REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE 01 NOV 93	3. REPORT TYPE AND DATES COVERED THESIS/DISSERTATION

4. TITLE AND SUBTITLE MULTI-AGENT COORDINATION AND COOPERATION IN A DISTRIBUTED DYNAMIC ENVIRONMENT WITH LIMITED RESOURCES SIMULATED AIR WARS	5. FUNDING NUMBERS
6. AUTHOR(S) GREGORY D. ELDER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) AFIT Student Attending: ARIZONA STATE UNIV	8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/CI/CIA-93-25D
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) DEPARTMENT OF THE AIR FORCE AFIT/CI 2950 P STREET WRIGHT-PATTERSON AFB OH 45433-7765	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES	
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for Public Release IAW 190-1 Distribution Unlimited MICHAEL M. BRICKER, SMSgt, USAF Chief Administration	12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

14. SUBJECT TERMS	15. NUMBER OF PAGES 230
	16. PRICE CODE

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

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. Z39-18

DEPARTMENT OF THE AIR FORCE
HEADQUARTERS UNITED STATES AIR FORCE ACADEMY
USAF ACADEMY COLORADO 80840-6264

FROM: HQ USAFA/CWII (Maj Gregory D. Elder)
2354 Fairchild Dr, Suite 6A6
USAF Academy CO 80840-6264

1 Nov 93

SUBJ: Completion of PhD

TO: AFIT/CISS (Maj James P. Hogan)
2950 P Street
Wright-Patterson AFB OH 45433-6583

1. As I stated during our telephone conversation last week, I successfully defended my dissertation and have completed my PhD. Enclosed is one unbound copy of my dissertation and a letter of completion from the Registrar's Office at Arizona State University. The university will mail to you a final official copy of my transcript in December.

2. If you require any additional information, please call me at DSN 259-3250. If you have Internet access, you may direct electronic mail to me at eldergd%cwi@cwmail.usafa.af.mil.

Gregory D Elder

GREGORY D. ELDER, Major, USAF
Chief, Instructional Technology Division
Dept Cmdt for Military Instruction

2 Atch
1. Letter of Completion
2. Dissertation

DTIC QUALITY INSPECTED 5

Accesion For		
NTIS CRA&I		☒
DTIC TAB		☐
Unannounced		☐
Justification ______		
By ______		
Distribution /		
Availability Codes		
Dist	Avail and / or Special	
A-1		

Arizona State University

Office of the Registrar
Tempe, Arizona 85287-0312

To Whom It May Concern: October 15, 1993

RE: Gregory Dean Elder	461-23-9306
STUDENT'S NAME	I.D. NUMBER
Doctor of Philosophy	October 15, 1993
DEGREE	COMPLETION DATE
Computer Science	December 17, 1993
MAJOR	CONFERRAL DATE

This is to certify the student named above has completed all degree requirements as of the date indicated.

Conferral date and degree will be posted to the student's transcript after the grades are recorded at the close of the semester.

Sincerely,

Mary A. Mullen

Mary A. Mullen
Supervisor/Graduation Section

NOT OFFICIAL WITHOUT THE UNIVERSITY SEAL

OCT 15 1993

93-25D

MULTI-AGENT COORDINATION AND COOPERATION IN A DISTRIBUTED DYNAMIC ENVIRONMENT WITH LIMITED RESOURCES — SIMULATED AIR WARS

by

Gregory Dean Elder

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

ARIZONA STATE UNIVERSITY

December 1993

MULTI-AGENT COORDINATION AND COOPERATION IN A DISTRIBUTED DYNAMIC ENVIRONMENT WITH LIMITED RESOURCES — SIMULATED AIR WARS

by

Gregory Dean Elder

has been approved

August 1993

APPROVED:

____________________, Chairperson

____

____

____

____

Supervisory Committee

ACCEPTED:

____

Department Chairperson

____

Dean, Graduate College

ABSTRACT

Coordination and cooperation are two major issues of concern in Distributed Artificial Intelligence (DAI) systems. How can a group of geographically distributed agents properly allocate a set of tasks among themselves? Also, in an environment of limited resources, how can agents resolve resource conflicts so as to effectively accomplish tasks? This research has examined these two problems and has implemented techniques to promote multi-agent coordination and cooperation. A method of negotiation allows agents to bid for tasks based upon the agents' capabilities. Furthermore, the use of a threshold value ensures that only the best agents for a task become task commanders, as well as allowing some tasks to be re-negotiated as agents improve their bids. To resolve resource conflicts, a technique known as Hierarchical Iterative Conflict Resolution has been used. This technique allows conflicts to be resolved in an iterative manner, based upon a hierarchy of task priorities. Agents with higher priority tasks have preference for borrowing resources from agents with lower priority tasks. This ensures that higher priority tasks will be solved before those of lower priority.

These two techniques were employed in a DAI testbed which simulates an air war environment. Empirical studies were conducted using the testbed. The studies consisted of air war simulations between two opposing forces. During the simulations, tasks (air missions) had to be accomplished. Various decision making entities were examined in this study — human decision makers, distributed computer agents with resource sharing, distributed computer agents without resource sharing, and a single computer agent. The results of the studies indicate that within a dynamic and volatile environment, distributed agents sharing resources can achieve a higher level of task accomplishment than the other decision making entities.

ACKNOWLEDGEMENTS

I would like to sincerely thank my committee chairperson, Dr. Nicholas Findler, for allowing me the opportunity to work with him and his research group on a project of importance to the United States Air Force Academy. I am especially grateful for his knowledge, insight, and suggestions during this research. I would also like to thank my committee members, Dr. James Collofello, Dr. Jennifer Elgot-Drapkin, Dr. Subbarao Kambhampat, and Dr. Alyn Rockwood.

Thank you to Sudheer Dhulipalla for letting me incorporate portions of his communications package into the testbed constructed during this research. I would also like to express my thanks to the other members of the research group in the Artificial Intelligence Laboratory — Dr. Uttam Sengupta, Francis Annareddy, Serban Catrava, Raphael Malyankar, Umeesh Panchaksharaiah, Anand Pashupathy, and Sudeep Surender. I appreciate having been able to exchange ideas with them throughout my three years with the group.

I would also like to thank Major (retired) Joe Murphy for his suggestions and assistance in developing air war scenarios for my research. I am also grateful to Squadron Leader Rodger Dunsford, Captain Brad Barret, and Captain Ed Martin for giving up a week of their time to participate in some of the experiments and for providing me with valuable feedback concerning the testbed. Thanks to the United States Air Force for giving me this opportunity to pursue a PhD.

Lastly, I am deeply thankful to my family — my wife, Ann, and my two sons, Jonathan and William. Without their support and encouragement these past three years, I could never have succeeded.

TABLE OF CONTENTS

Page

CHAPTER Page

CHAPTER Page

CHAPTER Page

CHAPTER	Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure		Page

CHAPTER 1

INTRODUCTION

1.1 Problem Description

During an air war, the air commander must make specific decisions concerning the selection of targets, allocation of resources, assignment of tasks, and delegation of authority [United States Air Force, 1992]. According to basic Air Force doctrine as outlined in Air Force Manual 1-1, the air commander's decision making pattern in employing air resources is a continuous cyclic process as represented in Figure 1.1.

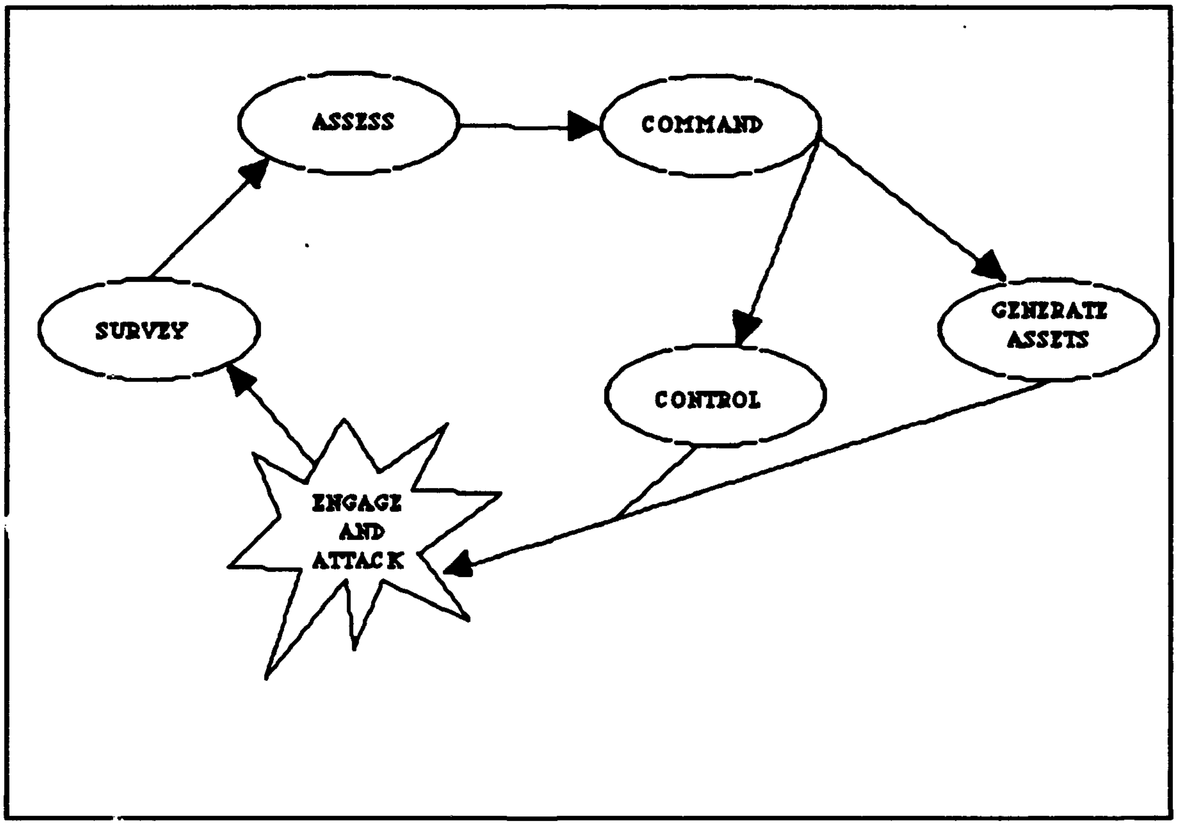

Figure 1.1. The air commander's decision making cycle for employing resources.

The air commander must *survey* the battlefield situation to identify both threats and targets of opportunity. Next, the commander must *assess* what needs to be accomplished in order to meet objectives and to establish priorities. The air commander must then *command* and *generate assets* — air units are identified and allocated in sufficient strength and in time to carry out required tasks. Once forces are allocated, the commander must *control* their actions to give direction to the overall effort to *engage and attack* the enemy. Finally, the results of engagements are evaluated and the process for employing forces is repeated.

The air commander's direct subordinates are the wing commanders. They are located at operational air fields within the theater of conflict and are usually geographically separated from both the air commander and each other. Each wing commander has control over a set of air resources under his command. When developing courses of action for employing forces, the air commander must coordinate the activities of the wing commanders for mission assignments and resource allocation. Using the strategic guidance established by the theater commander, the air commander and his subordinates must deal with two categories of air missions: offensive and defensive.

Offensive missions employ friendly resources to inflict damage upon enemy resources such that the enemy's war fighting capability is reduced or eliminated. The major resources available to the air commander include various types of aircraft (fighters, bombers, tankers, etc.), aircrew members (pilots, navigators, weapon control officers, etc.), and aircraft armaments (gravity bombs, cluster bombs, heat-seeking missiles, and so forth). The enemy resources targeted by offensive missions do include aircraft, but the more important resources are land-based targets. These consist of command centers, airfields, transportation centers, dams, power plants, factories, POL (petroleum, oil and lubricants) centers, munition depots, and any other ground resources which may affect the enemy's war making capability.

Defensive missions employ resources for the purpose of preventing the enemy from damaging the war fighting capability of friendly forces. The air commander will use fighter aircraft with associated air crews to intercept and deter enemy aircraft intent upon attacking friendly resources.

When planning for air operations, the air commander and his staff develop a set of specific missions or tasks. These missions are then assigned to wing commanders for accomplishment. In addition, resources must be allocated for each task. The basic problem to solve becomes two fold:

1) Which missions should be assigned to which wing commanders?

2) What is the best mix of available resources to accomplish each task effectively?

Since resources are limited and have different functionalities, they must at times be shared amongst the various wing commanders in order to fulfill mission requirements. Resource conflicts may arise when the number and types of required resources exceed the available resources for a given set of missions over certain time windows.

Offensive missions are normally planned well in advance and are concerned with tasks that are to be accomplished within the range of hours to a few days. The commander and his staff have ample time to analyze the current battle conditions before making specific decisions which impact offensive operations. Since the majority of offensive targets are fixed, land-based resources, temporal constraints are at a minimum for offensive missions (e.g., ground targets will not change position over time).

Unlike offensive missions, defensive missions cannot be planned well in advance. A commander can never know with absolute certainty what actions the enemy would take.

Therefore, defensive missions are reactive in nature, meaning that the air commander and his staff must be able to handle "unscheduled" tasks (the interdiction of enemy aircraft over friendly territory, for example). With defensive missions, tasks must be accomplished within the range of minutes to a few hours. Temporal constraints are highly likely with defensive operations. Airborne enemy aircraft will, of course, continuously change position over time. If they are not intercepted within a certain time period, friendly ground resources can be lost.

The process of developing air war missions involves a staff of human experts analyzing resource and environmental data collected by geographically distributed sensors and other humans. The desired result is an equitable assignment of tasks and optimal allocation of resources for achieving desired goals. This process is very time consuming. In addition, humans working under time constraints and the pressures of warfare, combined with an abundance of data (information overload) can be prone to error. Thus, resources may not be allocated in as timely a manner or optimal amount as possible.

1.2 A Distributed Artificial Intelligence Approach

Due to the complex and knowledge-intensive nature of warfare, automated systems have been developed to assist human decision makers operating in a military environment (Position Locator Reporting System, Joint Tactical Information Distribution System, and automated fire support systems, to name a few) [Dunnigan, 1988]. Furthermore, while the performance of computers and communications systems have improved, the associated hardware costs have declined. This has led to decentralized approaches for computer systems in such areas as distributed control systems and distributed data bases. Taking a distributed approach provides certain advantages: faster response, increased flexibility, robustness, resource sharing, graceful degradation, and better adaptability. Traditional

Artificial Intelligence (AI) systems have taken centralized approaches in such areas as game playing, search strategies, expert systems, and natural language understanding — basically, modeling the intelligent behavior of a single agent. In recent years, researchers have also been taking a distributed approach to AI to model the behavior of groups of agents. Certain aspects of the air war environment lend themselves to the use of Distributed Artificial Intelligence (DAI) to provide assistance to the commanders.

DAI can be defined as a special area within AI concerned with the cooperative solution of problems by a decentralized group of agents [Huhns, 1987]. The agents may range from simple processing elements to more complex entities which behave in a rational manner. Problem solving as well as task accomplishment is collaborative since the group of agents must share information and resources in order to successfully produce a plan for the solution and to execute it. Furthermore, the agents are decentralized in that both data and control are logically, and in many cases physically, distributed.

According to Huhns [1987], there are five reasons for which one would want to use and study DAI:

1. to provide insight and understanding about humans who organize themselves in order to solve problems;

2. to provide a means of interconnecting different problem solving systems so as to solve problems whose domains are outside the bounds of any one problem solving system;

3. to solve problems that are too large for a centralized system;

4. to overcome a current limitation of knowledge engineering: the use of only one expert;

5. to solve problems which are inherently distributed.

Additionally, Findler [1990] provides other factors which point towards the use of a DAI strategy for problem solving:

- geographically or functionally distributed resources;
- time-criticality of needed solutions;
- the need for reliable computation when dealing with "uncertain" knowledge;
- the need for graceful degradation.

Recent research involving DAI has included distributed vehicle monitoring [Durfee, Lesser, & Corkill, 1987]; distributed air traffic control [Findler & Lo, 1991; Lo, 1988; Lo & Findler, 1991; Steeb et al., 1981]; distributed manufacturing control [Ge & Findler, 1988; Parunak, 1987; Parunak, 1990]; distributed control of traffic lights [Stapp, 1990]; distributed problem solving for knowledge based vision [Decker, Durfee, & Lesser, 1989]; multi-agent motion planning [Fraichard & Demazeau, 1990]; and distributed control of moving resources [Sengupta, 1991].

Characteristics of the air war environment enable the use of a DAI methodology for solving air war problems. Consider the following aspects:

- Commanders (agents) are geographically distributed throughout the military theater of operations.
- The commanders must accomplish a common set of tasks (problems).
- Resources are both geographically and functionally distributed.

- Certain tasks are time-critical.

- The environment is volatile (resources change their location, are lost and replenished).

1.3 Research Goals

This research is concerned with applying the principles of DAI to an air war environment so as to model the behavior of cooperating agents within this domain. A Distributed Problem Solving (DPS) testbed has been constructed to allow empirical studies to be conducted concerning multi-agent communication, cooperation, task distribution, resource sharing, and management of resource conflicts. A task distribution technique has been implemented which lets agents evaluate a given set of tasks and become self-appointed task commanders. Another mechanism has been implemented which provides for the resolution of resource conflicts in a hierarchical and iterative manner.

