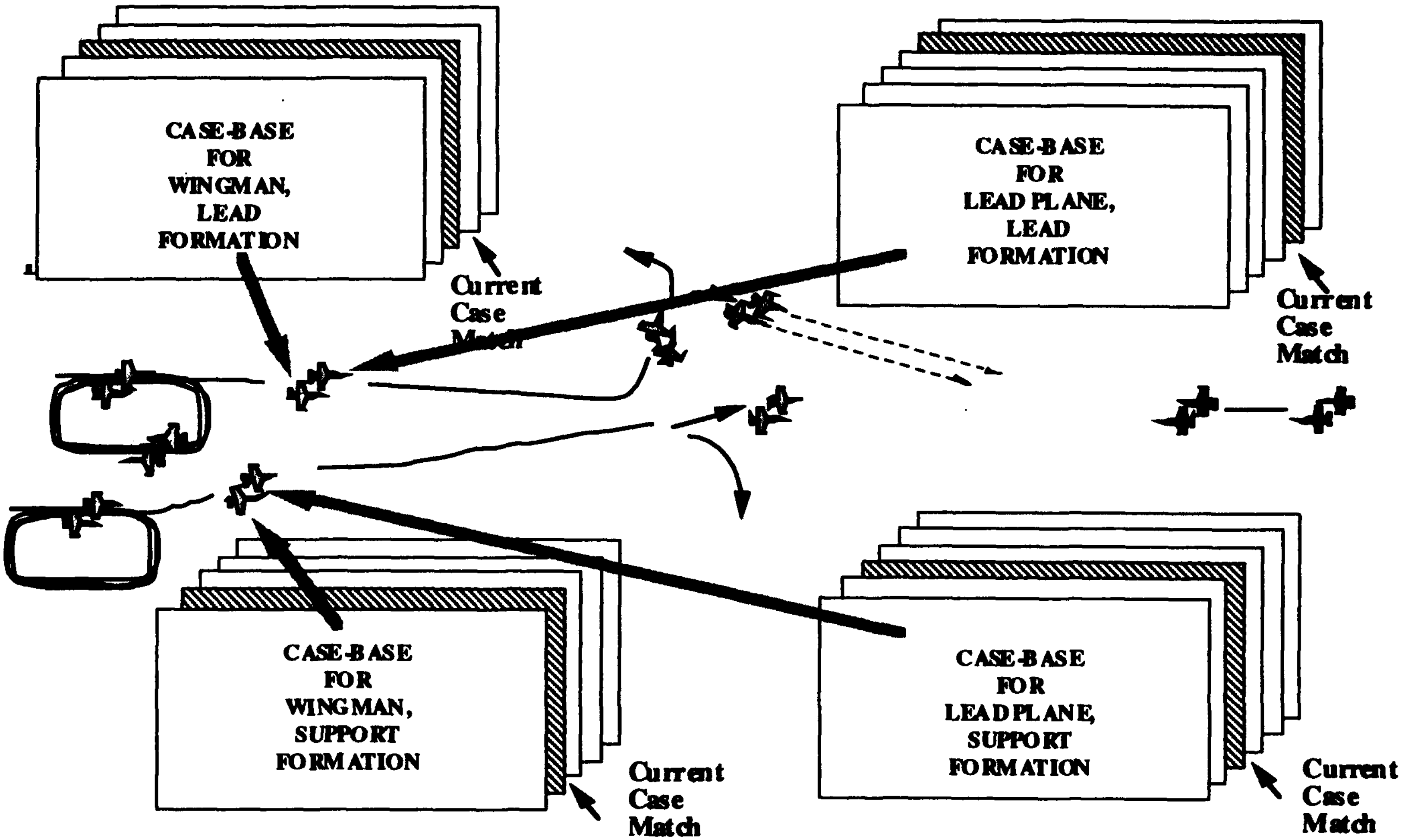


Figure B10. For multi-agent scenarios, such a this 2v4 scenario, a case base exists for each plane involved, so that at any time during a simulation a current case match controls each plane concurrently.

Appendix C: Experimental Analysis of Case-Based Tactics Acquisition

The desired goal for our case-based IFOR , in the context of the CAAT program, is that it serves as a useful knowledge acquisition tool to aid in the development and improvement of IFOR agents. To achieve this goal, it is necessary for the case-based IFOR to rapidly capture of a broad range of tactics. Additionally, the case-based agents must be suitably challenging opponents to the SOAR-based IFOR such that they can help to pinpoint deficiencies in the SOAR knowledge base.

In light of these goals, we have evaluated the case-based IFOR in terms of their rate of capturing new tactics and their effectiveness as opponents in simulation exercises. Typically, it is difficult to estimate the time needed to acquire a specific amount of knowledge. However, using case-based methodology, we are able to estimate the time required to create new cases and develop agents that exhibit distinct variations in their tactical behavior. The effectiveness of an agent as an opponent is also hard to quantify. In the role of an opponent agent, the most important aspect is not so much whether it wins or loses, but whether it is able to isolate a deficiency of knowledge in the SOAR agent. Therefore, if the case-based IFOR can produce engagements that lead to marked improvements to the SOAR knowledge base, then it is serving as an effective opponent and will have accomplished its goal.

Tactic Sets

During the performance of this program, the SOAR group's focus was on the development of a robust IFOR agent for 1v1 engagements. This agent is intended to handle a range of scenarios and conditions within the realm of 1v1 beyond-visual-range engagement. To support this development, we created a number of distinct tactical scenarios using our case-based IFOR. Each scenario is defined by a tactic set, which consists of a number of cases that together define a series of alternative tactics that are executed in response to the opponent's actions.

For this program, we developed three primary tactic sets for the case-based IFOR, with a number of variations on two of these sets. The three primary tactic sets are referred to as sets A, B, and C in Figure C1. In all of these scenarios, the Blue agent is controlled by the SOAR IFOR, and the Red agent is controlled by our case-based IFOR. At the time these scenarios were developed, there was some degree of uncertainty about whether radar warning receivers would be supported in the planned simulation. To cover either possibility, we developed some tactic sets that work with, and some that work without the radar warning capability. A brief description of these tactic sets follows.

In tactic set A, the Red and Blue agents are intended to fly toward each other until the Blue agent fires a missile. The Red agent detects this event with its simulated radar warning receiver, and responds by turning away from the Blue agent. The Red agent then turns beam to the Blue agent for a short time, and then turns back toward Blue to fire its own missile. The details of this scenario vary depending on how close Blue is to Red when Blue fires its missile. If Blue fires at a closer range, Red will turn beam to Blue, double back on itself, and turn beam on Blue again. This is intended to serve as an evasive maneuver to increase the likelihood that Blue will lose track of Red. Again, when Red gets to the appropriate orientation, it will turn back and fire at Blue.

In tactic set B, Red does not use a radar warning receiver to detect missile firing. Instead, Red waits until Blue is within a given range. At this range, Red turns beam to Blue and dives 5,000 meters. If Blue continues to follow red after this maneuver, Red will turn back and go home. On the other hand, if Blue appears to have lost track of Red, then Red will turn back and fire a missile at Blue.

In tactic set C, Red climbs 5,000 meters instead of diving as it did in tactic set B. Otherwise, the horizontal maneuvers are very similar to those in tactic set B. This alternative is intended to explore the tradeoff of gaining a tactical altitude advantage at the expense of a slower avoidance response.

Figure C1 also shows the variations that were created for tactic sets A and B. Some of these variations were created before observing the first version of the SOAR IFOR agent, and some were developed afterwards. Although many of these variations constitute qualitatively different tactical behavior, each was derived by modifying or supplementing the cases from the parent tactic set. It is important to note how quickly these variations may be generated, so an interesting and broad range of behavior can be explored in a relatively short amount of time.

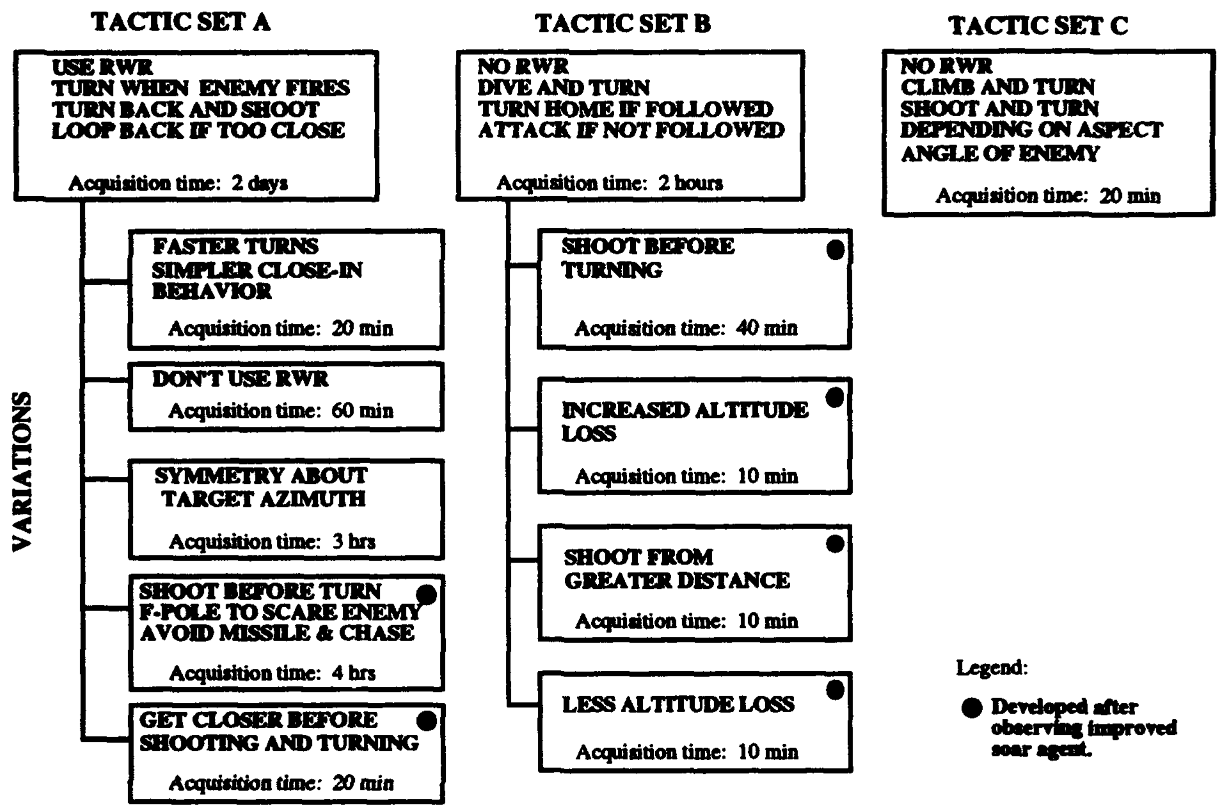


Figure C1. Three primary tactic sets were developed for 1v1 experiments to evaluate how well the case-based IFOR can be used to improve the SOAR IFOR. The initial experiments led to several improvements in the SOAR IFOR. Additional tactic sets were constructed to provide new challenges to the improved SOAR IFOR agent.

Our estimates of development time for each tactic set are shown in Figure C1. These figures show an average of approximately five hours development time per tactic set. This result is heavily influenced by the extensive time required to create the first tactic set. Nevertheless, under most circumstances, development time for capturing new tactical knowledge is fairly short using our case-based agent.

Experimental Results

The above tactic sets were used in an experiment to evaluate the value of case-based IFOR in enhancing the capabilities of the SOAR-based IFOR. The experiment was conducted in two sessions. In the first, our case-based IFOR was set against a SOAR IFOR agent at a time when neither development group had seen the capabilities of the other group's agent. Problems detected in the SOAR agent were then presented to the SOAR developers so that the SOAR agent could be improved. A second session was then held, using the original case-based IFOR as opponents to the improved SOAR agent. This allowed us to benchmark the progress in the development of the SOAR agent.

As the results in Table C1 indicate, the case-based IFOR (Red) uncovered many weaknesses in the SOAR agent's (Blue) behavior during the first session. Most of these weaknesses were due to missing knowledge in the SOAR agent at the time of the experiment. Some highlights of the missing knowledge are:

- The SOAR agent did not identify an opponent as hostile if that opponent always headed directly toward the SOAR agent. This is because the SOAR agent expected to see an F-pole maneuver as an indication of a missile firing. Without seeing that maneuver, the SOAR agent flew straight toward the opponent until getting shot down.

- The SOAR agent failed to support its missile my maintaining radar contact after it had the advantage of having shot first. In several scenarios, the SOAR agent would shoot before its opponent had shot. If the SOAR agent were to have maintained support for its missile, it could have killed the opponent and rendered the incoming missile harmless. Instead, the SOAR agent turned away, and disengaged radar contact as soon as it got an indication that its opponent had fired a missile. While this was intended to be a low-risk tactic, it occasionally led to the SOAR agent's destruction.

- The SOAR agent could not maintain radar contact while pursuing a collision course intercept. At certain times, the opponent would turn beam to the SOAR agent. Based on the opponent's new direction and speed, the SOAR agent would determine a new collision course to intercept its opponent. Because the opponent was moving perpendicular to the SOAR agent's path, the new collision course required that the SOAR agent turn away from the immediate opponent's position. This, in turn, caused the SOAR agent to lose sight of its opponent because it kept the radar pointing straight ahead even though the SOAR agent's plane was turning. The agent clearly needed additional knowledge in order to keep the radar pointing at its opponent.

- During a retreat maneuver, the SOAR agent would not try to confuse the enemy by changing heading or altitude. This often made the agent an easy target. This problem could be eliminated with a small amount of additional knowledge.

We presented our initial results to the SOAR community so that the SOAR agent knowledge base could be enhanced. Using these results in conjunction with their own insights, the SOAR community produced a far more robust agent. For benchmarking purposes, we ran the new SOAR agent against the same set of case-based agents that we had used in our first phase of experiments.