The testbed simulates the salient features of an air war environment so that air wars between two opposing forces can be simulated. This includes a set of problem solving/decision making agents (commanders) in charge of specific air resources. These resources possess different functionalities and are in various states of readiness (availability) during an air war simulation. Additionally, ground-based resources (land targets) are available. Offensive and defensive tasks associated with air missions must be appropriately distributed among the agents for accomplishment. The agents cooperate with one another for assignment of tasks, resource allocation and sharing, and the resolution of conflicts. This system also simulates the fog and friction (uncertainty) prevalent in warfare.

The primary goal of this research is to show that DAI is a viable strategy for the problems of assigning tasks and allocating resources within the air war environment. The specific objectives which underlie this goal are as follows:

- Develop a DAI methodology for task distribution, resource allocation, and conflict resolution for both offensive and defensive air war missions.
- Design and construct a testbed which implements the above methodology in a simulated air war environment.
- Perform empirical studies to analyze the effectiveness of this approach.

1.4 Generalization of the Research

While this research used the air war environment as a basis for studying the issues of concern, the problems being examined and their solutions are general enough to apply to other homomorphic distributed environments. The generic DAI problems addressed in this research are:

- How can a group of geographically distributed agents allocate a set of tasks among themselves?
- How can a group of geographically distributed agents share limited resources to achieve an effective level of task accomplishment?

1.5 Dissertation Organization

The remainder of this dissertation is organized as follows. Chapter 2 consists of a literature survey focusing upon not only DAI research, but AI research related to military systems as well. Chapter 3 describes the overall DAI methodology developed for use in this research. In Chapter 4, the architecture, design, and implementation of the Distributed Air War testbed is explained. Next, Chapter 5 discusses the empirical studies, covering the design of experiments, computer runs of simulated air wars, and the analysis of the results. Finally, Chapter 6 summarizes the overall findings of the research and describes areas for future research.

CHAPTER 2

LITERATURE SURVEY

2.1 Issues in DAI

"DAI is concerned with solving problems by applying both artificial intelligence techniques and multiple problem solvers" [Decker, 1987, p.729]. With multiple agents working to accomplish a common task or problem, certain issues of concern must be addressed. The first of these is *cooperation*. Cooperation involves the pooling of agents' expertise, knowledge and resources to solve a complex problem that no single agent can solve on its own. Another important issue to consider is *coordination*. How do the agents decompose problems and distribute sub-tasks to one another? In other words, in what manner are the actions of single agents coordinated so that the agents as a whole work together effectively? Finally, cooperation and coordination require that agents interact with one another by exchanging information. This brings to light a third important issue—*communication*. How do the agents communicate with each other?

2.1.1 Cooperation

Cooperation is an important aspect of DAI. In some cases, agents may operate independent of one another without the need for interaction. However, the scope of some problems may be such that agents must cooperate in order to arrive at a global solution. For example, multiple expert systems covering diverse domains may need to share knowledge so that a problem which encompasses portions of their domains can be solved. Additionally, agents may have to cooperate by sharing a limited set of resources which are available for accomplishing tasks. Some of the goals of cooperation include [Durfee, Lesser & Corkill, 1987]:

- to improve performance;
- to increase the variety of solutions;
- to increase the probability that solutions will be found;
- to reduce communication via selective message exchanges; and
- to improve the use of resources by exchanging tasks to better balance the workload.

The nature of the agents has a direct impact on cooperation. Cooperation occurs when an agent adjusts its goals and intentions (plans for action) to assist another agent [Werner, 1990]. Thus, in a sense, an agent believes it will benefit by cooperating with others. Negotiation strategies can be used to enhance cooperation amongst a group of agents [Lo, 1988; Parunak, 1987; Smith, 1980; Smith & Davis, 1981].

2.1.2 Coordination

Closely related to cooperation, coordination involves the manner of how information and resources are shared between a group of agents, and how tasks are distributed. Approaches to coordination, include the following [Rich & Knight, 1991]:

- Autocratic—one agent is in charge who assigns tasks to other agents. The other agents do as they are told and report results back to the "master" agent.
- Contractual—one agent is in charge who announces available tasks. Agents negotiate for tasks and are awarded "contracts" for task accomplishment.

- Democratic—no one agent is in charge, however, there is a shared goal among the agents. Agents work together in determining and accomplishing tasks.

- Anarchistic—no one agent is in charge and there is no guarantee that goals will be shared among agents. Agents may even compete with one another.

One of the earliest, and perhaps best known, DAI coordination techniques is the contract net [Davis & Smith, 1983; Smith, 1980]. In a contract net, agents assume the roles of managers and contractors. Manager agents decompose tasks into sub-tasks and announce the sub-tasks one at a time to the remaining agents. The other agents submit bids for the sub-tasks to the manager, who in-turn awards the sub-tasks to specific agents. Agents that "win a bid" then become contractors. In essence, a contract exists between the manager and the agent responsible for a particular sub-task. As sub-tasks are accomplished, contractors report the results to the manager. Furthermore, some sub-tasks themselves may be further decomposable. In this case, the respective contractor agent may also become a manager for its sub-task. The agent would decompose its sub-task into smaller pieces and then go through a bidding process to assign contractors for each part. Hence, agents can assume the roles of both manager and contractor.

As a part of his research, [Lo, 1988] modified the basic contract net scheme. His approach divides agents into coordinators and coworkers. Rather than announcing sub-tasks one at a time, coordinator agents announce entire lists of available sub-tasks. Other agents then submit bids for each sub-task in the list for which they desire to be responsible. The coordinator examines each bid list and then assigns sub-tasks accordingly to agents which become coworkers. Other research involving ideas based upon contract nets includes work by [Cammarata et al., 1983; Ge & Findler, 1988; Parunak, 1990].

2.1.3 Communication

Agents may coordinate activities without communicating with one another [Rich & Knight, 1991; Rosenschein & Breese, 1987]. Agents possess meta-models of one another which they use in making rational decisions. A basic technique is to use aspects of game theory, such as a payoff matrix. In most cases, cooperation and coordination require agents to communicate with one another. The two communication architectures that have been used with DAI are *blackboard systems* and *message-passing systems*.

2.1.3.1 Blackboard Systems

Blackboard systems use a shared knowledge structure (memory) called a blackboard for communication [Englemore et al., 1988; Nii, 1986a; Nii, 1986b; Rich & Knight, 1991]. Agents, known as knowledge sources, can read and write to the blackboard. Each knowledge source is typically an expert in a particular sub-field of a domain. Problems are posted to the blackboard. Knowledge sources operate concurrently and opportunistically to solve the problem. As knowledge sources solve parts of the problem, partial solutions are also posted to the blackboard. These partial solutions may further trigger other knowledge sources into refining the solution. This process continues until an overall solution is obtained. A good analogy of a blackboard system is described by [Nii, 1986a]. Consider a group of people attempting to solve a jigsaw puzzle. Each person holds a number of pieces to the puzzle. The blackboard is a physical board upon which pieces of the puzzle can be placed. Each person takes their most "promising" puzzle piece and places it on the blackboard. Next, each person examines their remaining pieces to determine if any of them fit what is already on the board. Those that do fit are placed onto the blackboard. These new updates will cause other pieces to "fall into place." In this manner, the blackboard is updated opportunistically. One drawback to blackboard systems is the shared knowledge structure. If the hardware or software involved with the knowledge structure fails, the agents cannot communicate with each other.

2.1.3.2 Message-Passing Systems

With message-passing systems, all agents can communicate directly with one another to exchange information. Agents tend to know more about one another in such systems than in a blackboard architecture [Rich & Knight, 1991]. This knowledge allows agents to be selective when communicating with others. Some messages may be directed to all agents, e.g., a manager agent in a contract net announcing an available task at large. In some instances, an agent may send a message to only a subset of the other agents. For example, an agent may need access to a certain type of resource which two out of a group of five agents possess. The first agent would direct resource request messages to just the two agents known to have that type of resource. Since communication may be costly, many DAI systems which use message-passing attempt to minimize the amount of communication between agents.

2.1.4 Conflicts

Depending on the characteristics of the agents and the nature of tasks, conflicts may arise in the distributed environment. Agents could have individual goals which conflict with one another. Robots navigating a room may have paths that conflict and robots could collide. Another example involves resources. Agents have access to limited resources over the same time window. Conflicts can occur if more than one agent requires the use of the limited resource. Galliers [1990, p. 40] defines multi-agent conflict as "...when the agents' beliefs or goals with respect to the same proposition are believed by the one agent to be in opposition, and this agent is also committed to a goal to change the other's belief or goal."

Conflict resolution has been investigated by others [Adler et al., 1989; Galliers, 1990; Sycara, 1988]. One way to resolve conflicts is to impose a master-slave relationship between the agents. The master agent resolves all conflicts and imposes its desires upon the other agents according to its knowledge and interpretation of the environment. Another resolution technique involves negotiating. The conflict essentially becomes another task to solve. Agents negotiate for the task, and the agent which wins the task then handles the conflict. In this manner, no one agent is in charge of dealing with all conflicts. Sycara's persuasive argumentation can also be used to resolve conflicts. In this case, each agent would determine how to resolve the conflict and would then attempt to convince the other agents that its solution was best.

2.2 DAI Systems and Testbeds

2.2.1 Hearsay-II

Hearsay-II, implemented in the early 1970's, is one of the first blackboard systems [Erman et al., 1988; Lesser & Erman, 1988; Nii, 1986b]. Hearsay-II is a speech understanding system which recognizes a 1000-word vocabulary and has a 90% success rate in correctly understanding spoken sentences. The basic problem it solves is the correct interpretation of spoken sound. The Hearsay-II blackboard is divided into distinct knowledge levels. Each level corresponds to a different aspect of the problem space, e.g., segments, syllables, words, phrases, and sentences. Knowledge sources operate concurrently upon the data stored on the blackboard to generate, combine, and evaluate hypothetical interpretations of speech. Each knowledge source has a functional expertise related to speech interpretation, such as signal acquisition, word spotting, phrase generation, and so forth. Any knowledge source can create new hypotheses on the blackboard or modify existing ones. Starting with a spoken utterance placed on the blackboard, the knowledge sources build hypotheses about the various parts of speech until a correct sentence is produced. Hearsay-II was integrated with a database system and could successfully answer questions and perform database functions when given spoken commands. Some example spoken commands which Hearsay-II could interpret were:

"Which abstracts refer to theory of computation?"

"List those articles."

"What has McCarthy written since 1974?"

The Hearsay-II project showed that blackboard systems represent a viable distributed problem solving paradigm.

2.2.2 The Distributed Vehicle Monitoring Testbed

The Distributed Vehicle Monitoring Testbed (DVMT) project is another early research effort into DAI [Durfee et al., 1987; Lesser & Corkill, 1983]. The testbed simulates distributed agents solving the problem of creating a dynamic map of vehicles moving through a monitored area. Each agent possesses an acoustical sensor to detect vehicle sounds. Data from a single sensor can contain significant amounts of error. Therefore, over time, the agents exchange, correlate, and interpret acoustic signals from all sensors in order to generate a correct map of vehicle movement.

Each agent in the DVMT is a complete blackboard system, along the lines of Hearsay-II. An individual agent is capable of solving the overall mapping problem if given all sensory data and knowledge. In addition, each agent has a special knowledge source used for communicating hypotheses and goals to the other agents. Thus, the agents cooperate by exchanging partial hypotheses concerning characteristics of data collected by the sensors.

The DVMT is implemented in CLISP and runs on a VAX computer system under VMS. Since the DVMT operates on a single processor, the distributed agents must be simulated. As such, the system is slow, requiring 3-5 hours of CPU time for moderate size test cases. Experiments with the DVMT have examined agent communication strategies to include voluntary (an agent transmits hypotheses at pleasure), requested (an agent transmits hypotheses only when requested by another agent), and mixed (a combination or voluntary and requested).

2.2.3 Multi-Agent Computer Environment

The Multi-Agent Computer Environment (MACE) is a testbed for building a wide variety of experimental DAI systems at differing levels of granularity [Gasser et al., 1987]. MACE maps agents onto processors, handles inter-agent communication via message-passing, and provides the following:

- A language for describing agents;
- Tracing and instrumentation;
- A facility for remote demons; and
- A collection of system agents.

Thus, MACE is a language, programming environment, and a testbed for the study of DAI. While other DAI testbeds have been suitable for only one type of problem-solving architecture, MACE is intended to be useful with various kinds of architectures.

MACE has also been described as a distributed, object-oriented system. The main objects provided by MACE include:

- Agents - These are the basic computational units of MACE. Agents have knowledge of the environment and are aware of other agents. Furthermore, agents may be organized into coalitions which act in response to certain problems.

- System Agents - These are pre-defined and provide command interpretation, a user interface, error handling, tracing, and execution monitoring.

- Facilities - These are standard functions which all agents can use. Some of these are a pattern matcher, a simulator, and standard messages.

- Description Database - Maintains agent descriptions.

- Kernels - Handles communication, message routing I/O, and maps agents onto processors.

MACE is written in LISP and runs on a 16-node Intel SYM-1 hypercube machine. Various DAI paradigms have been implemented and tested with MACE. For example, a contract net system and a blackboard system have both been implemented on MACE. Also, MACE has been used to model low-level parallelism via a production rule system whereby each rule was an agent. Since MACE operates in a true multi-processor environment, much of the limitations of earlier testbeds which run on single processors have been overcome.

2.2.4 The Distributed Air Traffic Control Testbed

The Distributed Air Traffic Control Testbed (DATCT) is designed to study the control of air traffic without human intervention [Findler & Lo, 1986; Findler & Lo, 1991; Lo, 1988; Lo & Findler, 1991]. In this environment, each aircraft possesses a processor (agent) which can communicate with other agents within a certain region of airspace. The agents cooperate with one another to resolve incidents that may arise, such as violation of Federal Aviation Administration rules (for example, aircraft flying too close to each other).

Every agent maintains a flight plan. Agents also maintain a simulated world that reflects the surrounding environment visible to their radar scope. By extrapolating into the future, agents look ahead to detect incidents. When incidents are detected, the agents exchange flight plans and negotiate to select one agent to be a coordinator. The coordinator is responsible for resolving the incident. This will involve making changes to agents' flight plans. If an incident represents a large problem which is decomposable, then agents may place bids with the coordinator to become coworkers to solve the sub-problems. In this manner, a hierarchical relationship is built between coordinators and coworkers.

The DATCT is written in LISP and operates on a VAX computer system under VMS. Like the DVMT, the distributed agents are simulated on a single processor. This makes it difficult to determine the speed-ups expected when using a DAI strategy.

2.2.5 SENTINEL

The SENTINEL system is a DAI testbed for studying problems of planning for resource allocation under constraints of limited time and resources, and for testing different communication protocols [Sengupta, 1991]. The testbed simulates the command, control, and communication operations of the United States Coast Guard (USCG). Agents in this system plan for the interception of suspect vessels which may arrive in regions of USCG control at anytime. Each agent has jurisdiction for a specific area of the coast line. Suspect vessels entering an area become the responsibility of the respective agent. Using a rule-based approach, agents construct a plan to intercept these vessels as they arrive. The agents in SENTINEL use message-passing to communicate and cooperate with each other.

SENTINEL uses a relaxed organizational scheme for the agents. The strict USCG command hierarchy results in slower decision making. To improve the decision making process, the agents use a constrained lattice-like organizational structure. This allows some agents to communicate directly with each others, rather than routing messages through the hierarchical chain of command. Another concept used by SENTINEL is dynamic scoping. If an agent cannot accomplish a task due to insufficient resources, the agent will request resources from its two closest, adjacent neighboring agents. If these agents cannot supply the needed resources, then the request is extended to the next two farther agents. This process of extending resource requests continues until enough resources are available, or all agents (within a so-called 'envelope of effectiveness') have been contacted. Initial experiments with SENTINEL indicate that dynamic scoping does promote inter-agent communication and better resource utilization. Also, the use of the lattice-like organizational structure resulted in a higher suspect interdiction rate, as compared to using a strictly hierarchical chain of command arrangement.

2.3 Artificial Intelligence and Military Systems

"Military commanders and their staff have been processing information and making decisions long before the arrival of either computers or the development of AI" [Leedom, 1984, p. 61]. Ages past, armies might travel for days or weeks on foot before engaging the enemy. However, in today's modern world with advanced military technology, time and space have become compressed on the battlefield. Air strikes can reach opponents in a matter of hours. The modern commander must process a wealth of information in a short amount of time. According to [Lehnert & Sullivan, 1989], modern battlefield planning is beyond the reach of conventional automation techniques. While traditional automated approaches can assist the commander with data management, the best hope for improving the process of battle management is AI [Bonasso, 1988]. As such, research has been conducted which applies AI to the military environment.