After the inital face-off, the enhanced SOAR agent showed marked improvements over the original agent. In this new round of experiments, the SOAR agent defeated the case-based agent in almost every encounter (thus accomplishing a primary goal of the case-based IFOR task). Still, we were able to uncover an interesting error in which the SOAR agent would hang indefinitely in an elaboration cycle. This turned out to be due to an error in the way certain conflicts were resolved. The problem, once identified by the experiments, was fixed by a change in the structure of the SOAR problem spaces.

Table C1. Experimental results obtained from competing the SOAR IFOR against the case-based IFOR show that the modifications to the SOAR IFOR yielded substantial performance improvements. Still, some important errors were found as well.

		INITIAL SOAR AGENT		IMPROVED SOAR AGENT	
TACTIC	GEOMETRY	OUTCOME	COMMENTS	OUTCOME	COMMENTS
A	HEAD-ON	RED	red fires, blue turns but is hit	BLUE	blue fires first
A	SIDE	RED	both fire - super missile wins for red	BLUE	red turns but is hit
A	CROSS	TIE	blue doesn't follow missile	BLUE	red turns but is hit
A	POP-UP	RED	red shoots immediately	?	soar hangs, elaboration bug
A	BLIND B	RED	red follows and kills	RED	red follows and kills
A	BLIND R	TIE	blue doesn't see red as hostile	?	soar hangs, elaboration bug
B	HEAD-ON	RED	blue doesn't react	BLUE	red evades but blue shoots first
B	SIDE	RED	blue doesn't follow missile	BLUE	red evades and shoots, but blue shoots first
B	CROSS	RED	blue doesn't follow missile	BLUE	red evades and shoots, but blue shoots first
C	HEAD-ON	RED	blue doesn't react	BLUE	red evades but does not shoot
C	SIDE	TIE	blue doesn't react, red misses opportunity	BLUE	red evades but does not shoot
C	CROSS	BLUE	red doesn't evade missile	BLUE	red evades but does not shoot

The face-off and enhancement cycle was iterated several times in order to further improve SOAR IFOR performance. Once the case-based IFOR was exposed to the enhanced SOAR agent, we created new tactic sets which could be applied in the next face-off. In the new scenarios that resulted, the case-based IFOR again could defeat the SOAR agent. By repeatedly pursuing this process, we can continue to identify areas in the SOAR knowledge base that might be improved. The most important advantage of this process is that we can always benchmark the improvements made in the SOAR knowledge base by running the SOAR agent against the existing case-based agents.

Experiments with 2v4 Scenarios

More complex engagement exercises were conducted to evaluate the scalability and complexity issues when larger numbers of IFOR are involved. The most complex one conducted in this program was a 2v4 scenario, comprised of 2 Blue Bogeys and 4 Red planes. The knowledge acquisition process utilized for this scenario is that described in Appendix B. The results shown here are taken from a final run of the simulation after a sufficient number of cases (as determined by the tactics expert) were acquired to guide the decision making strategy, and viewed by the expert as sufficient for the exercise. A total of 87 cases were acquired for this scenario, including a contingency plan and all symmetrical combinations of maneuvers. Appendix B illustrates the cases developed for the Lead Planes of the Lead Formation and Support Formation in achieving these results.

The simulation results for the initial phase of the scenario are shown in Figure C2.. The 4 Red Fighters fly as two formation pairs, maintaining two Combat Air Patrol (CAP) formations until a command decision initiates an investigation towards the Blue Bogey ingress direction. The results are displayed in a plan view of a 5000 x 5000 meter grid. The CAP behavior is illustrated in these results, as well as the formation flying behavior. As these planes fly, the CAP behavior keeps them flying in the racetrack pattern, while the formation flying behavior keeps them properly spaced apart. This illustrates one example of a multi-objective task being achieved through behavior-based concurrent control.

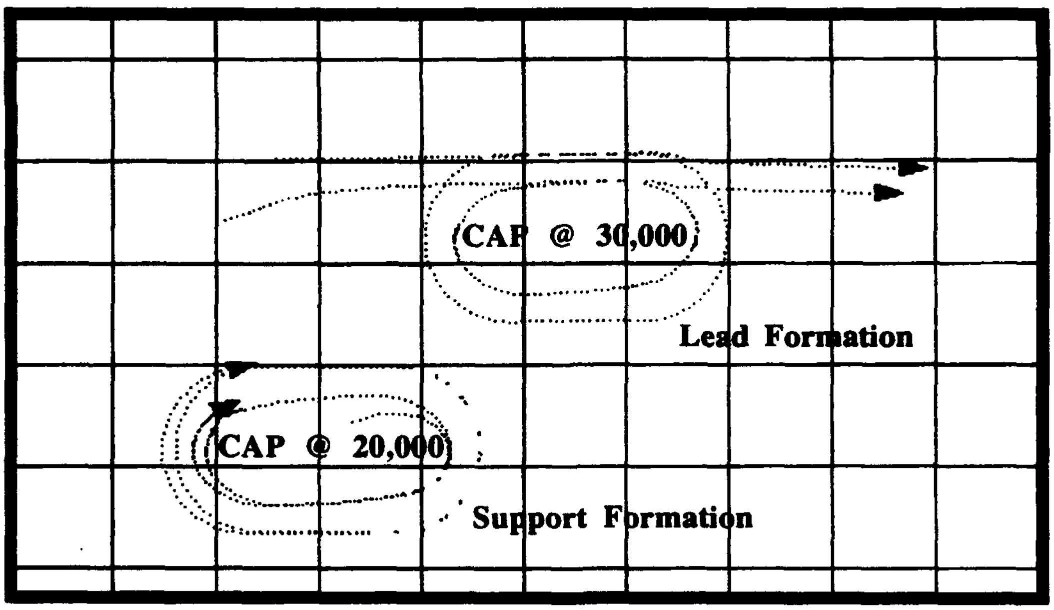


Figure C2. ModSAF simulation data shows how Case-Based agent control directs the Lead and Support Formations to fly two CAP flight patterns until information locating two Bogeys causes the Lead Formation to break the CAP. The Lead Formation executes a lead pursuit of the Bogeys while dropping down to the altitude of the Bogeys.

Formation flying is one of the cooperative behaviors that was required of the Case-Based IFOR in the development of a behavior repertoire. The formation flying is a primitive within the case-based representation, allowing a tactics expert simply to specify that a plane

(Wingman) is to fly in formation with a particular lead plane, rather than having to specify the detailed heading, velocity, and altitude commands that produce this result. Thus, during knowledge acquisition sessions, the tactics expert merely specifies the Wingman plane and the type of formation flying geometry to be executed, and it will be produced in simulation by activating the appropriate behavior-based controls.

The results of 2v4 engagements are illustrated for several of the possible outcomes of the scenario. The nominal situation is shown in Figures C3 and C4, where the Bogeys do not react to the actions of the Red agents. The launching of missiles by the Red agents ends in eventual ordnance impact, and the Red agents head home.

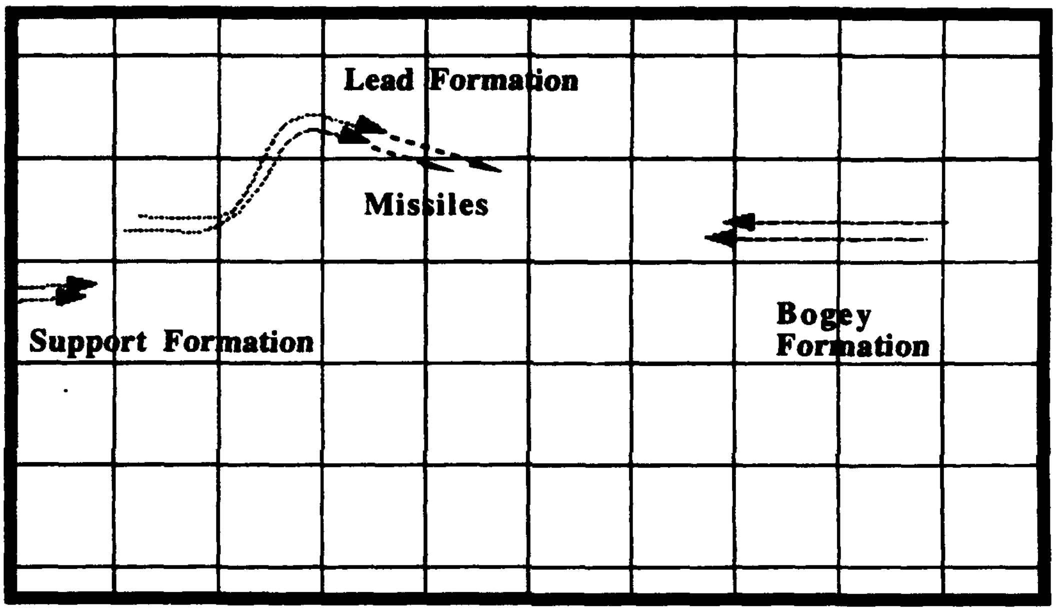


Figure C3. One of the possible branches that the scenario can take is that the Bogey planes never react to the Red agents. As shown in these simulation results, the Red planes will proceed with their nominal plan to perform a drag maneuver and acquire the lateral separation from the Bogeys for a good missile firing position. The Support formation planes are in a supporting posture in case the Bogeys react to actior of the Lead Formation.

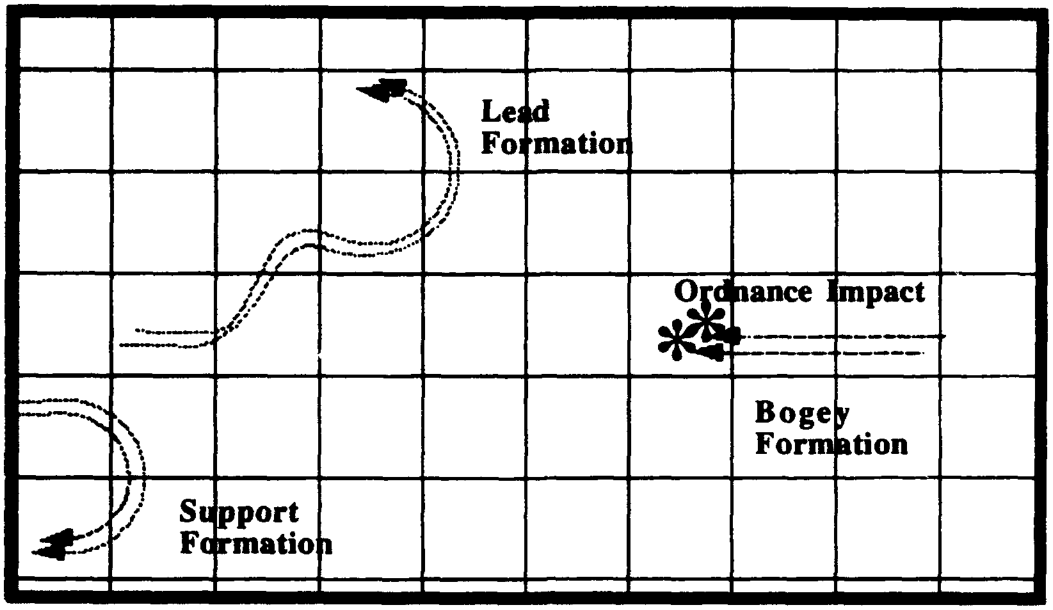


Figure C4. After an ordinance impact, the Lead and Support Formations go home.

The case database also includes all the cases for symmetric maneuvers in the scenario. For instance, if the Bogey formation is forward to the left, then the engagement will be started with a drag maneuver to the right; similarly, if forward to the right, the engagement will be started with a drag maneuver to the left. Figure C5 and Figure C6 present the simulation

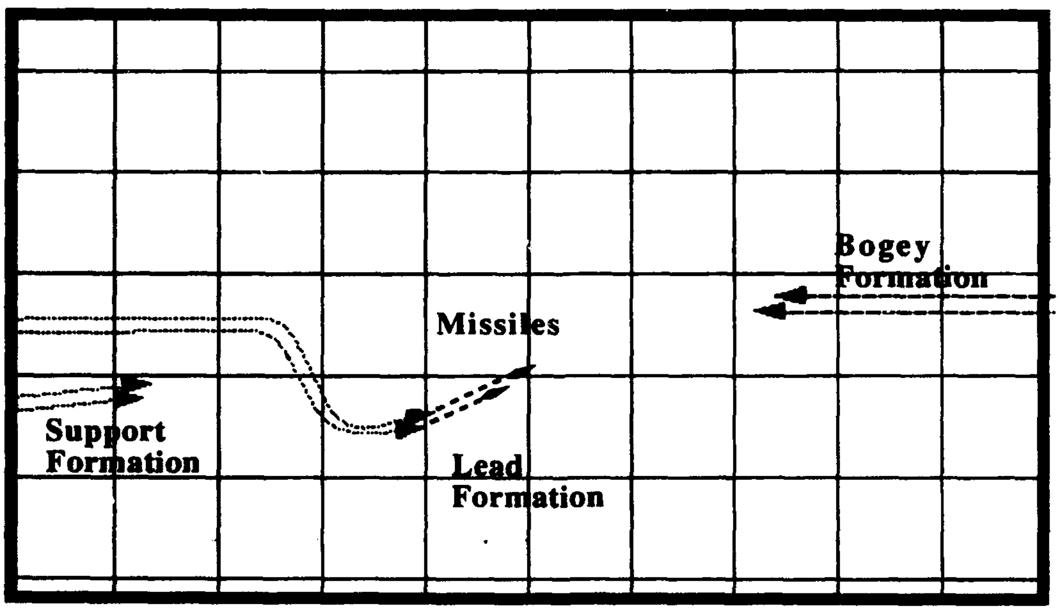


Figure C5. The case database also includes all the cases for symmetric maneuvers in the scenario. As shown in this simulation data, if the Bogey formation is forward to the left then the engagement will be started with a drag maneuver to the right.

results for the scenario when the engagement occurs to the right. Notice how the engagement occurs similarly to the previous results. Although the Support Formation is not directly behind the Lead Formation in either of these examples, the cases in the scenario still form matches and the tactics expert accepts the location of the Support Formation as fulfilling a supportive role.