2.3.1 AirLand Loosely Integrated Expert System

The AirLand Loosely Integrated Expert Systems (ALLIES) integrates three separate AI systems for research in command and control [Benoit et al., 1986; Tachmindji & Bonasso, 1988]. ALLIES consists of an expert system for constructing mission plans (OPLANNER), an expert system for analyzing enemy forces (ANALYST), and a system which models the battlefield (BEM—Battlefield Environment Model). OPLANNER is a hierarchical, divide-and-conquer planner. It operates in three phases: plan generation, order dissemination, and plan monitoring. OPLANNER constructs tactical mission plans for confronting an enemy force. The second component of ALLIES, ANALYST, is a knowledge-based system for making predictions about enemy behavior. Using a data-driven approach, ANALYST processes sensor data about enemy forces to formulate hypotheses concerning the types of enemy units and their capabilities. In addition, ANALYST uses an a priori model of enemy forces to predict the location of undetected enemy units from detected units. Finally, the third component of ALLIES, BEM, creates simulated battlefield environments upon which OPLANNER and ANALYST can operate. BEM is an object-oriented simulator which models the actions of military units on a battlefield.

The three parts of ALLIES are loosely connected to one another to allow studies in command and control to be conducted. ANALYST and OPLANNER operate on LISP machines, while BEM is hosted on a VAX 11/780. The machines are connected via Ethernet. ANALYST develops enemy situation reports based upon simulated sensor data from BEM. As OPLANNER gets mission orders, it requests specific information from ANALYST. OPLANNER uses this information to create a tactical mission plan and then disseminate orders to friendly units simulated by BEM. Next, the friendly units in BEM execute the orders. OPLANNER monitors the plan execution via situation reports provided by BEM. "ALLIES has demonstrated that AI systems need to cooperate to help solve complex problems such as managing an airland battle and that much research is still needed" [Tachmindji & Bonasso, 1984, p. 184].

2.3.2 Mission Planning Using the Blackboard Model

A blackboard approach has been taken by [Pearson, 1988] for mission planning. This planning system creates a sequence of actions to be executed by an autonomous vehicle. The basic technique involves decomposing mission statements into constraints and subgoals, which in turn generate additional constraints and subgoals or actions. The knowledge sources possess expert knowledge in such areas as terrain features, physical characteristics of the vehicle, military doctrine, and planning strategies. The knowledge sources cooperate as in the standard blackboard paradigm to perform mission planning. This system is implemented on a Symbolics LISP machine using the Flavors package.

The mission planning for this system has been reduced to the problem of finding the best location to perform tasks. For example, one mission of the autonomous vehicle is to conduct reconnaissance. This involves finding locations where the target can be seen, yet the vehicle remains undetected by the target. When solving this problem, the constraints and subgoals are posted to the blackboard. The knowledge sources then interact via the blackboard to construct a solution that satisfies the constraints and subgoals. This system has shown that a class of mission-planning problems (finding locations appropriate for tasks) can be solved using a blackboard architecture. This system does have two problems, however. First, the blackboard consumes lots of memory — it does not discard any information posted to it. Secondly, this system operates on a single CPU and the knowledge sources are CPU intensive. This is a serious drawback for real-time dynamic planning problems.

2.3.3 Multi-agent Adversarial Planning

The MITRE Corporation has conducted research in Multi-agent Adversarial Planning (MAP) [Applegate et al., 1990; Benoit et al., 1990]. The general scheme of MAP involves a single, centralized agent creating plans and assigning tasks to other agents known as Action Managers (AMs). In creating such plans, the planning agent must consider the adversary (enemy force) which will be intent on thwarting its plans.

MAP uses preemptive counterplanning to deal with an adversary. MAP's reasoning alternates from an action-based approach ("What actions can get me to the goal?") to an obstruction-based approach ("How can the accomplishment of this goal be made more difficult, considering my adversary's abilities?"). In MAP, the centralized planning agent first constructs an offensive plan for the AMs. Next, the planner switches sides, so to speak. It assume the role of the adversary and develops hypothetical planned actions by the adversary which could interfere with the offensive plan. Then, the planner determines how to modify the original plan so as to nullify the adversary's interference. This process is similar to mini-max used in game playing in AI. Heuristics are used to reduce the combinational explosion of possible counterplans. If the planning process reaches quiescence, then a plan which the enemy cannot counter has been found. Heuristics may also be used to stop the planning process if quiescence is not achieved. The MAP process has been defined but not yet fully implemented on a computer. MITRE intends to completely implement the ideas of MAP to test its feasibility.

2.3.4 AI for Tactical Decision Support

The Command and Control Testbed Using Simulation (CACTUS) and the Situation-based Autonomous Reasoner in a GBB Environment (SARGE) were developed to facilitate research in applying AI techniques to tactical decision support systems [Lehnert & Sullivan, 1989]. CACTUS is an object-oriented land combat simulation between Blue and Red forces. It allows for evaluating battlefield command and control technologies. CACTUS simulates command, communications, combat, movement, and visibility between units. Using a blackboard architecture, SARGE directs the actions of individual military units in CACTUS.

CACTUS provides information to SARGE concerning the status of forces. Using this information, and being provided a goal (mission), SARGE constructs a plan and sends it to CACTUS for action by simulated units. SARGE monitors plan execution via data posted by CACTUS to the blackboard. If necessary, SARGE will replan portions of a mission depending on the actions of enemy units. SARGE uses a generic command model based upon a study of command and control literature. Part of the research with CACTUS and SARGE is to show that this generic command module is suitable for all levels of the command hierarchy (from small units to large units).

Both CACTUS and SARGE are implemented on a single Texas Instruments Explorer machine using Common LISP. Future work will involve the use of multiple processors to allow multiple planning systems to interact with CACTUS. In addition, SARGE will be modified to anticipate enemy actions to improve its planning process.

CHAPTER 3

DAI METHODOLOGY FOR THE AIR WAR ENVIRONMENT

3.1 Overview

The majority of military AI research devoted to assisting commanders has focused upon a somewhat centralized approach. These have included systems using blackboard architectures with a central, shared memory component and multi-agent systems with a central agent planning and assigning tasks for others. The distributed aspect of the air war environment lends itself to a distributed problem solving system. Distributed problem solving differs from multi-agent planning in that agents interact with one another and cooperate for the distribution and accomplishment of a given set of tasks. In other words, no one agent assigns tasks to others. The agents as a group determine the task distribution.

3.2 Background

In all but the simplest of cases, a problem may be looked upon as a set of tasks to be completed.

$$(T_1, T_2, T_3, ..., T_n)$$

A single problem solver can then develop an over-all solution by solving each individual task. The tasks may have certain constraints in regard to time, location, resource type required, etc. This fact may negate the approach of simply solving each task one at a time, in any convenient order. Accordingly, some tasks may require a temporal ordering, e.g., tasks T_3 and T_5 must be completed within the same time frame. Other tasks may have a constraint based upon a relationship with other tasks. For example, T_5 cannot be solved until T_4 is solved because a solution to T_5 depends upon the solution to T_4. Furthermore, a single problem solver may itself be constrained, in that it does not possess the required resources to solve all tasks while also satisfying the constraints.

In order to facilitate the solving of multiple tasks, multiple problem solving agents are to be employed,

$$(A_1, A_2, A_3, ..., A_m)$$

Each agent has at its disposal, a limited number of resources to use in completing tasks,

$$A_1\text{'s resources} = (R_{11}, R_{12}, R_{13}, ..., R_{1n})$$

$$A_2\text{'s resources} = (R_{21}, R_{22}, R_{23}, ..., R_{2o})$$

$$A_3\text{'s resources} = (R_{31}, R_{32}, R_{33}, ..., R_{3p})$$

$$\bullet$$

$$\bullet$$

$$\bullet$$

$$A_m\text{'s resources} = (R_{m1}, R_{m2}, R_{m3}, ..., R_{mq})$$

These resources may also be constrained based upon their functionality and availability. For example, some resources may be used only for specific tasks, while other, more general resources, can be used for any task. In addition, certain resources may be unavailable at certain times (undergoing maintenance, currently accomplishing another task, etc.).

In the ideal situation, one would have unlimited agents and resources. Each task would map to a single agent. Each agent would have the capability and resources necessary for completing the task. Figure 3.1 illustrates this idea of one task per agent.

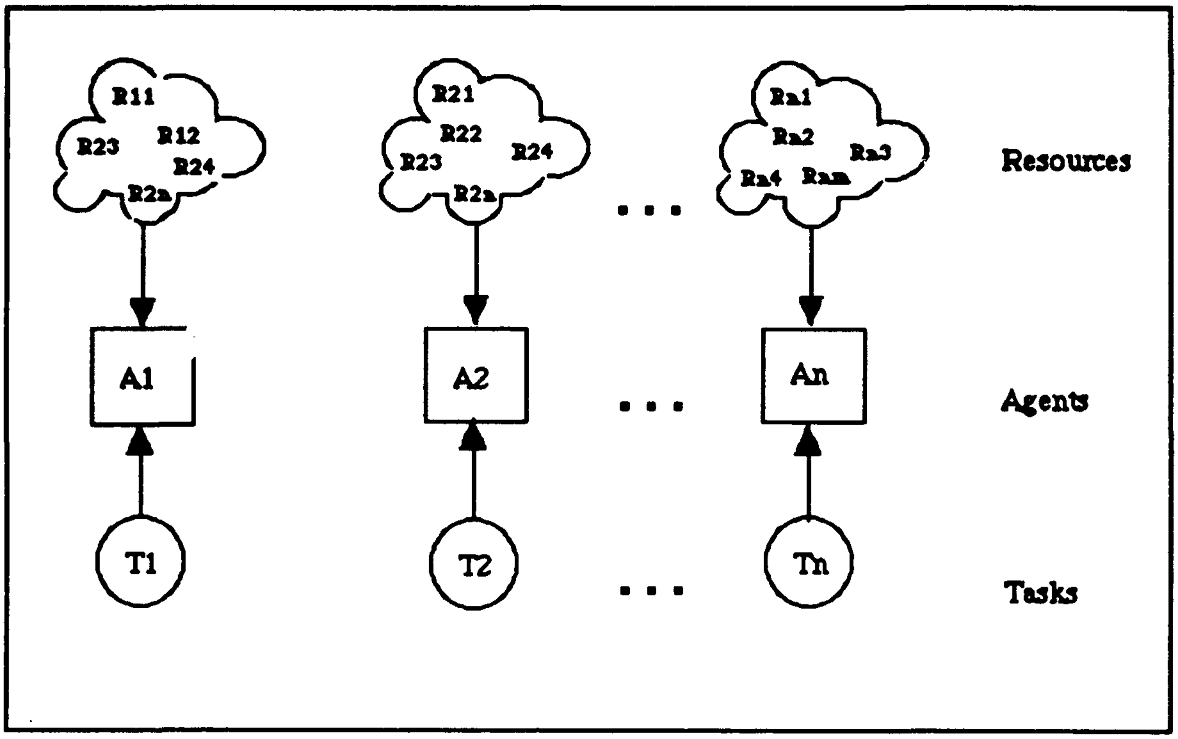

Figure 3.1. One-to-one mapping of tasks to agents.

In most cases, the ideal situation and the real world do not coincide. There are more tasks than agents. Moreover, individual agents may not possess enough resources or of the necessary types to complete a task. In this situation, agents must share resources with one another in order to accomplish the totality of tasks. The next figure represents this type of situation. Here there is one more task than the number of agents. Furthermore, the number and types of resources "owned" by each agent is different. Thus, agent A_2 is in charge of 2 tasks, and agent A_n must "borrow" resources which it does not have.

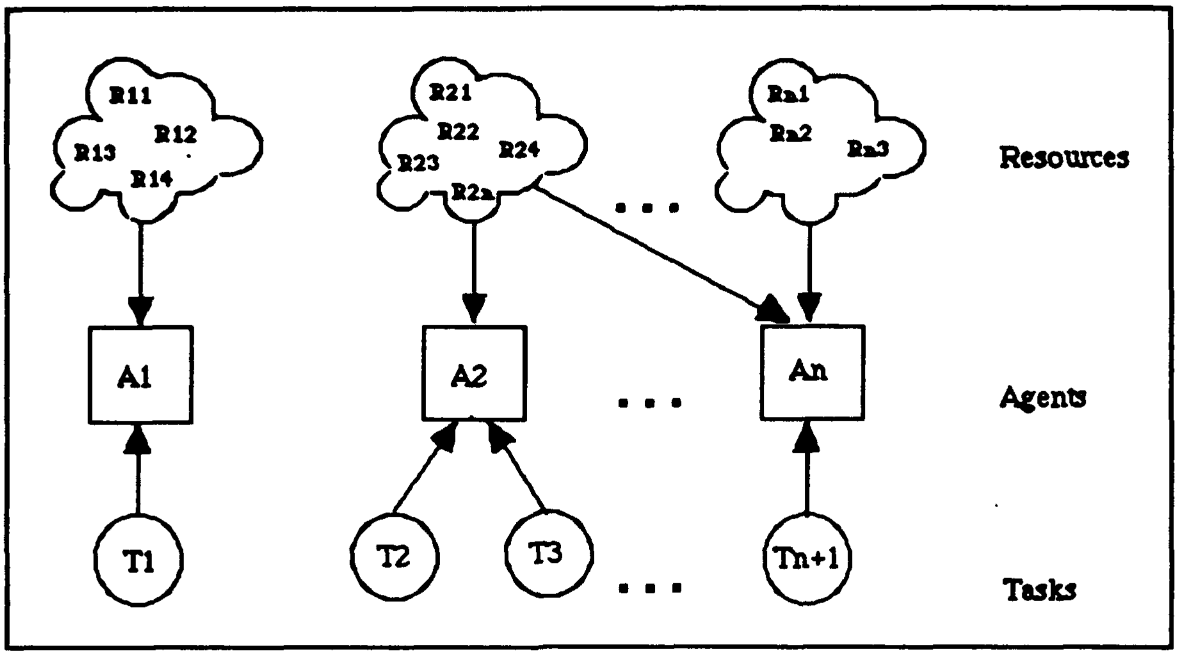

Figure 3.2. Multiple tasks and resource sharing.

The sharing of tasks and resources amongst a number of problem solving agents can provide a number of benefits:

- More tasks can be solved in a given time period using multiple agents than by using individual, isolated agents. A corollary of this is: generally, a collection of tasks can be solved faster with multiple agents than with individual, isolated agents.

- Graceful degradation can be achieved using multiple agents; i.e., if one agent fails, the entire problem solving system does not stop. It continues to operate, albeit in a degraded (more slowly, at a reduced level of quality) mode. In a single agent system, the failure of the one agent stops the entire problem solving process by that agent.

- With multiple agents, task solutions may be more flexible and robust than single agent solutions.

- Having agents specialized in certain problem sub-domains, the quality of the overall solution should be better.

The sharing of tasks and resources is not a simple matter nor trouble-free. Agents must somehow distribute the tasks amongst themselves and determine which agents will be responsible for which tasks at what point of time. As discussed in Chapter 2, various approaches to task distribution have been suggested and implemented. In some cases, a controlling or master agent may simply assign tasks to other agents. Other methods involve some type of negotiation between the agents to determine an equitable assignment of tasks.

Once tasks have been distributed and the agents begin solving their particular tasks, conflicts may arise with respect to resource sharing. These conflicts can occur since each agent has a limited number of resources, and some resources have limited functionality. Conflicts will arise within a time window for task accomplishment. If the urgency of task accomplishment is high, and since concurrent usage of resources is not possible, conflicts develop. Two major types of conflicts can thus be present:

- A total of *n* resources are required to complete all tasks; however, less than *n* resources exist between all agents.

- Two agents require the use of a particular type of resource but only one resource of this kind is available.

These conflicts must be resolved in order to achieve a satisfactory solution to the tasks at hand.

3.3 Agents in the Air War Environment

The agents comprising the distributed problem solving system in the air war environment have the following characteristics:

- Agents are fully-cooperative.
- Agents are distributed geographically throughout the region of conflict.
- Agents can freely communication with one another.
- Each agent has control over a specific set of resources (aircraft units).
- Agents are non-uniform (the type and number of resources controlled by each agent may differ).
- Agents operate at the wing commander level of authority in the Air Force chain of command.
- Each agent has a unique identifier (ID).

3.4 Distribution of Tasks

A form of negotiation is used amongst the agents to distribute tasks. Given a set of tasks by the air commander, each agent evaluates its ability and availability to accomplish each task in the set. This results in each agent providing a bid for each task based upon its estimated level of accomplishment. In other negotiation strategies, the agent with the best bid is assigned the respective task. However, this does not take into account the agents' ability to improve their bids subsequently, using the knowledge of other agents' best bids for each task. For example, suppose an agent, A1, has bids for tasks T1, T2, and T3.

Based upon its resources, A1 has low bids for T1 and T2, and a high bid for T3. (The lower the bid the better the prospect of being assigned the task.) Another agent, A2, also has bids for T1 and T2, while a third agent, A3, has a bid for T3. As determined by the bids, A1 has the best bid for T1, A2 the best for T2, and A3 the best for T3. However, if A1 had known that it would not have the best bid for T2, it could have provided a better bid for T3. This is because resources which it thought would be applied to T2, could now be applied to T3, and thus improve the quality of accomplishing T3 (cost, timing, success rate, etc.). This negotiation approach is used in this research to allow agents to modify bids.