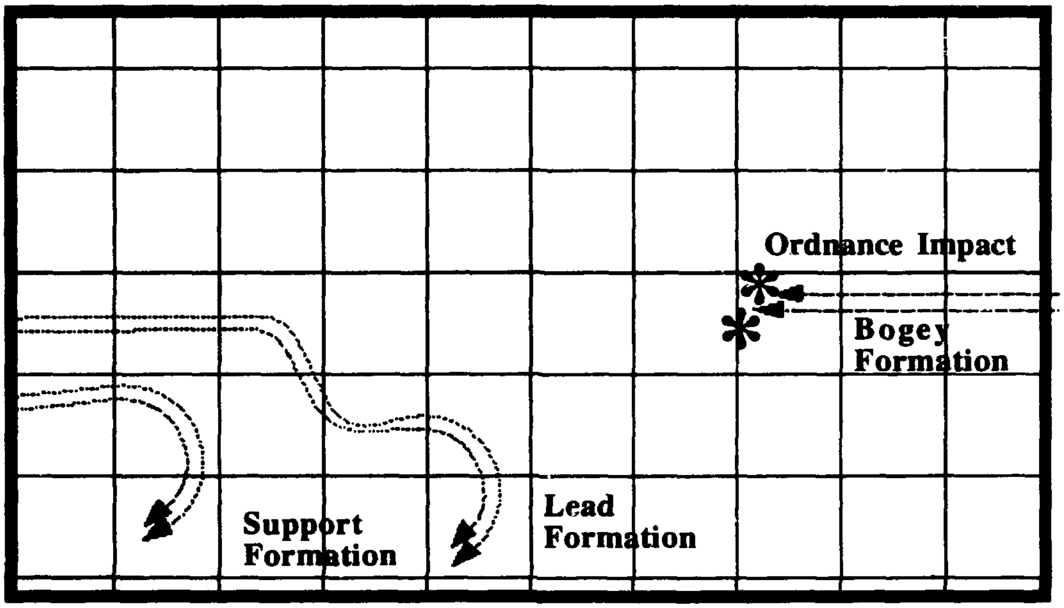


Figure C6. After an ordinance impact, the Lead and Support Formations go home.

As a contingency plan in this scenario, the Support Formation planes are in place for offensive action if the Bogey planes pursue the Red Lead Formation planes. Figures C7 and C8 show the simulation results for this possible outcome.

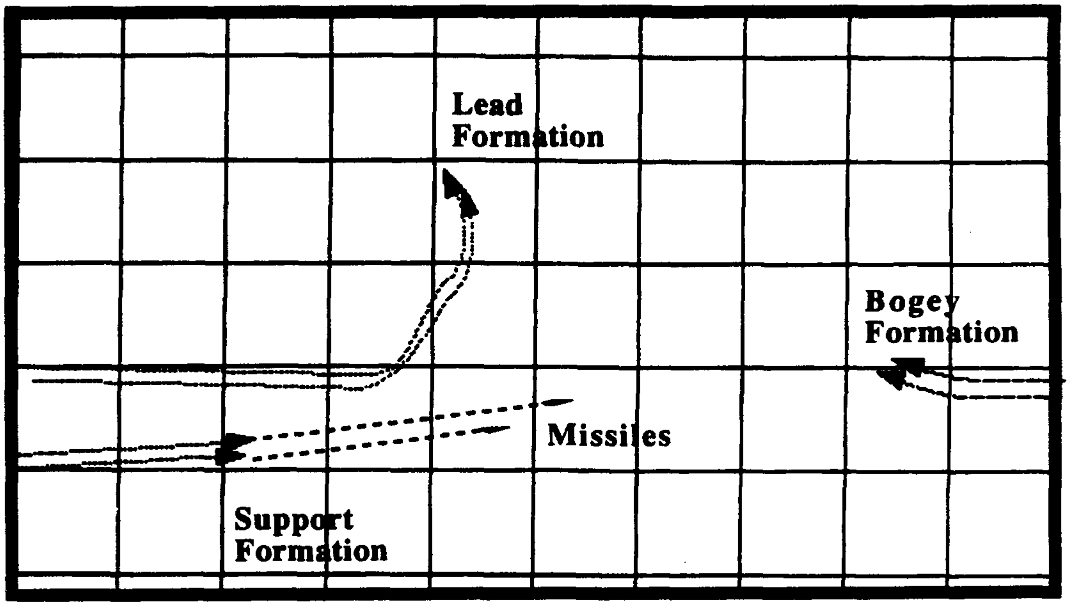


Figure C7. As a contingency plan in this scenario, the Support Formation planes are in place for offensive action if the Bogey planes pursue the Red Lead Formation planes. This simulation data shows how the Lead Formation will bug-out if the Bogeys pursue them; the Lead Formation assumes that the Support Formation is in place for an offensive strike.

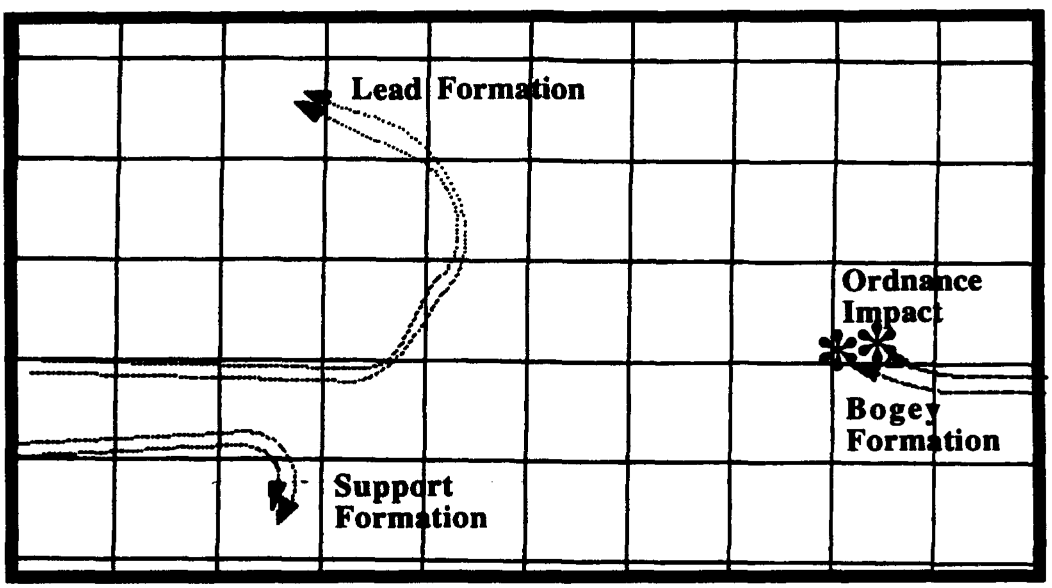


Figure C8. After an ordnance impact, the Support Formation and Lead Formation go home.