3.4.1 Task Evaluation

Agents are most concerned with solving those tasks that are closest to them. Therefore, each agent first sorts the task list according to distance, i.e., the first task to consider will be the closest geographically, the second task to consider is the one next in distance, and so forth. Using this sorted list, each agent then evaluates its ability to solve the task. The following function is used for the evaluation:

$$QM_{ij} = C_1T_i + C_2P_i + C_3S_i + C_4W_j$$

where QM_{ij} is the quality measure of task i by agent j, T_i is the timeliness for completing the task i, P_i is the cost of primary resources to be used, S_i is the cost of support resources, and W_j is the current workload of the agent. C_1, C_2, C_3, and C_4 are weighting factors. Each term in this function is calculated as follows:

- Timeliness is calculated as $T = D/S$, where D is distance to the task and S is the speed of the slowest resource to be used for task accomplishment;

- The primary resource cost factor will be a value within the range 0-2. Aircraft resources have various capabilities. Some support specific types of missions and are best for accomplishing tasks involving those missions. Other aircraft have general, multi-purpose roles. If an agent has aircraft available which specifically support the mission associated with the task, then the cost value is 0. If the agent can only use general purpose aircraft for this task, then the cost value is 1. If the agent does not possess any aircraft for this mission (and will have to borrow from other agents), the cost is 2.

- The support resources factor is calculated in a similar manner to primary resources. A value in the range 0-2 is used for support cost. Support aircraft are those which support the primary mission of a task. For example, fighter aircraft have a support role (bomber escort) for strategic bombing missions.

- The workload factor is simply the number of tasks which are currently assigned to the agent.

3.4.2 Task Assignment

The basic technique for task assignment involves a negotiation strategy. Agents evaluate their ability to accomplish each task (calculate quality measures) and then exchange these with one another. Agents whose QMs meet a specific criterion as the "best QM", become self-appointed commanders for those respective tasks. Tasks for which no commanders are assigned are renegotiated. (This procedure is heuristic in nature. A different "best bid" could be offered later by another agent that no longer submits bids for all tasks that it bid before. However, considering such possibilities would make the whole task assignment process prohibitively long in complex, real-life situations.) Using knowledge about which tasks were assigned, agents will recalculate QMs for the remaining tasks in order to improve their "bids". This process repeats until all tasks have commanders. Since the agents do not simply use the best initial bids to assign all tasks, a better distribution of tasks amongst the agents should result. The following describes the task assignment process in detail.

Each agent evaluates its ability to accomplish each task as stated in Section 3.4.1. Since the task list is sorted by distance, the agents give primary consideration to those tasks that are closest to them. Therefore, a task which is farthest away may have a large QM due to the distance involved and the fact that resources are being considered for use with closer tasks.

After the agents have completed the evaluation of tasks, they exchange the task quality measures. This is done by sending a message to all other agents. The message is packaged in a simple data structure which lists each task's ID and the QM for each task. Having each agent's QM for each task, all agents can now determine which are the best "bids." (Best bids being the smallest QMs.) An agent becomes a self-appointed commander for a task if the following hold true:

1. It has the smallest QM for the task.

2. Its QM exceeds a certain *threshold*. Specifically, the best QM must exceed the next best QM by a certain value. This value is referred to as the threshold value. The idea here is that QMs which are relatively close to one another in value may not discriminate which is definitely *the very* best because of possible shortcomings of the evaluation function and in the system's assessment of the environment.

Agents notify all other agents when they become self-appointed commanders. Some tasks may not initially have commanders because the best QM did not exceed the threshold requirement. In this case, all agents will re-evaluate their QM's for those tasks, using the knowledge of which tasks have already been assigned to agents and which of its own resources have already been committed to a task. An agent may now be able to give a better QM to a task because it "lost" a bid to another agent and this made additional resources available for another task. This process of re-evaluating tasks continues until all tasks have a commander using the two criteria listed above, or a level of quiescence is reached. Quiescence is reached if the agents can no longer improve their QMs for the remaining tasks. In this case, the task commanders will be those agents with the smallest QMs. If two or more agents have the smallest QM for a task, then the agent with the least workload becomes the commander. Finally, if two or more agents have the smallest QM and the same workload, then the commander becomes the agent with the smallest ID. (Each agent has a unique ID value. This ID value is the final tie-breaker to use in assigning commanders, if all other factors are equal.)

At this point, all tasks will have a commander. All agents will then set about solving their particular tasks. This will involve allocating specific resources and determining the best path for the resources to use in flying towards the target associated with a task.

Consider the following scenario involving two agents, A1 and A2. We also have a set of available tasks: (T1, T2, T3, T4) and a threshold value of 2. Exemplary initial QMs calculated by each agent are shown in Table 3.1.

Table 3.1 Initial Quality Measures

	Agent 1	Agent 2
Task #1	8.0	11.0
Task #2	7.1	6.0
Task #3	9.0	10.2
Task #4	6.3	7.1

If tasks were assigned simply based upon the smallest QM, then A1 would be assigned T1, T3, and T4, while A2 would be assigned T2. However, using the technique outlined in this chapter, A1 is assigned T1 since it has the smallest QM (8.0) and its QM exceeds A2's QM (11.0) by more than the threshold value. The other tasks are not assigned since none of the agent's QMs for these tasks meet the criteria for task assignment. Therefore, T2, T3, and T4 are renegotiated. Using the knowledge that it lost the bid for T1, A2 can improve its bids for the remaining tasks. (Resources that would have been used for T1 can now be used with the other tasks instead and, in turn, A1 can no longer count on those resources that are expected to be lost in performing T1.) After a second iteration, the agents compute the following QMs shown in Table 3.2.

Table 3.2 Quality Measures After a Second Iteration

	Agent 1	Agent 2
Task #2	7.2	4.1
Task #3	9.0	9.0
Task #4	6.0	3.0

A2 is now assigned T2 and T4. T3 must be renegotiated. Assume that after the third iteration, the agents achieved quiescence, i.e., they could not improve their bids. In this case, A1 would be assigned T3 since it has the same QM as A2 but the smallest workload (1 task versus 2 tasks). The final assignment of tasks is now A1 with T1 and T3, and A2 with T2 and T4.

3.5 Tasks in the Air War Environment

The tasks in the air war environment relate to specific air missions to be flown. The type of mission and characteristics of the air war environment dictate the types and numbers of air resources needed. The solution to a particular task, then, involves allocating air resources and computing a flight path. The following attributes are associated with each task:

- A unique identifier (ID).
- A description of the task (air mission, location, and target).
- A statically assigned importance value, as determined by the air commander.

- An urgency value related to time. The sooner a task must be accomplished, the more urgent the task.
- A deadline. This is the time by which the task must be accomplished; otherwise, the task accomplishment is no longer relevant.

3.6 Hierarchical Iterative Conflict Resolution

To resolve conflicts in such a distributed problem solving system, we can take an iterative approach. To use this technique, each task must have a priority level. Thus, tasks with a higher priority should be solved before tasks with a lower priority. This technique also eliminates the time expense associated with a contract net for resolving conflicts. Agents attempt to solve the specific tasks that are assigned to them. Conflicts are resolved based upon task priority. If the agent with the highest priority task has a resource conflict, that agent may "borrow" resources from any other agent to eliminate the conflict. This, in turn, could create a conflict with a lower-level priority task. If this is the case, the conflict resolution favors the agent with the task that has the next highest priority. In essence, an agent which resolves a conflict is making the following statement to the agent from which a resource is being borrowed:

"If my taking this resource affects your solving your task, then you must replan your solution (re-allocate resources)."

With this approach, conflicts are resolved iteratively based upon priority. Since some tasks are more important than others, the agents responsible for the higher priority tasks have the right to borrow and allocate resources owned by other agents. Using this scheme, either all conflicts will be resolved, or lower priority tasks will remain unsolved due to resource constraints. The general high-level algorithm for iterative conflict resolution is shown below:

Distribute tasks to agents

Each agent constructs a solution plan for its task

WHILE tasks not marked **DO**

Agent with next highest priority task resolves its conflicts

Mark this task solved or unsolvable

Notify other agents

OD

While an agent is resolving resource conflicts, the other agents should not be idle. Otherwise, this would diminish the benefit of distributed agents. Instead, all agents should be generating alternative, tentative solutions for their specific tasks. If an agent with a higher priority task must allocate a resource which affects others' solution, a list of alternatives will then be available for the affected agents. This will reduce the time needed for the latter to eliminate problems which were introduced by having to lend some of its resources.

As mentioned earlier, the tasks to solve involve allocating air units to attack, intercept, and destroy enemy resources (aircraft and land-based targets). The problem solving agents exist at the wing commander level in the Air Force command hierarchy and are located at the airfields. Each agent has a limited number of aircraft at its disposal.

Furthermore, each type of aircraft squadron has a certain functionality and may be used only for specific types of tasks.

To solve a task, an agent must allocate aircraft squadrons with the desired capability. For destroying land-based targets, bomber aircraft must be employed. For suppressing enemy air defense systems, Wild Weasel aircraft are needed. For intercepting enemy aircraft, fighters must be used. For long range missions, tanker aircraft are necessary. Of course, combinations of the above may be needed for complex missions. As an air war progresses, resources will be lost to combat. Also, at any given time, some resources will be unavailable due to maintenance activities. Therefore, individual agents will, at times, lack the resources needed to solve a given task. This in turn leads to resource sharing between agents, which can cause resource conflicts.

The priority associated with a task is calculated based upon two factors — importance (I) and urgency (U). The importance value of a task indicates the task's relative worth with respect to other tasks being considered. It is dependent upon the type of task, conditions of the environment and, in any domain, the relative and absolute value of the task to all cooperating or competing agents. This static importance value is provided when available tasks are announced to agents. The second component of task priority is urgency. This component is dependent upon time, such that the closer the current time is to the deadline for task accomplishment, the more urgent is the task. Thus, urgency is inversely proportional to the difference between the deadline and current time. The smaller the available time, the greater the urgency. In this manner, tasks are not simply prioritized along one dimension (importance or time). Instead, priority becomes a function of the two components:

$$\text{Priority}(T_i) = \text{Importance}(T_i) * \text{Urgency}(T_i)$$

where T_i represents the i-th task. In fact, the importance value associated with a task may also be subject to change. This can happen if the environment changes or more/better information becomes available. For example, in a military environment a commander may reassess the importance of a task as more information becomes known. An enemy bunker targeted for attack may have a high initial importance value. However, if later intelligence reports indicate that no enemy troops are deployed at the bunker, its importance value may be lowered.

The use of a method to prioritize tasks based upon some importance value and time (urgency) has been used in other domains. The scheduling of jobs for access to one limited resource (the CPU) in Operating Systems of time-shared computing is one such example [Peterson & Silberschaty, 1985]. A job's priority may be computed based upon the size of the job (a static value) and the amount of time the job has spent waiting for the CPU. Initially, a large job may have a low priority and have to wait for access to the CPU due to the presence of shorter jobs. However, the large job's priority will increase over time, such that it will finally obtain access to the CPU even if shorter jobs are present. However, there is only one type of task and one type of resource in this domain. Real-time database transaction processing is another example where priorities have been used to schedule processes for access to a single limited resource — a database record [Graham, 1992; Huang, et al., 1989; Stankovic & Zhao, 1988]. With a real-time database, transactions must be processed within a certain time limit (deadline). Additionally, in a study by [Huang et al., 1989], transactions were also given "criticalness" values. Priorities for transactions to be processed were assigned based upon a combination of criticalness values and time constraints.

Applying the general Hierarchical Iterative Conflict Resolution algorithm to the air war environment, results in the following revised algorithm:

1. The air commander distributes the problem to the agents. The problem consists of a list of tasks with respective attribute values. Each ground-based target will have a statically assigned importance value associated with it. (Thus, the importance value of a task will be the importance value of the target associated with the task.) Under some conditions, certain tasks could have an extremely high urgency value. Furthermore, it is possible for several tasks to have the same priority. In these cases, preference will be given to the task with the smallest deadline value.

2. Each agent evaluates its ability to accomplish each task as discussed in Section 3.4.1.

3. Agents exchange their quality measures for each task.

4. Agents become task commanders (assigned tasks) as described in Section 3.4.2.

   When all tasks have a commander, the agents exchange information regarding their resources. This information consists of the status of resources (maintenance, available, on a mission, or returning to base). With task commanders identified and resource information exchanged, agents can construct detailed solutions (plans) for their tasks.

5. At this point, each agent begins solving (planning for) its particular task. The agents attempt to find the best solution possible using their own resources. When a solution is constructed, the respective agent notifies all other agents that it has a tentative solution for its task.

6. If conflicts arise, they will be resolved according to task priority.

    a. The task commander of the highest priority task may take resources from other agents, if needed. This may create conflicts in other agents' plans for tasks which are of lower priority.

    b. If so, such agents will have to replan for the affected tasks. Conflicts may arise and be resolved in lower and lower levels in a similar manner, in this iterative plan refinement process.

    c. This process is repeated in task priority order for all tasks.

    d. Agents with lower priority tasks do not sit idle, waiting for higher priority tasks to be solved. Instead, they will be constructing alternative, tentative solutions for their tasks. When notified that plans for higher priority tasks have been completed, they may then finalize their solution from their list of alternatives.

7. Either all conflicts can be resolved and all tasks taken care of or, if there are not enough resources, some lower priority tasks will remain unaccomplished.

As can be seen by the description of this process, it is hierarchical since we are planning for tasks based upon a hierarchy of priorities. All agents may not be able to solve all given tasks; however, this method ensures that the more important tasks will be solved (planned for) before tasks of lesser importance. The following is a pseudo code description of this conflict resolution process.

```
Send prioritized task list to each agent

WHILE all tasks not assigned a TASK-COMMANDER DO

    Each agent recomputes quality measure (QM) of its ability to accomplish
    remaining tasks

    Agents broadcast their QMs to all other agents

    FOR TASK = Task-1  TO Task-n DO

        IF quiescence has not been reached THEN

            TASK-COMMANDER[TASK] = Agent with best QM
            AND this QM >= threshold value

            TASK-COMMANDER[TASK] notifies other agents of
            self-appointment as task commander

        ELSIF quiescence reached THEN

            TASK-COMMANDER[TASK] = Agent with best QM
            (smallest workload or ID value for tie-breaker)

            TASK-COMMANDER[TASK] notifies other agents of
            self-appointment as task commander

        FI

    OD

OD
```

```
Agents exchange resource lists

Agents construct tentative solutions (plans) for their tasks

TOP-PRIORITY = priority of task with highest priority

WHILE all tasks not marked DO

    IF conflicts present AND task priority < TOP-PRIORITY THEN

        Generate alternative solutions

    ELSIF conflicts present AND task priority = TOP-PRIORITY THEN

        IF other agents with lower priority tasks have resources

        which will resolve conflicts THEN

            Borrow resources to resolve conflict from agents with least
            priority tasks

            Mark task plans as completed

            Notify other agents of the plan

        ELSE

            Mark task unsolvable

        FI

        TOP-PRIORITY = priority of task with next highest priority

    FI
```

In resolving conflicts, certain heuristics will be adhered to in order to minimize the impact of borrowing resources from other agents:

- Resources with specific and limited functionality will be allocated first.
- When conflicts occur, available resources will be borrowed from agents which have "solved" their tasks, if possible
- When conflicts occur, resources will be borrowed from agents having the task with the smallest priority, if possible.

The next three figures illustrate the process of Hierarchical Iterative Conflict Resolution. This scenario involves three tasks (T1, T2, and T3) and three agents (A1, A2, and A3). Each agent has a specific set of aircraft squadron resources. These are specified with a type designator, followed by an identifier. For example, B-1 is bomber squadron #1. The other type designators are:

F - Fighter FB - Fighter-Bomber W - Wild Weasel

Figure 3.3 shows the assignment of tasks, the priority of the tasks, and the current tentative solutions (plans). In this case, A1 has allocated resources B-1 and F-1 for its task. A2 has allocated B-2, F-2, and W-1. A3, on the other hand, has allocated F-3 but is lacking another fighter resource which is needed to accomplish its task. This is designated in the figure by substituting a question mark for the resource ID, i.e., F-?.