Appendix D: Description of Air Domain Behaviors

In BBN's ModSAF simulation, aircraft control is accomplished through basic command parameters such as heading, velocity, and altitude. In order to provide a higher level control for IFOR, we created a set of routines that are capable of manipulating the basic command parameters in such a way as to evoke behaviors that achieve specific mission objectives. Using our concurrent control paradigm, these behaviors may be blended to obtain a plethora of interesting tactical maneuvers. When using our case-based approach, each case specifies a particular combination of behaviors to be used in a certain situation or context. The behaviors, therefore, can be thought of as the essential primitives for control within our system. This section defines the behaviors implemented for the IFOR/WISSARD program.

The input to behaviors must characterize physical relationships between the IFOR and other entities in the simulation environment. For example, to fly in formation, an IFOR wingman must know where its lead plane is. Similarly, to attack an enemy, an IFOR must know the relative distance and orientation to that enemy. In a complex multi-agent scenario, however, there might be several candidate enemy planes. In order to obtain meaningful action from a behavior such as "pursue target," it is necessary for the IFOR to be in pursuit of only one enemy at any given moment. It is therefore helpful to organize an IFOR's perception of the world in terms of a more consistent and stable set of features. We call these features "markers."

Markers allow the assignment of labels such as "best target" or "nearest threat" to specific objects in an agent's environment. Markers are used by an IFOR to identify objects in accord with the critical roles these objects play in the IFOR's attack or survival responses. As the tactical situation evolves, yielding new threats or new target opportunities, the IFOR may assign its markers to different objects. This corresponds to a shift in attention as new information becomes available. Table D1 describes the markers used for the IFOR/WISSARD program.

Table D1. Cased-based Marker Set. Features of the markers were used in the case-based system to match the situation.

MARKER	DESCRIPTION
ATTACK-BOGEY	The designated hostile plane that the agent is pursuing or intends to attack.
THREAT-BOGEY	A hostile plane that has the best position to shoot at the agent.
OBSTACLE	The plane in front of the agent with the smallest time-to-impact distance to the agent.
SUPPORT-VEHICLE	Wingman or Leader of the agent. The designated friendly partner of the agent. Initially the closest friendly agent.
SUPPORT-FORMATION	The designated friendly that is tasked as either lead or support formation of the task force.
FIRED-MISSILE	The hypothesized status of the most recent launch missile by the agent.

Each marker has a set of attributes that are used to characterize the object it is tracking. For example, the ATTACK-BOGEY marker has attributes: MARKER-AGE, SPEED-RATIO,

SLANT-RANGE, ANGLE-OFF, TARGET-ASPECT, LATERAL-SEPARATION, and VERTICAL-SEPARATION. The definitions for these attributes are illustrated in Figures D1 and D2. Whether a marker is tracking an observable object, or is being used to track the hypothesized position of an unseen object, the marker has the same set of attributes. If the marker is tracking an observable object, then the attribute MARKER-AGE has the value zero. If the marker is hypothesized, then the attribute MARKER-AGE has a value proportional to the amount time since the last observation of its associated object. The other attributes are assigned values based either on the observed position and movement of the associated object, or on the estimates of these values if the object cannot be seen or is outside of sensor range.

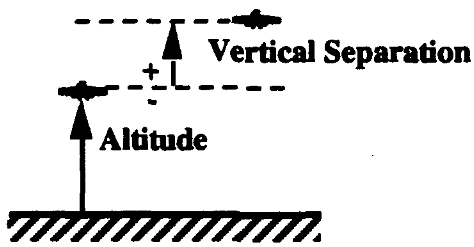


Figure D1. The vertical geometry of two planes flying into the page, showing the variables of interest: Altitude and Vertical Separation.

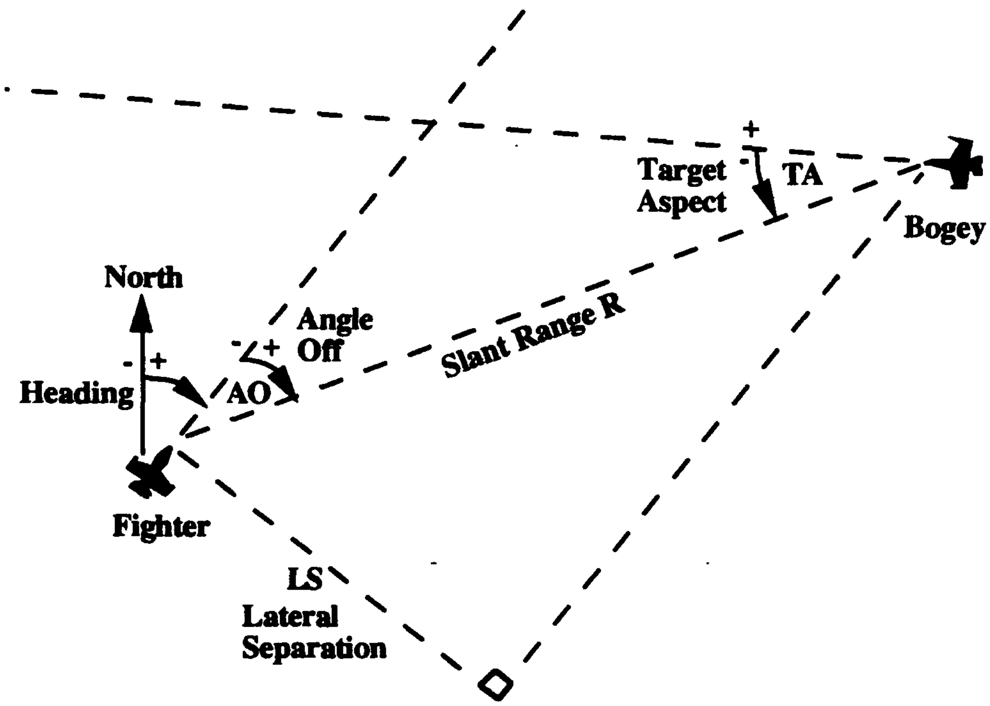


Figure D2. A plan view illustrating the important variables that describe the geometry of Bogeys relative to the Fighter.

The following behaviors were implemented to provide high level control to the IFOR:

ATTAIN-HEADING

Three ATTAIN-HEADING behaviors are used to specify the heading of the IFOR: ATTAIN-ABSOLUTE-HEADING commands the IFOR to a desired absolute heading direction; ATTAIN-RELATIVE-HEADING commands the IFOR to a desired relative heading with respect to the current heading of the vehicle; and ATTAIN-OFFSET-HEADING commands the IFOR to a desired relative heading with respect to the current heading of the ATTACK-BOGEY marker.

ATTAIN-SPEED

The ATTAIN-SPEED behavior commands the IFOR to attain a desired speed.

ATTAIN-ALTITUDE

The ATTAIN-ALTITUDE behavior commands the IFOR to attain a desired altitude.

PURSUE-TARGET

There are three PURSUE-TARGET behaviors which are designed to command the IFOR to close in on the target designated by the ATTACK-BOGEY marker. PURSUE-TARGET-HEADING issues a heading command based on the current heading of the vehicle relative to the position of the target; a "Lead-Distance" parameter allows for a lead pursuit, pure pursuit, or lag pursuit (Figures D3 and D4). PURSUE-TARGET-ALTITUDE attempts to attain and maintain the same altitude as the target minus the value of the parameter Vertical-Separation. PURSUE-TARGET-SPEED attempts to attain and maintain its speed relative to the target's speed; the speed is determined by the parameter "Speed-Ratio".