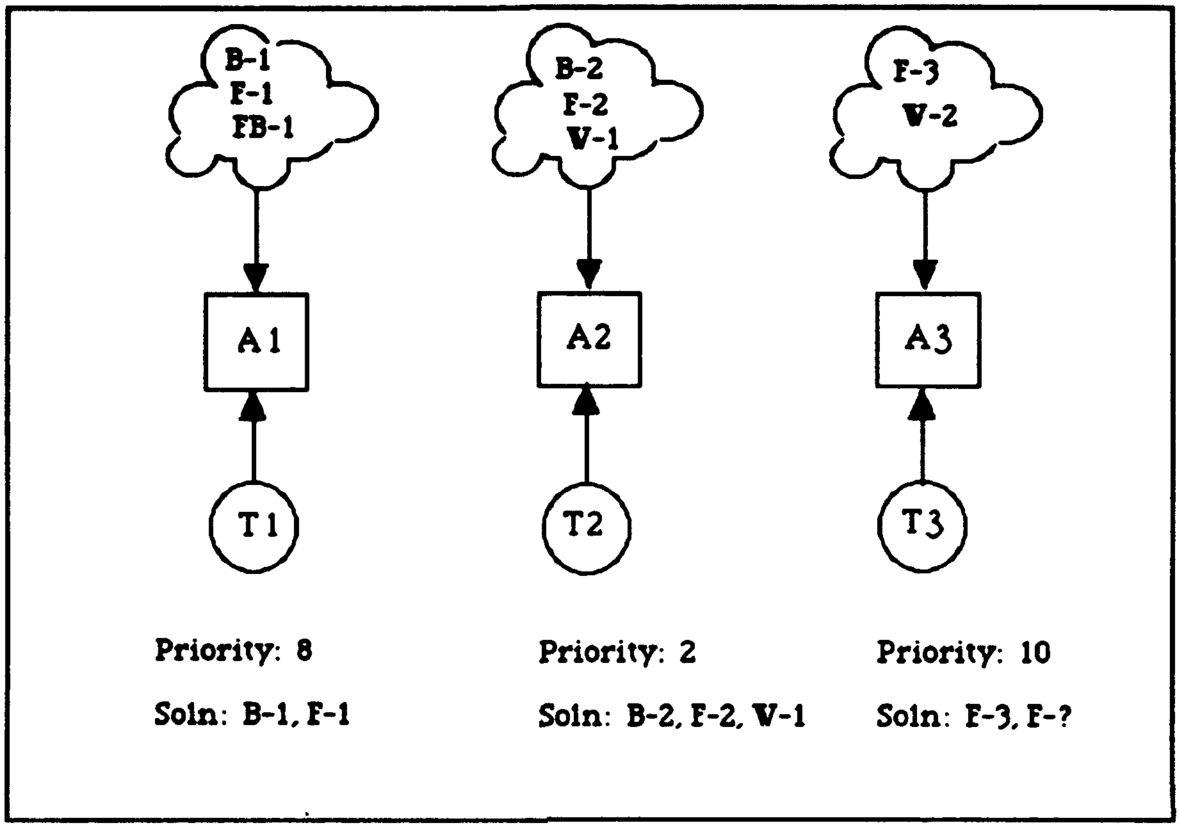

Figure 3.3. Resource conflict with A3 after first iteration.

After the second iteration of the conflict resolution technique, A3 has borrowed resource F-2 from A2. This is shown in the Figure 3.4 by the dashed arrow indicating that F-2 is being allocated by A3. Notice that A3 has the task with the top priority (10), and that it has allocated a resource from the agent with the task having the least priority (2). Also, notice that A3's allocation of F-2 has now caused a resource conflict in A2's solution. A2 must now attempt to allocate another resource in order to solve its task.

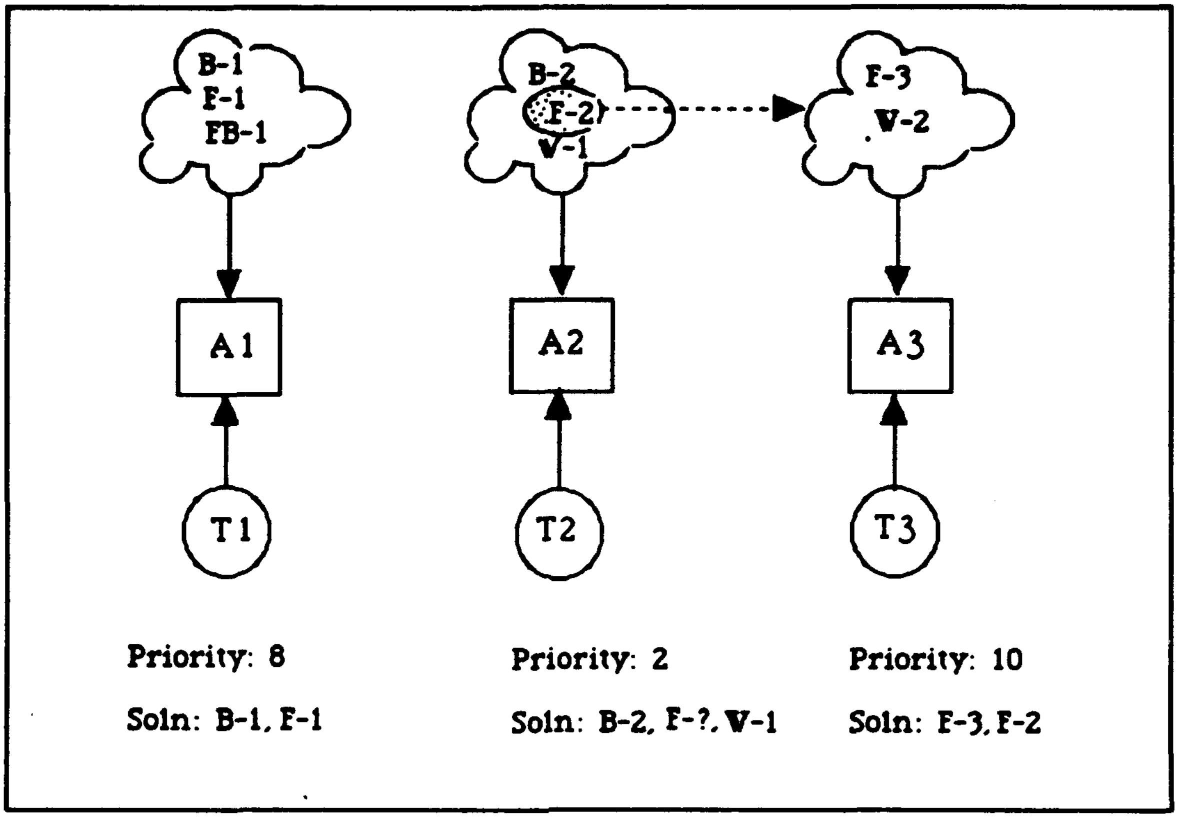

Figure 3.4. Resource conflict with A2 after second iteration.

Finally, Figure 3.5 shows that after the third iteration of the conflict resolution technique, all tasks have been solved. A2 resolved its conflict by allocating resource FB-1 from A1. In this example, T2 could use either a fighter or fighter-bomber. T3 was constrained by the fact that only fighters could be used in its solution.

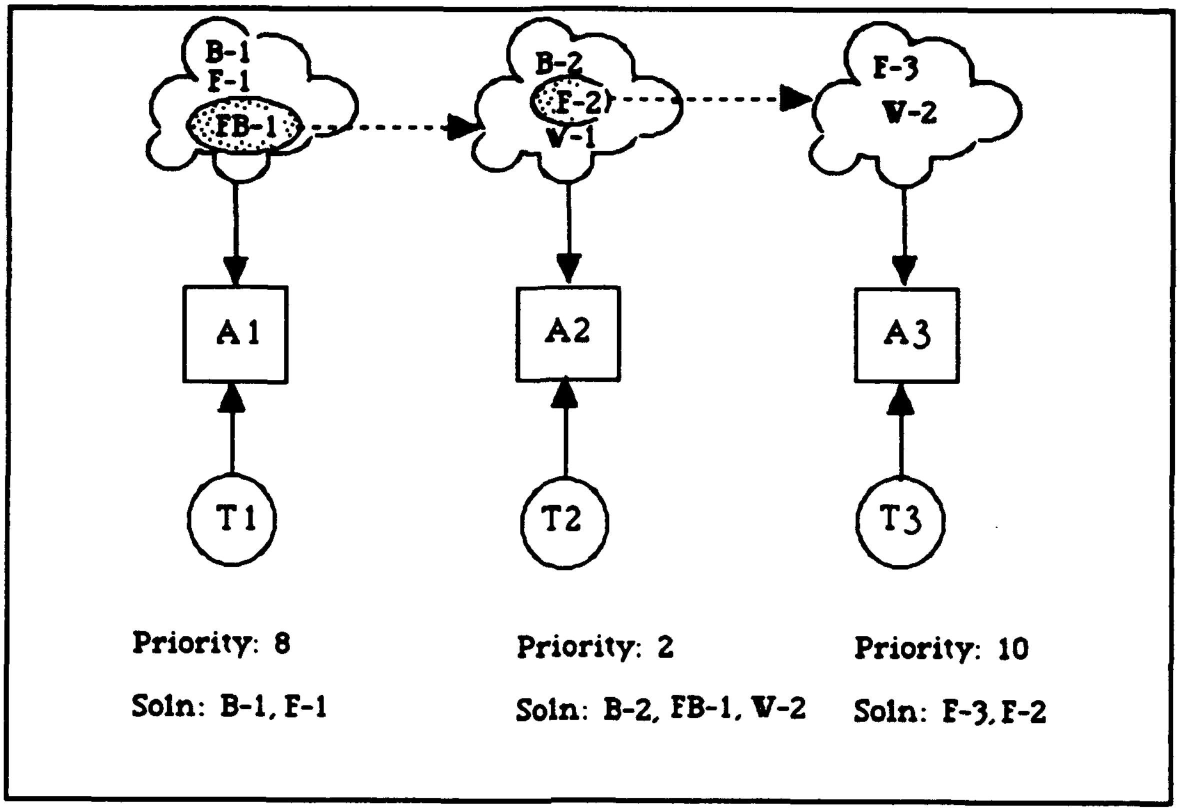

Figure 3.5. All tasks solved after third iteration.

These innovative techniques for task distribution and agent cooperation-coordination have been implemented in the distributed air war problem solver testbed. The specific implementation details are described in the following chapter.

CHAPTER 4

ARCHITECTURE, DESIGN AND IMPLEMENTATION OF THE TESTBED

4.1 Architectural Overview

The Distributed Air War (DAW) testbed is a flexible system for conducting studies related to the use of distributed intelligent agents for solving resource allocation problems within an air war environment. As shown in Figure 4.1, the testbed consists of three main modules: the Battle Simulation Module (BSM), the User Interface Module (UIM), and the Distributed Problem Solving Module (DPSM). The BSM simulates key aspects of an air war. The UIM presents a graphical representation of the simulated air war environment to

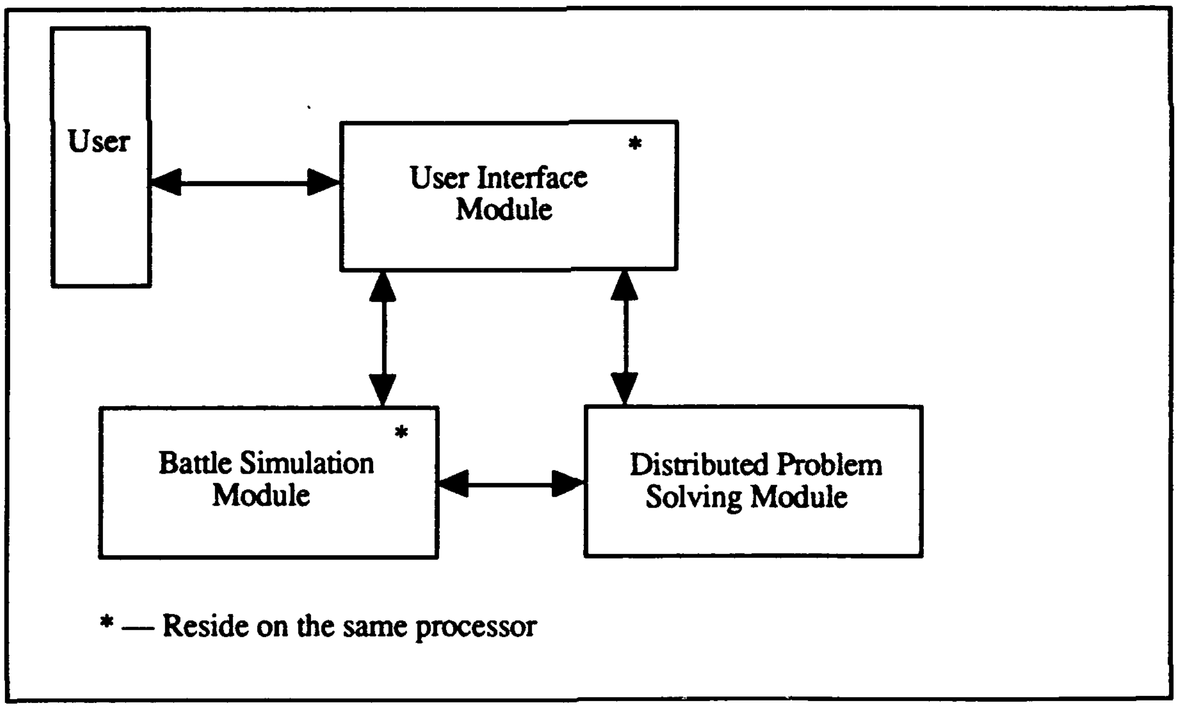

Figure 4.1. Architectural view of the testbed.

the user and allows the user to easily interact with the testbed. Finally, the DPSM provides a system of distributed agents to solve problems presented in the testbed. The DAW testbed was developed on a Sun workstation with the UNIX operating system. When running the testbed, the BSM and UIM reside on one workstation while the DPSM resides on a series of networked SUNs. The BSM is written in the C programming language. The UIM uses the X-Windows library, and the Motif toolkit for providing a mouse-driven, graphical interface. Lastly, the DPSM uses CLIPS (C Language Integrated Production System), an expert system development tool.

The testbed was constructed with modularity and flexibility in mind. As such, the three main modules can operate independently of one another and allow new modules to be integrated with the testbed if necessary. C was chosen as the primary programming language due to its efficiency and portability across UNIX-based platforms. Similarly, X-Windows and Motif were chosen for development of the user interface, as they represent current graphical standards for UNIX workstations. Finally, CLIPS was selected for the construction of the problem solving module because of its functionality and, more importantly, the ease with which it can be interfaced with standard procedural languages, such as C. In addition, CLIPS operates on a number of systems (UNIX workstations, MS-DOS computers, and Macintosh computers). This provided a degree of flexibility during development of the testbed.

4.2 The Battle Simulation Module

The method of simulating air wars in the testbed is based upon the manner in which air war simulations are conducted at the United States Air Force Academy for their Air Power Doctrine and Strategy course [United States Air Force Academy, 1990]. (The several parametric values that appear in this dissertation for computing the effects of combat have also been adapted from this reference.) The BSM simulates the salient features of an air war. These features consist of the following:

- a map representing an area of conflict in which locations can be identified using some type of coordinate system;
- two forces which oppose one another in an air war;
- various types of fixed, land-based resources which may be viewed as potential targets by each force (command centers, airfields, munition depots, power centers, transportation centers, etc.);
- various types of air resources, with differing capabilities, normally found in an air war environment (fighter aircraft, bombers, etc.);
- phases of aircraft operations to include maintenance, take-off, movement between waypoints (map coordinates or other reference points), landing, and combat;
- the fog and friction (unknown and unexpected events) associated with warfare; and
- ground-based air defense systems.

4.2.1 Map and Coordinate System

The BSM employs a map covered by hexagonal elements, as is used with most military simulations. A coordinate system consisting of a combination of letters and digits is used to identify locations on the map. Horizontal map positions are indicated using letters ranging from A to KK. (Coordinates after A are referenced with two letters. For example, the coordinates after Z are AA, BB, CC, and so forth.) Vertical map positions are numbered from 1 to 51. Thus, any of the 1887 hex locations on the map may be specified using a letter-number combination, such as A1 or C27. With such a coordinate system, aircraft flight plans may be listed in the traditional manner as a sequence of waypoints, i.e., a series of map positions an aircraft should follow.

4.2.2 Land-Based Resources

The testbed provides twelve different kinds of land-based resources as shown in Table 4.1. Each land-based resource has a hardness value associated with it. This value indicates the relative strength the resource has against air-to-ground attacks. For example, an ammunition depot has a hardness of 600 while a bridge has a hardness of 100. This means that the ammunition depot is more difficult to destroy, and hence, more firepower is required to destroy an ammunition depot than a bridge.

The data structure shown in Figure 4.2 is used for maintaining information about a land-based resource (target). The *type* field indicates the type of target. The *sam* field contains the ID of the Surface-to-Air Missile (SAM) unit defending the target. If no SAM unit is located with the target, then this field will have a value of 0. The map location of the target is maintained in the *location* field. The *damage* field holds the current damage level sustained by the target. The damage level will be between 0% (no damage) to 100% (completely destroyed). Finally, the hardness value of the target is kept in the *hardness* field. As targets sustain damage, their hardness value decreases by a like amount. For example, a target which receives 35% damage has its hardness value reduced by 35%. This implies that less firepower is required to destroy the target as a whole when it has sustained some damage, as compared to when the target is fully operational.

Table 4.1 Land-Based Resources (Targets)

Type	Hardness
Airfield	1000
Ammo Depot	600
Command Center (HQ)	500
Industrial Area	400
Radar	300
City	300
Power Plant	300
POL Site	200
Rail Center	200
Dam	200
Staging Area	200
Bridge	100

```
typedef struct Target
{
        short       type;       /* Type of target */
        short       sam;        /* SAM unit defending target */
        long int    location;   /* Map coordinate */
        short       damage;     /* Current damage level */
        short       hardness;   /* Current hardness value */
} Target;
```

Figure 4.2. Target data structure.

4.2.3 Air Defense Resources

SAM units are used for air defense in the BSM. They are co-located with ground resources and are used to fire at airborne enemy aircraft located over the associated target. The data structure listed in Figure 4.3 maintains information about a SAM unit. Each SAM unit consists of a particular type of SAM, a number of SAM launchers, and a number of spare SAMs. As SAMs are fired during simulation of combat, the launchers are reloaded with the spares. Two types of SAMs are provided for in the testbed — S-01s and S-02s. Their effectiveness in shooting down enemy aircraft depend on the type of aircraft and the speed of the aircraft. Slower moving aircraft are more susceptible to SAMs than faster moving aircraft. In addition, SAM suppression aircraft, known as Wild Weasels, are immune to SAMs. Table 4.2 shows SAM effectiveness according to aircraft speed. The effectiveness is stated as a probability of shooting down a plane. As shown in the table, an S-01 has a 1 in 3 chance of downing an aircraft flying at a speed of 6 hexes/cycle (hpc), a 1 in 6 chance of shooting an aircraft with a speed of 8 hpc, and no chance of hitting an aircraft with a speed of 10 hpc.

```
typedef struct SAM_Unit
{
        short       type;              /* Type of SAM */
        short       num_launchers;     /* Number of launchers */
        short       num_spares;        /* Number of spare SAMs */
        long int    location;          /* Map coordinate */
} SAM_Unit;
```

Figure 4.3. SAM unit data structure.

Table 4.2 SAM Effectiveness According to Aircraft Speed

	Aircraft Speed (hexes/cycle)		
SAM	6	8	10
S-01	1 in 3	1 in 6	None
S-02	1 in 2	1 in 3	1 in 4

4.2.4 Aircraft Resources

The BSM supports eleven different types of aircraft resources. These aircraft are shown in Table 4.3. Each type of aircraft has two combat factors associated with it. Combat factors indicate the relative strength of an aircraft in combat. As listed in the table, an F-02 has a combat factor twice the value of an F-01 (10 versus 5). Thus, an F-02 is more likely to defeat an F-01 in combat than vice versa. An air-to-air (A-to-A) combat factor indicates combat strength for aerial engagements, while an air-to-ground (A-to-G) combat factor indicates combat strength for bombing engagements. Each type of aircraft also has a maximum endurance level. This is the maximum number of simulation cycles the aircraft can remain airborne before running out of fuel (without in-flight refueling). In addition, each kind of aircraft has a maximum speed and is limited to the number of times it can engage the enemy. After engaging the enemy in combat its maximum number of times, an aircraft must land to re-arm. For flexibility and robustness, data concerning aircraft characteristics is maintained in an external file. Thus, these features can be modified easily to accommodate specific experiments and to accommodate real data for operational use.

Table 4.3 Aircraft Resources

A/C	Combat Factor				Maximum Engagements	
Type	A-to-A	A-to-G	Endurance	Speed	A-to-A	A-to-G
F-01	5	0	3	8	3	0
F-02	10	0	4	8	3	0
F-03	15	0	5	10	3	0
F-10	3	5	3	8	2	1
F-20	6	8	4	8	2	1
F-30	10	12	5	10	2	1
B-40	0	15	3	10	0	2
B-50	0	30	4	6	0	2
B-60	0	40	5	8	0	2
W-11	0	10	5	8	0	2
K-12	0	0	10	6	0	0

The following summarizes the capabilities of aircraft resources:

- F-01s, F-02s, and F-03s are fighter aircraft. They may only engage in air-to-air combat. They are used for defending friendly resources and escorting other aircraft on missions.
- F-10s, F-20s, and F-30s represent fighter-bombers. These are multi-role aircraft and, as such, may engage in both air-to-air and air-to-ground combat.
- B-40s, B-50s, and B-60s are bombers. They may engage only in air-to-ground combat. They have no air-to-air capability, and thus, must be protected by other aircraft when on bombing missions.
- W-11s are Wild Weasels used for suppression of SAMs.
- K-12s are tankers. Tankers have no combat capability and are used to refuel other aircraft while in-flight.

Figure 4.4 shows the data structure which maintains information about an aircraft squadron. Each squadron is composed of one type of aircraft. Squadrons are under the control of specific agents indicated by the *owner* field. Furthermore, squadrons may have different types of crews and weapons assigned to them which will impact the squadron's capabilities. Three kinds of crews are available: novice, standard, and experienced. Novice crews decrease the combat factor of a squadron by 10%. Standard crews have no effect on combat factors, while experienced crews increase the combat factor by 10%. Similarly, three types of weapon systems are available: dumb, standard, and smart. Dumb weapons decrease a squadron's combat factor by 15%. Standard weapons have no impact, and smart weapons increase the combat factor by 15%. The *refuel_stat* field is applicable to tanker squadrons only. This field relates to the amount of fuel the tanker has available to provide to other aircraft squadrons during in-flight refuelings. The last field in the data structure, *status*, indicates the current status of a squadron. The status of a squadron will be one of the following:

- Dead — All aircraft in the squadron have been destroyed.
- Maintenance — The squadron is undergoing maintenance. Whenever squadrons land, they enter a period of maintenance for refueling/re-arming and are unavailable for air missions during that time.
- Available — The squadron is on the ground, available for a mission.
- Mission — The squadron is airborne, accomplishing an air mission.
- RTB — The squadron is airborne and has completed its mission. It is returning to base (RTB).

```
typedef struct Squadron
{
        short       owner;        /* Which agent owns this resource */
        short       type;         /* Type of aircraft */
        short       number;       /* Amount of A/C in this squadron */
        long int    location;     /* Current map location */
        short       crew;         /* Crew skill level */
        short       weapon;       /* Weapon level */
        short       speed;        /* Current air speed */
        short       a_a_stat;     /* A-A engagements left */
        short       a_g_stat;     /* A-G engagements left */
        short       fuel;         /* Fuel status */
        short       refuel_stat;  /* For tankers only */
        short       engage;       /* Engaged with enemy? */
        short       status;       /* Squadron status */
} Squadron;
```

Figure 4.4. Aircraft squadron data structure.

4.2.5 Simulating Events

To simulate air wars, resource data files must first be established for the blue force and the red force. (Refer to Section 4.3.) Each cycle of simulation represents a period of time in which events occur in the air war environment. Data concerning aircraft movement for each cycle is provided by the user, via the UIM, and the DPSM. (Normally, a user will have control over one force and the DPSM will have control over the other force.) Aircraft movement data is created by both the user and DPSM when planning for tasks. Figure 4.5 is a simplified architectural view of the BSM. Figure 4.6 shows a data flow diagram representing the flow of data between the main operations during simulation.

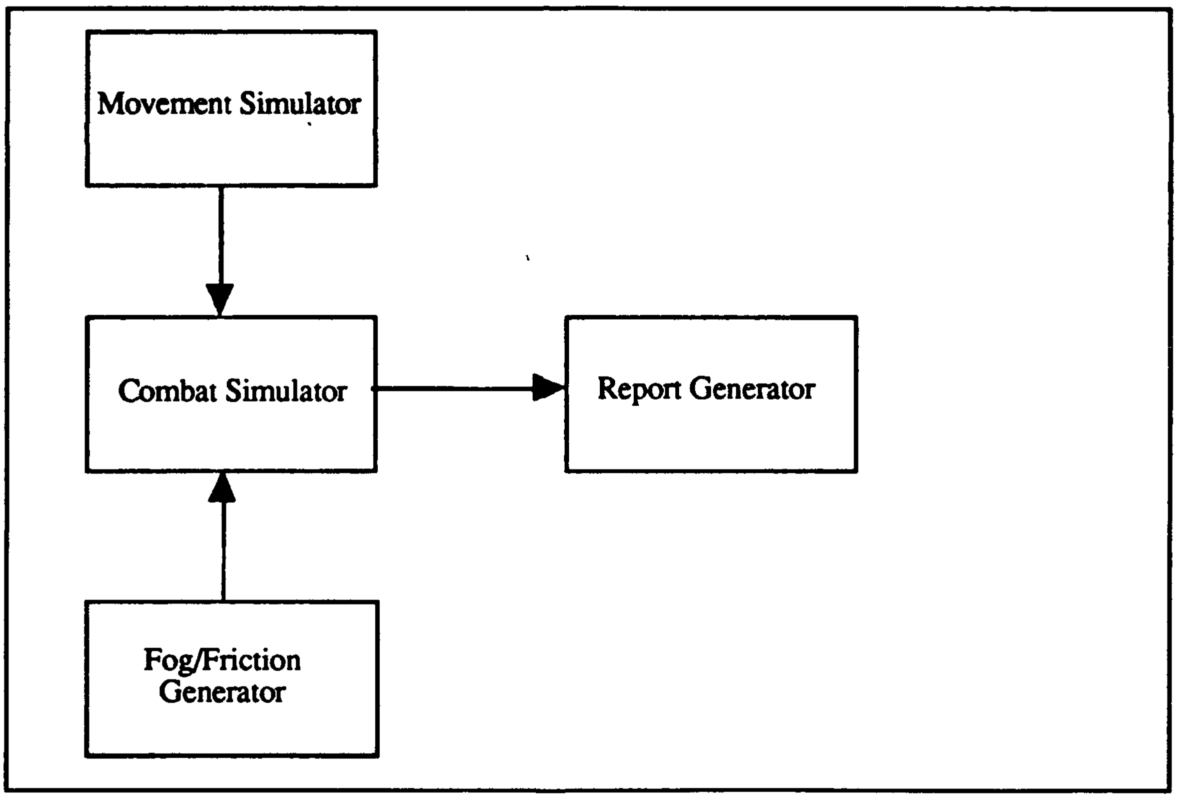

Figure 4.5. Architectural view of the BSM.

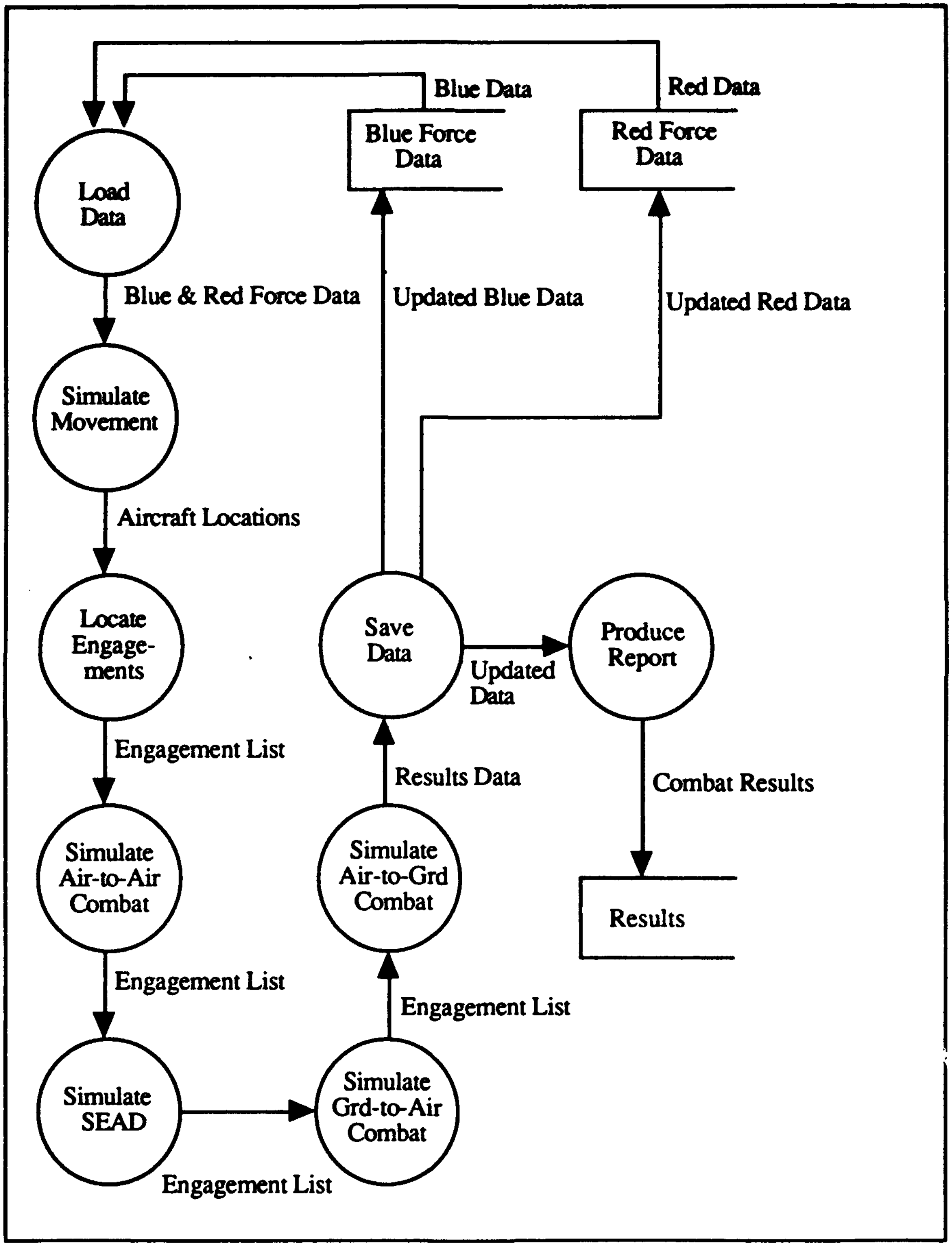

Figure 4.6. BSM data flow diagram.

4.2.5.1 Simulation of Aircraft Movement

After loading data for the blue force and the red force from external files, the aircraft squadron movement is simulated. This simulation occurs in the following sequence:

1. Landings — Squadrons designated to land during this cycle are landed at the appropriate airfields and begin maintenance. If a required airfield is non-operational, i.e., has sustained too much damage from an attack, then the respective squadrons will not be able to land there. Instead, the squadrons will remain airborne and must locate an operational airfield at which to land.

2. Movement — Aircraft will change positions on the map, flying to their next waypoints as determined by the flight plan information loaded from the data files. Aircraft which are on the ground at an airfield, will take-off and move to their first waypoint. Aircraft which are airborne will simply move to their next waypoint.

3. In-flight Refuelings — Any in-flight refuelings which are planned for the current cycle are simulated next. For refueling to take place, both the tanker squadron and squadron to refuel must be at the same map coordinate. In addition, the tanker squadron must have fuel to provide.

4. Crashes — Squadrons which are airborne and have depleted their supply of fuel are crashed. All aircraft in the squadron are destroyed and are no longer available for use.

4.2.5.2 Simulation of Combat

Combat occurs whenever opposing forces occupy the same location. This may involve opposing aircraft squadrons for aerial combat, aircraft squadrons and an opposing fo.ce's land resources for air-to-ground combat, or a combination of the two. After aircraft have been moved, the BSM creates an engagement list based upon the aircraft squadrons' current locations. Combat is then simulated for each engagement in the list. For combat involving different types of aircraft, the following rules of engagement are used. These rules of engagement were developed by the Military Art and Science department at USAFA.

1. Air-to-air combat is simulated first. This involves all airborne aircraft in an engagement. Within this category of combat, fighters versus fighters is first simulated. If the attacking force has fighters escorting other non-fighter aircraft, and the defending force has more fighters than the attacking force, then the defending force will engage the attacking force with an equal number of fighters. The defending force's remaining fighters will be used to engage the attacking force's non-fighter aircraft. If the defending force has less fighters than the attacking force, then all of the defender's fighters will be used to combat the opponent's fighter aircraft. Here is a short example to illustrate this concept. Assume the Red force has 30 fighter aircraft escorting 10 bombers. The Blue force defends with 35 fighters. In this case, 30 Blue force fighters will combat the 30 Red force fighters, and the remaining 5 Blue fighters will combat the 10 Red bombers. Fighter-bombers have air-to-air capability and are thus treated as fighters when engaging enemy fighters. However, if fighter-bombers are out numbered

by the opponent's fighters, they will jettison their bombs (without having armed them). This means they loose their air-to-ground capability until they land to re-arm. After fighter versus fighter combat is simulated, fighter combat against other types of aircraft is simulated.

2. Suppression of Enemy Air Defenses (SEAD) is simulated next. If W-11s are present over an opponent's land resource havingSAM units, the W-11s will engage the SAMs.

3. Ground-to-air combat occurs after SEAD. This involves SAMs, if present, engaging enemy aircraft. The SAMs will fire upon any enemy aircraft; however, W-11s are immune to SAMs. During an engagement, SAMs will be fired until all of the SAM unit's launchers are empty, or all enemy aircraft involved in the engagement have been downed. At the conclusion of ground-to-air operations, the spare SAMs will be loaded onto the empty launchers for use during the next cycle of simulation.

4. The last type of combat to be simulated is air-to-ground. Here, bombers and fighter-bombers will drop bombs on the enemy's ground targets.

4.2.5.3 Computing Combat Results

The results of combat are computed based upon combat factors and hardness values. The basic process involves comparing the total combat factor for one force, to the total combat factor of the opposing force. (For air-to-ground engagements, the air-to-ground combat factor is compared to the target's hardness value.) The greater the difference between the two, the greater damage is inflicted by the stronger force upon the weaker force.