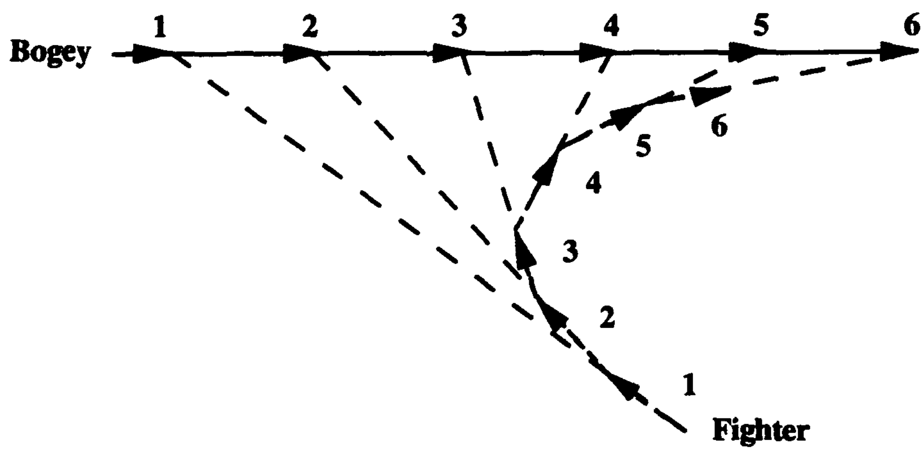


Figure D3. The Pure Pursuit option of PURSUE-TARGET-HEADING behavior. At every instant in time the Fighter pursues the exact location of the Bogey.

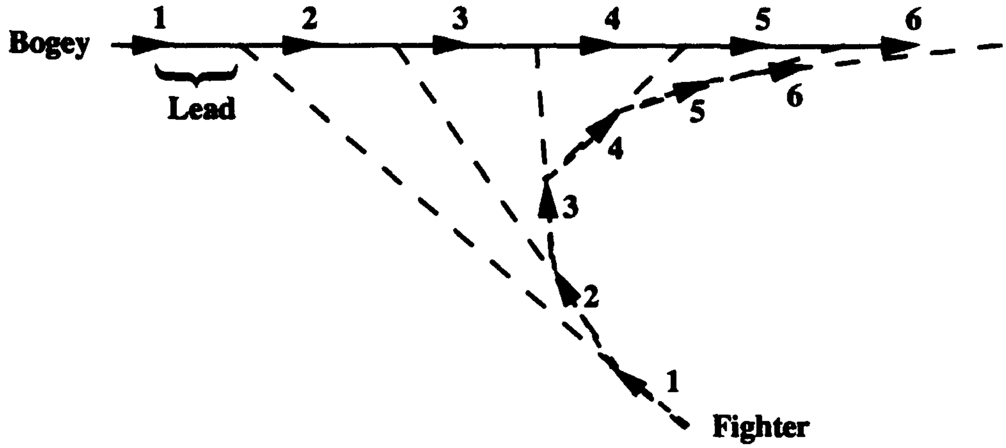


Figure D4. The Lead Pursuit option of PURSUE-TARGET-HEADING behavior. At every instant in time the Fighter pursues a point at a constant distance ahead of the Bogey.

MAINTAIN-FORMATION

There are three MAINTAIN-FORMATION behaviors which command the IFOR to maintain relative position to a supporting aircraft designated by the SUPPORT-VEHICLE marker. MAINTAIN-FORMATION-HEADING attempts to keep the same heading as the supporting aircraft by using a pursue scheme of control based on the current position of the support aircraft, and accounting for a position offset determined by the "Spread-Distance" parameter as shown in Figure D5. MAINTAIN-FORMATION-ALTITUDE tries to attain and maintain the same altitude as the supporting aircraft, minus an amount determined by the "Vertical-Separation" parameter. MAINTAIN-FORMATION-SPEED tries to attain and maintain a constant speed relative to the supporting aircraft, as determined by the parameter "Speed-Ratio."

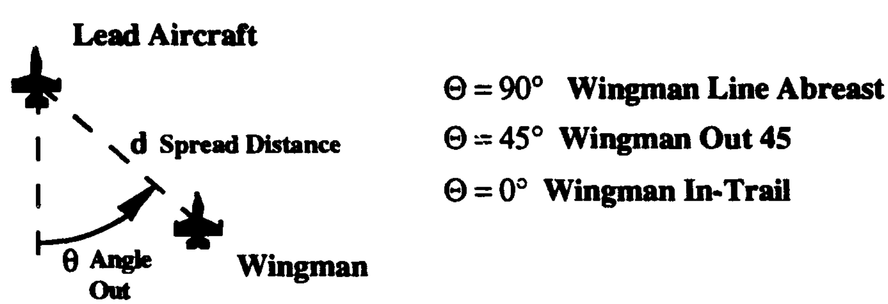


Figure D5. Formation Flying Parameters.

INTERCEPT-TARGET

The INTERCEPT-TARGET behavior commands the IFO… …aintain an intercept trajectory with respect to a target designated by the ATTAC…-BOGEY marker. A pure collision intercept or a lead collision intercept may be achieved by specifying the parameter "Lead-Angle." See Figures D6 and D7.

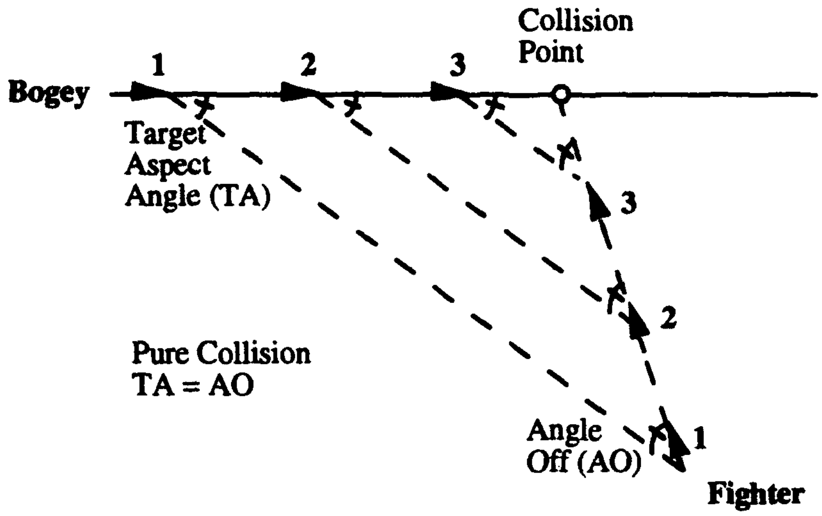


Figure D6. The Collision Intercept behavior, in which the Fighter maintains an angle off that is equal to the Target Aspect angle of the Bogey.