4.2.5.3.1 Simulating Fog and Friction of War

The famous Prussian military thinker Carl von Clausewitz coined the terms fog of war and friction of war. Fog of war concerns the unknown, in that a commander has to make decisions without having all of the information he needs. Friction of war concerns things going wrong. This is basically Murphy's Law — if something can go wrong it will and at the worst possible time. This may be due to human error, fatigue, or mechanical failure, just to name a few causes of friction of war. History has many examples of how the fog and friction of war have impacted battles. At the Battle of Midway during World War II, an entire American torpedo squadron was rendered ineffective when arming devices on the torpedoes were installed improperly. Thus, the weapons were useless and almost every one of the aircraft in the squadron were lost. In the same battle, a Japanese scout plane located the American fleet before the American commander knew the location of the Japanese forces. However, due to a mechanical problem with the scout plane's radio, the information could not be relayed to the Japanese commander. If not for this fog and friction of war, the Japanese may have won the Battle of Midway.

To simulate this important aspect of warfare, a fog and friction adjustment is made to each force's combat factors before determining the results of an engagement. Due to fog and friction, forces may not be 100% effective. A value between 54% - 100% is randomly generated for each force for each type of engagement. This value is then applied to the forces' total combat factors. The adjusted combat factors are then used to compute losses due to combat. In this manner, a user of the testbed, acting as a commander, cannot know with absolute certainty how effective his force may be in combat.

4.2.5.3.2 Combat Losses

Table 4.4 shows the loss rate of aircraft in fighter versus fighter combat. The strong force is the one with the greater combat factor. Aircraft losses are listed as percentage of fighter planes involved in the engagement. Since bombers, Wild Weasels, and tankers do not have air-to-air combat factors, Table 4.4 cannot be used to compute losses for combat involving fighters and other aircraft. Instead, other aircraft losses are calculated as a function of the number of active fighter aircraft (fighters with active air-to-air weapons) involved in the engagement. Other losses are calculated as the number of active fighters, minus 20 - 30% of the active fighters. The 20 - 30% figure is generated randomly, and represents other aircraft that successfully evade fighters. This loss calculation is shown in the equation below, where OL is the number of other (non-fighter) aircraft lost and AF is the number of active fighters.

$$OL = AF - (AF * \mathrm{RandInt}(20, 30) / 100)$$

Losses of SAMs due to W-11s is based upon the number of active W-11s and the type of SAMs. A W-11 has a 75% probability of destroying an S-01 launcher and a 50% probability of destroying an S-02 launcher. Losses of aircraft due to SAMs is based upon the information shown in Table 4.2. Finally, ground damage is simply computed as a function of the air-to-ground combat factor and the target's hardness value. The equation used is

$$GD = AG_{CF} / H * 100$$

where GD is ground damage, AG_{CF} is air-to-ground combat factor, and H is hardness value. For example, a combat factor of 25 and a hardness value of 200 would result in 12.5% ground damage. Computations are not allowed to exceed 100% ground damage.

Table 4.4 Losses for Fighter Versus Fighter Combat

Ratio of Combat Factors	Strong Force A/C Losses	Weak Force A/C Losses
1 : 1	0 - 10%	0 - 10%
1.1 - 1.3 : 1	0 - 10%	same + 10% more
1.4 - 1.6 : 1	0 - 10%	same + 20% more
1.7 - 1.9 : 1	0 - 5%	same + 35% more
2 - 2.5 : 1	0 - 3%	same + 45% more
2.6 - 3 : 1	0 - 2%	50 - 75%
>3 : 1	0%	80 - 100%

4.2.6 Combat Reports

At the conclusion of each cycle of simulation, a results file is generated. The file is named *results.n*, where *n* represents the cycle number. A sample results file is shown in Appendix B. This report lists all events occurring during that cycle of simulation, as well as the results of combat (aircraft losses and ground damage). The following short-hand notation is used in the reports to identify resources. Each resource is listed as a sequence of two letters, followed by a number. The first letter will be either a B or an R to indicate the Blue or Red force. The second letter will be either an A for aircraft squadron, S for SAM unit, or L for land-based resource. Finally, the number represents the resource's ID. For example, RA2 is the Red force's aircraft squadron #2, and BL12 is the Blue force's land target #12.

4.3 User Interface Module

The UIM presents the user with a graphical display representing the simulated air war environment. The interface is mouse-driven and uses pull-down menus for ease of use. The UIM provides the following functions to the user:

- A visual display of the air war map.
- Specification of resources to be used in an air war simulation.
- Deployment of resources to their initial map positions for the start of an air war simulation.
- Simulations of air wars.
- The capability to interrogate the graphical display to obtain status information about resources.

4.3.1 The Graphical Display

The UIM presents the simulated air war environment to the user via a graphical display on the workstation's screen. The hexagonal air war map is drawn on the screen, with coordinates listed across the top of the map and down the left-hand side. In addition, bitmapped graphic icons representing resources are displayed on the map at their proper locations. The only aircraft resources to be shown on the screen will be those currently airborne. If more than one type of resource occupy the same map location, then the appropriate icons are displayed one on top of the other with a slight off-set. This way the user can tell at a glance if more than one resource is at the same map coordinate. Finally, land-based resources which have been destroyed (damaged level is 100%) will have an X through their icons. Similarly, those ground resources with damage levels between 50 - 99% will have one diagonal line drawn through their icons. Figure 4.7 shows a sample portion of an air war map as displayed on the workstation's screen. Notice the command menu bar at the top of the map. Figure 4.8 illustrates the icons used to represent resources.

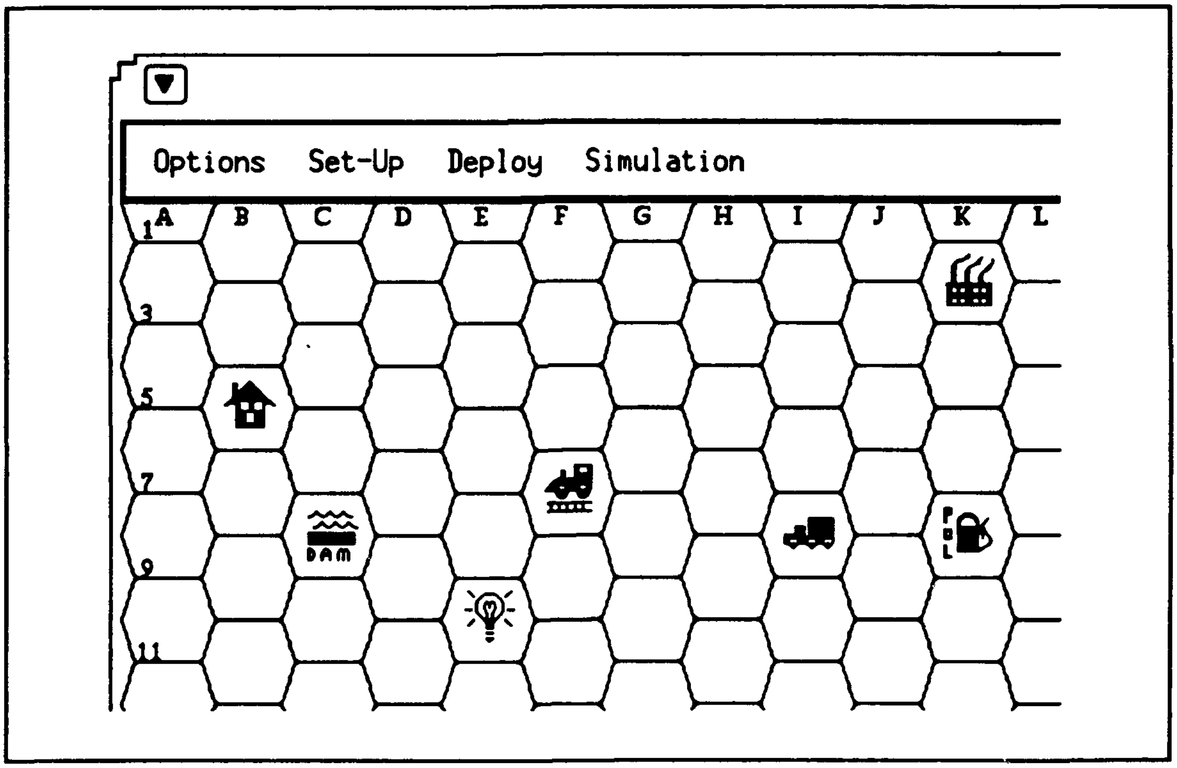

Figure 4.7. A portion of the air war map displayed by the UIM.

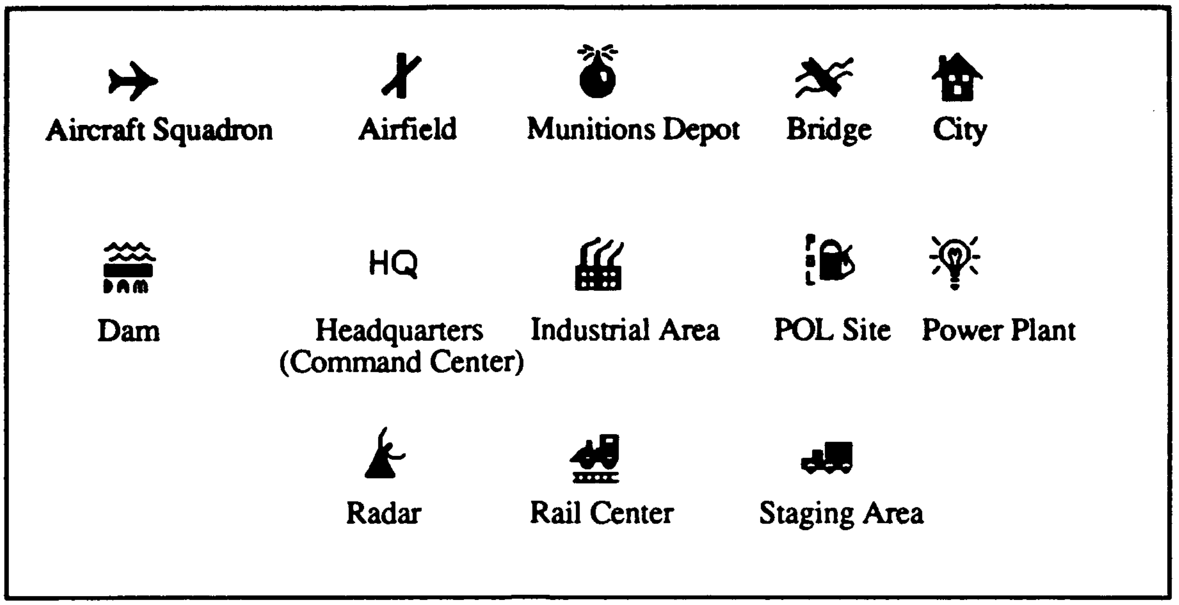

Figure 4.8. The graphical icons displayed by the UIM.

An array data structure is used to maintain information about the air war map. Each hexagon on the map corresponds to an element of the array. Each array element contains the x and y screen coordinates for the upper left most and lower right most vertices of the respective hexagon. Thus, any x-y screen coordinate may be mapped to a specific hexagon by determining which array element has the vertices which bound that coordinate. Furthermore, two functions provide for the translation of map coordinates to array index values. The first function, *coord_to_val*, takes a map coordinate (a string of letters and digits) and converts it to a unique numeric value which relates to a specific array index. The other function, *val_to_coord*, takes a numeric value and converts it to a map coordinate string. The locations of resources are stored internally as numeric index values, rather than as strings of letters and digits. This allows for fast look-up of x-y screen coordinates when displaying a resource icon on the map.

4.3.2 UIM Operation

The UIM is mouse-driven and employs pull-down menus for ease of use. The user selects a particular command by pointing to the command name in the menu with the mouse pointer (arrow displayed on the screen) and pressing the left-most mouse button. Depending on the command, a series of pop-up windows may appear on the screen, requesting additional information from the user. Refer to Appendix A, the testbed user's manual, for complete information about using the testbed and the UIM. Figure 4.9 is an architectural view of the UIM. A data flow diagram of the UIM is shown in Figure 4.10.

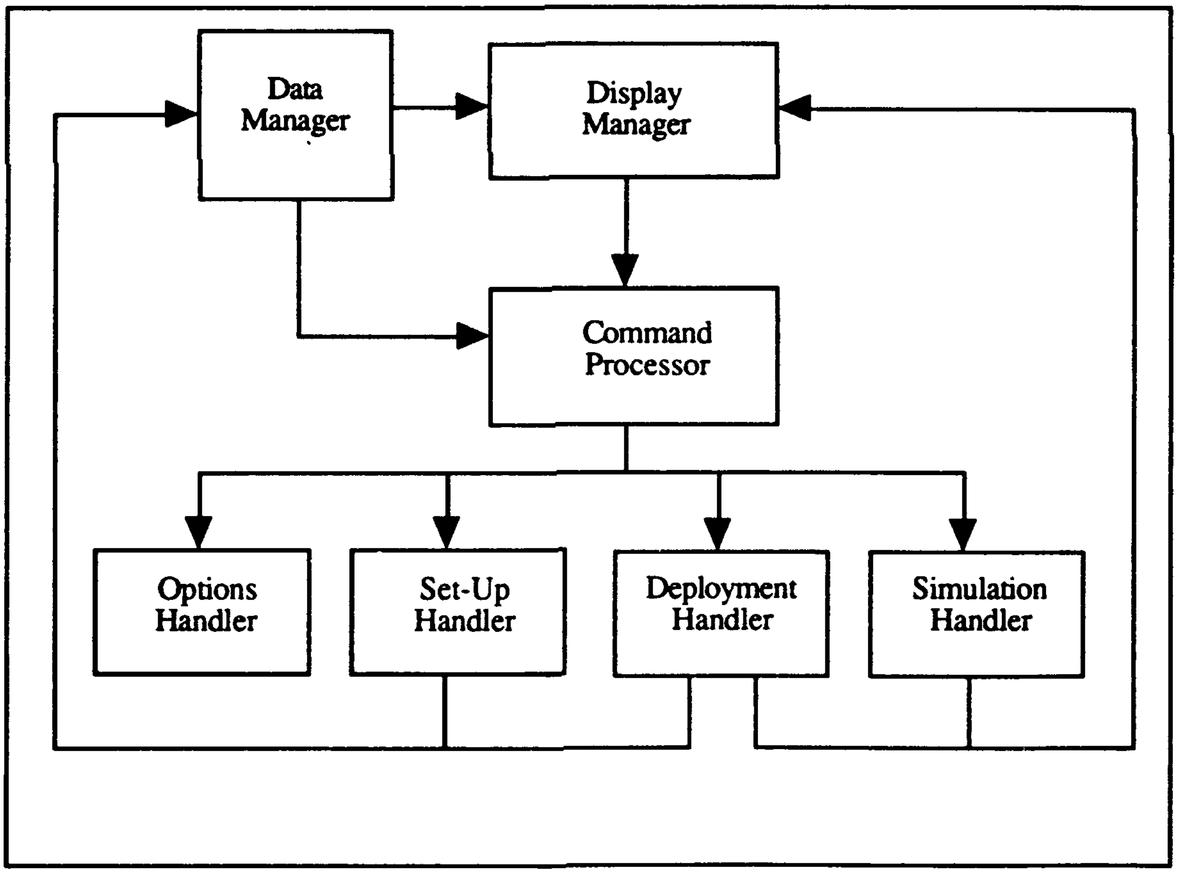

Figure 4.9. UIM architectural view.

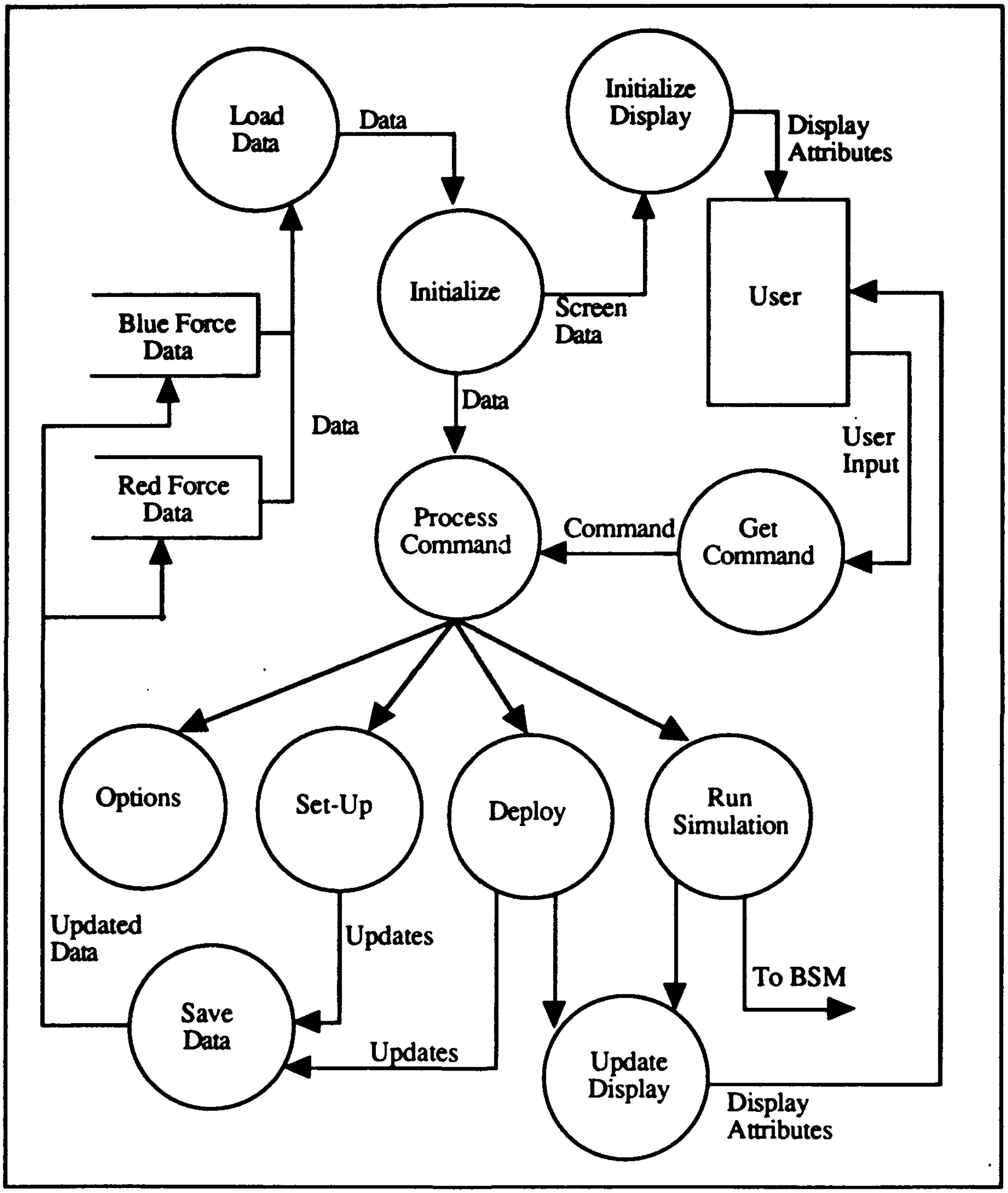

Figure 4.10. UIM data flow diagram.