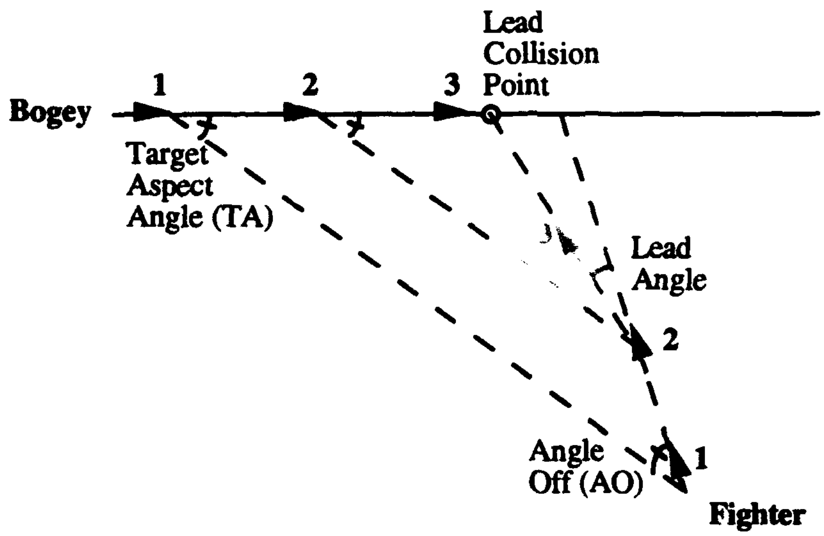


Figure D7. The Lead Collision Intercept behavior, in which the Fighter adds a lead angle to the angle off of the Collision Intercept.

AVOID-COLLISION

There are three AVOID-COLLISION behaviors which command the IFOR to make emergency maneuvers to avoid colliding with other aircraft designated by the *OBSTACLE* marker. The *OBSTACLE* marker is always attached to whichever aircraft has the shortest time to impact with the IFOR. AVOID-COLLISION-HEADING steers the IFOR away from the obstacle aircraft. AVOID-COLLISION-

ALTITUDE commands the IFOR to either dive or climb to avoid the obstacle. AVOID-COLLISION-SPEED alters the IFOR speed to prevent collisions.

MAINTAIN-CAP

The MAINTAIN-CAP behaviors maintain a Combat Air Patrol or CAP maneuver. The CAP geometry is shown in Figure D8.

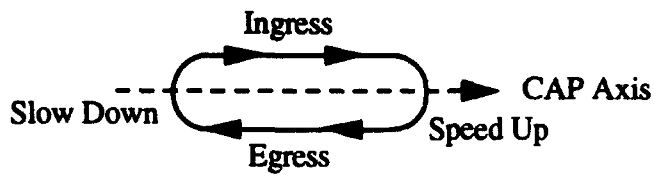


Figure D8. In the CAP behavior, the IFOR patrols in the CAP axis direction.

FIRE-MISSILE

The FIRE-MISSILE behavior commands the IFOR to fire a missile. The missile will be fired in the direction of the plane designated by the ATTACK-BOGEY marker. Once fired, a FIRED-MISSILE marker will be attached to the missile until it either hits its target, or runs out of fuel.

SUPPORT-MISSILE

The SUPPORT-MISSILE behavior constrains the heading of the aircraft so that the radar system can continue to guide the missile to the target. This behavior keeps the IFOR within a fixed range of orientations relative to the ATTACK-BOGEY and FIRED-MISSILE markers.

SOAR Integration

We implemented a SOAR/ModSAF interface to provide a compatible software environment to test and evaluate both case-based IFOR and SOAR IFOR agents. The ModSAF software provides a convenient basis for building distributed agents in a SIMNET/DIS simulation. This interface allowed real-time control of simulated IFOR using either SOAR-based or case-based control methods. Within the ModSAF simulation, any number and combination of SOAR, case-based, or ModSAF controlled IFOR can be specified, working in both cooperative and competitive modes. The interface consisted of code to: 1) incorporate the SOAR decision-cycle into ModSAF, 2) create a corresponding SOAR agent or case-based agent for each designated IFOR created in the ModSAF interface, 3) convert, in each SOAR decision cycle, the numeric information about the IFOR and its sensors into a symbolic form for each SOAR agent to use in its reasoning process, and 4) convert the controlling commands from SOAR or case-based agents into calls to the low-level control interface to ModSAF vehicles. The SOAR/ModSAF interface has been used by the SOAR consortium as part of their development process.

REFERENCES

[Agre 88] Agre, P.E., *The Dynamic Structure of Everyday Life*, Ph.D.. Thesis, M.I.T., 1988.

[Agre and Chapman 87] Agre, P.E. and Chapman, D., "Pengi: An Implementation of a Theory of Activity," *Sixth National Conf. on Artificial Intelligence*, Seattle, WA, pp. 268-272, July, 1987.

[Arai, et al 89] Arai, T., Ogata, H. and Suzuki, T., "Collision Avoidance Among Multiple Robots Using Virtual Impedance," *IEEE/RSG Intern. Workshop on Intelligent Robots and Systems*, Tsukuba, Japan, pp. 479-485, Sept., 1989.

[Franklin and Harmon 87] Franklin, R.F. and Harmon, L.A., *Elements of Cooperative Behavior*, ERIM, Ann Arbor, MI, Internal Research and Development Report 655404-1-F, Aug., 1987.

[Hammond 89] Hammond, K., *Case-Based Planning: Viewing Planning as a Memory Task*, Academic Press, 1989.

[Langton 89] Langton, C.G., *Artificial Life*, Addison-Wesley, 1989.

[Miller 90] Miller, D.P., "Multiple Behavior-Controlled Micro-Robots for Planetary Surface Missions," *IEEE Systems, Man, and Cybernetics Conf.*, Studio City, CA, Nov., 1990.

[Olin and Tseng 91] Olin, K. and Tseng, D., "Autonomous Cross Country Navigation," *IEEE Expert*, pp. 16-30, August, 1991.

[Payton, Rosenblatt, Keirsey 90] Payton, D.W., Rosenblatt, J.K., and Keirsey, D.M., "Plan Guided Reaction," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 20, No. 6, Nov./Dec., 1990.

[Payton 86] Payton, D.W., "An Architecture for Reflexive Autonomous Vehicle Control," IEEE International Conf. on Robotics and Automation, San Francisco, CA, pp. 1838-1845, April, 1986.

[Payton and Dolan 91] Payton, D.W. and Dolan, C.P., "Cooperative Control," *Seminars on Robotics in the Air/Land Battlefield NATO Defense Group*, Paris, France, March 6, 1991.

[Payton et. al. 91] Payton, D.W., "Do Whatever Works: A Robust Approach to Fault Tolerant Autonomous Control," *Journal of Aplied Intelligence*, Vol. 2, pp. 225-250, 1992.

[Preparata and Shamos 85] Preparata, F. P. and Shamos, M.I., *Computational Geometry: An Introduction*, Springer-Verlag, New York, NY, 1985.

[Rosenblatt and Payton 89] Rosenblatt, J.K. and Payton, D., "A Fine-Grained Alternative to the Subsumption Architecture," *Intern. Joint Conf. on Neural Networks*, Wash., D.C., June, 1989.