When the testbed begins operation, data is loaded from external files. This data is used to initialize the data structures for the UIM and the rest of the testbed. This information is also used to initialize the screen display on the workstation. After initialization, the system waits for the user to select a command from the menu. As the user invokes commands, they are processed, the external data files are updated if necessary, and the screen display is updated. Unless a *Quit* command is selected, the UIM will repeat the command loop and wait to process the next user input.

4.3.2.1 Options

The *Options* command allows the user to set the mode for simulations, to toggle the use of sound effects, and to quit the testbed. The testbed may be operated in one of three modes. Mode 0 uses one Sun workstation and lets both the Blue and Red forces to be controlled by human users. (The users alternate entering data for their respective force in this mode.) Mode 1 uses two workstations and has both forces controlled by human users. (With two workstations, the users may enter data simultaneously for their forces.) Mode 2 uses multiple workstations. A human user controls the Blue force from one workstation, while the Red force is under the control of the DPSM. The DPSM runs on multiple workstations which are networked to one another. Each workstation employed by the DPSM has an intelligent decision making process (agent) operating. These agents cooperate with each other for solving tasks during simulations.

4.3.2.2 Set-Up

The *Set-Up* function allows the user to establish resources to be used in air war simulations. Specifically, aircraft squadrons and SAM units may be created for both the Blue force and the Red force. As with the rest of the UIM, the *Set-Up* function prompts the user, via a series of pop-up dialog windows, to specify attribute values for each resource being constructed. For aircraft squadrons, these attributes consist of aircraft type, number of aircraft in the squadron, aircrew experience level, and weapon type. SAM unit attributes are SAM type, number of launchers, and number of spare SAMs. Figures 4.11 and 4.12 are flowcharts for the processes of establishing aircraft squadrons and SAM units respectively.

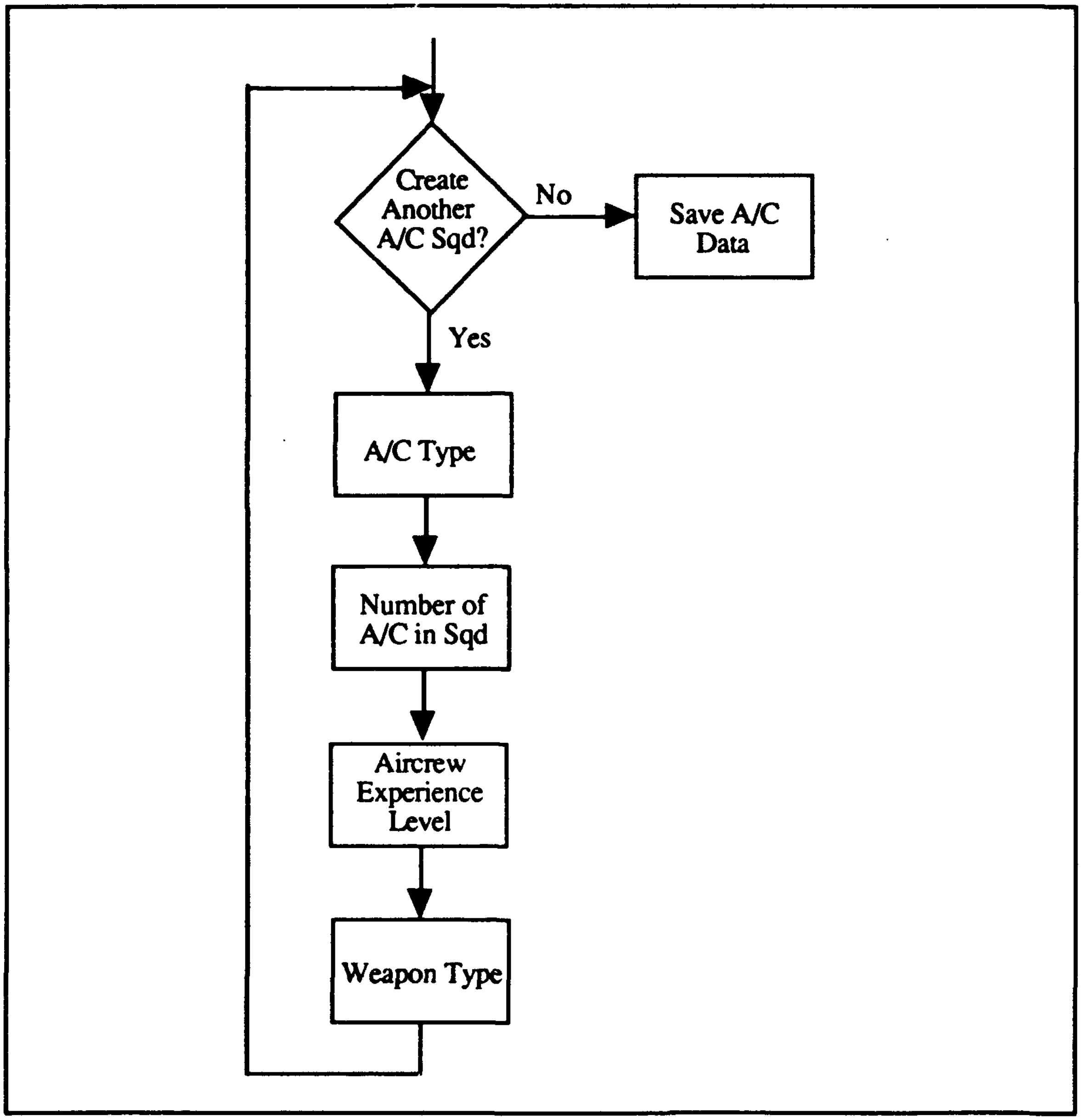

Figure 4.11. Flowchart for establishing A/C resources.

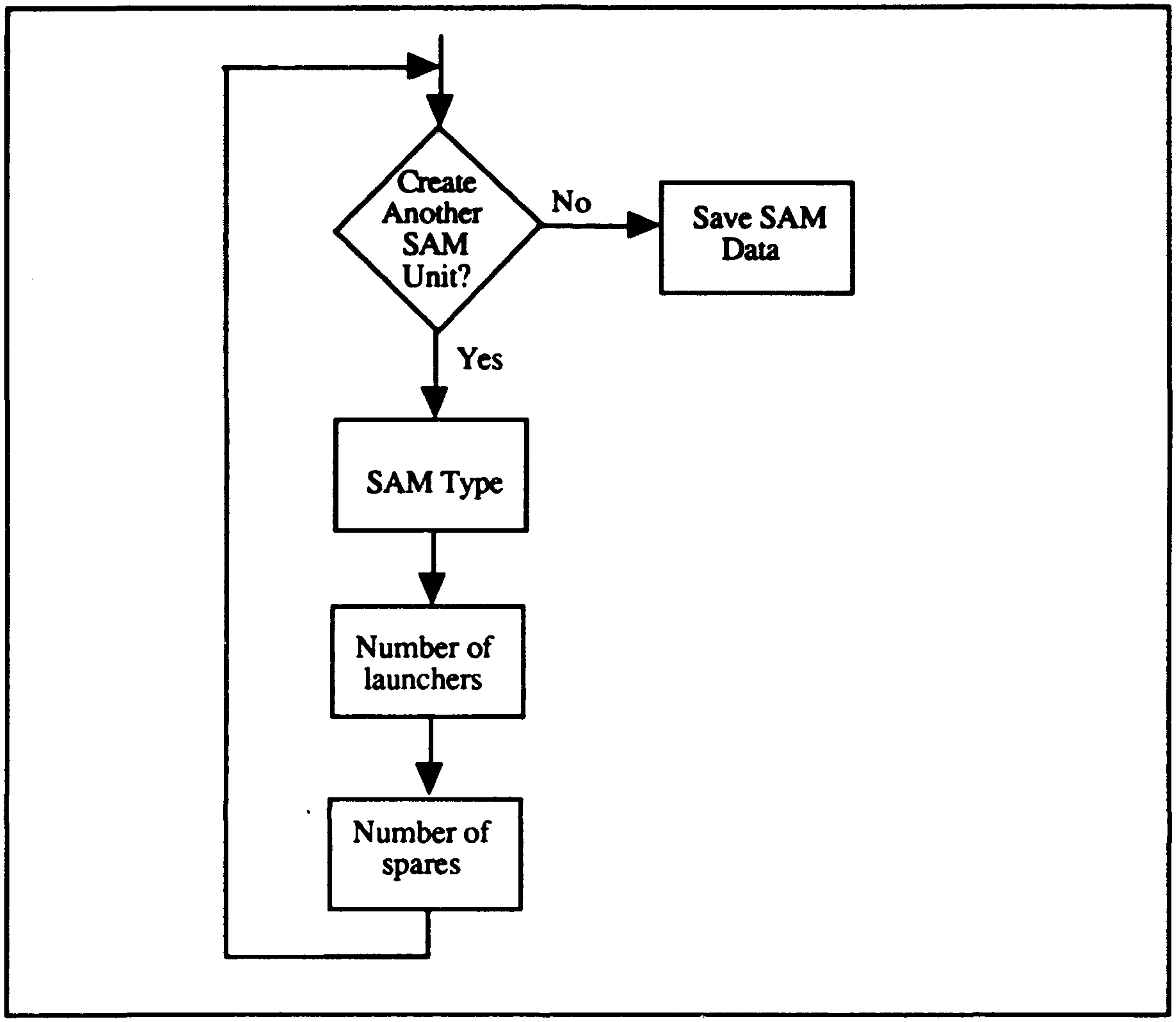

Figure 4.12. Flowchart for establishing SAM resources.

4.3.2.3 Deploy

The *Deploy* function lets the user place resources onto the air war map. In this manner, the user can specify the initial locations of resources for the start of an air war. When using the *Deploy* function, land-based resources must be deployed first. The UIM will prompt the user with a dialog window containing a list of valid land-based resources. The user can simply select the desired resource from the list, and then use the mouse to point to the location on the map to deploy the land target. An icon representing that resource will then appear on the map. The user can repeat this process as necessary. With land-based resources deployed, the user may deploy aircraft and SAM resources. Aircraft squadrons must be deployed at an airfield. Again, the UIM will prompt the user with a dialog window. The window will contain the list of aircraft squadrons established by the user with the *Set-Up* function. The user selects the desired squadron from the list, then points to the airfield icon at which the squadron should be deployed. This process is repeated for all remaining squadrons in the list. SAM deployment works in a similar manner, with the exception that they may be deployed at any land-based resource to provide air defense for that resource. In addition, only one SAM unit may defend a land target. When deploying aircraft squadrons and SAM units, the workstation will beep if the user specifies an invalid map location.

4.3.2.4 Simulation

The *Simulation* function provides two basic features — capabilities for the user to query the status of resources and to simulate the next sequence of events in the current air war. By selecting the Status command from the *Simulation* menu, the user can activate the status feature. With status active, the user may click on any map location and the UIM will provide a pop-up text window displaying the current status of all resources at that map location. This allows the user to query interactively the testbed after each cycle of simulation for up to date information about resources. Figure 4.13 is a sample status display. For land-based resources, the current damage level will be shown. For SAM units, the status function will list the current number of launchers and spares available for use. For aircraft squadrons, the status function will show the type of aircraft, the number of aircraft in each squadron, the number of air-to-air (AA) and air-to-ground (AG) engagements left, and the fuel status. For airborne aircraft, fuel status will be a number indicating the amount of fuel remaining. For aircraft on the ground, fuel status will be either an A (indicating the squadron is available for a mission) or an M (indicating that the squadron is in maintenance and not available for a mission).

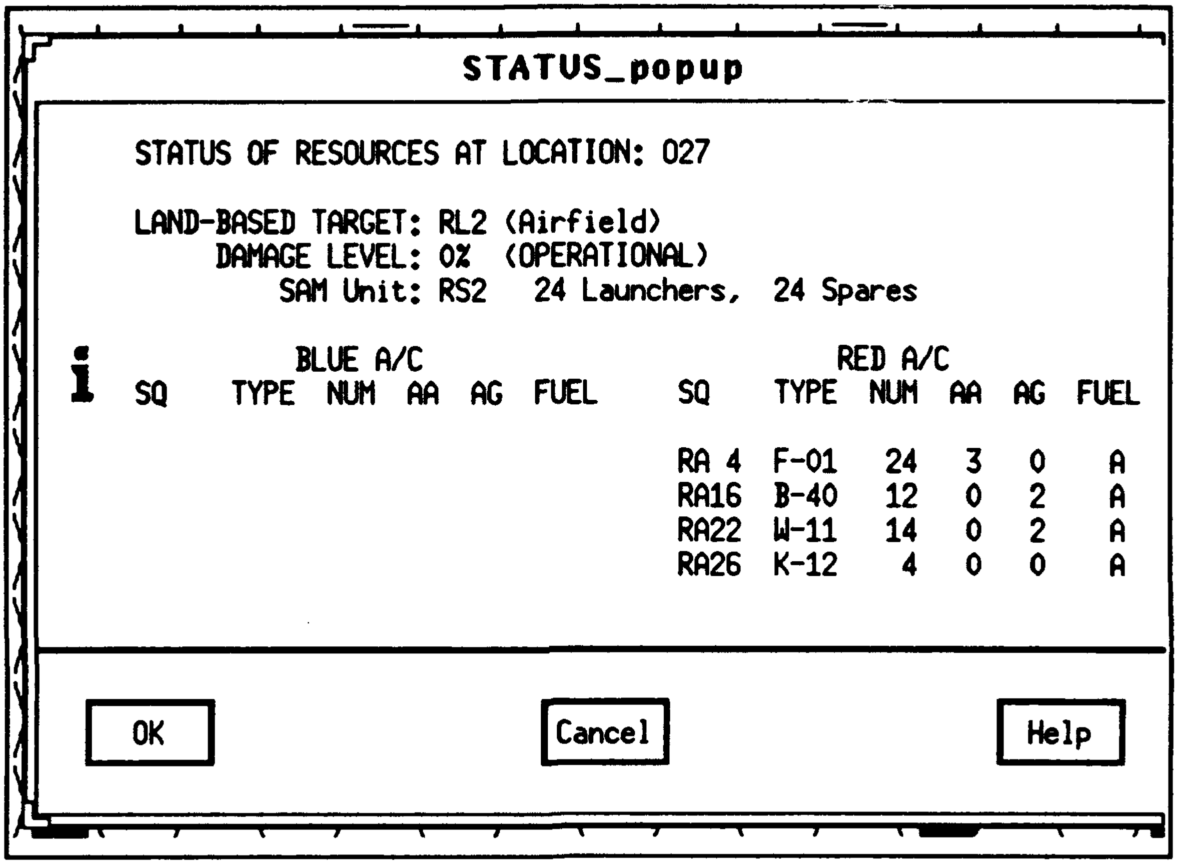

Figure 4.13. Sample status text window.

The Next Cycle command from the *Simulation* menu, will cause the next sequence of events in the air war to be simulated. Depending on the mode of operation, information about the next set of events may be requested from the user or the DPSM. If the UIM invokes the DPSM, the DPSM will access a tasks file, construct a solution for those tasks, and provide the resulting solution to the BSM (see Section 4.4). After obtaining the necessary data, the UIM will invoke the BSM to actually simulate the next cycle of the events in the air war environment.

4.4 Distributed Problem Solving Module

The DPSM implements the DAI methodology described in Chapter 3. The DPSM is an independent program, separate from the rest of the DAW testbed. The program is named *agent*, as it represents an intelligent problem solving agent. The program is run on each processor representing an agent for solving problems during air war simulations. The overall architecture of the DPSM is shown in Figure 4.14.

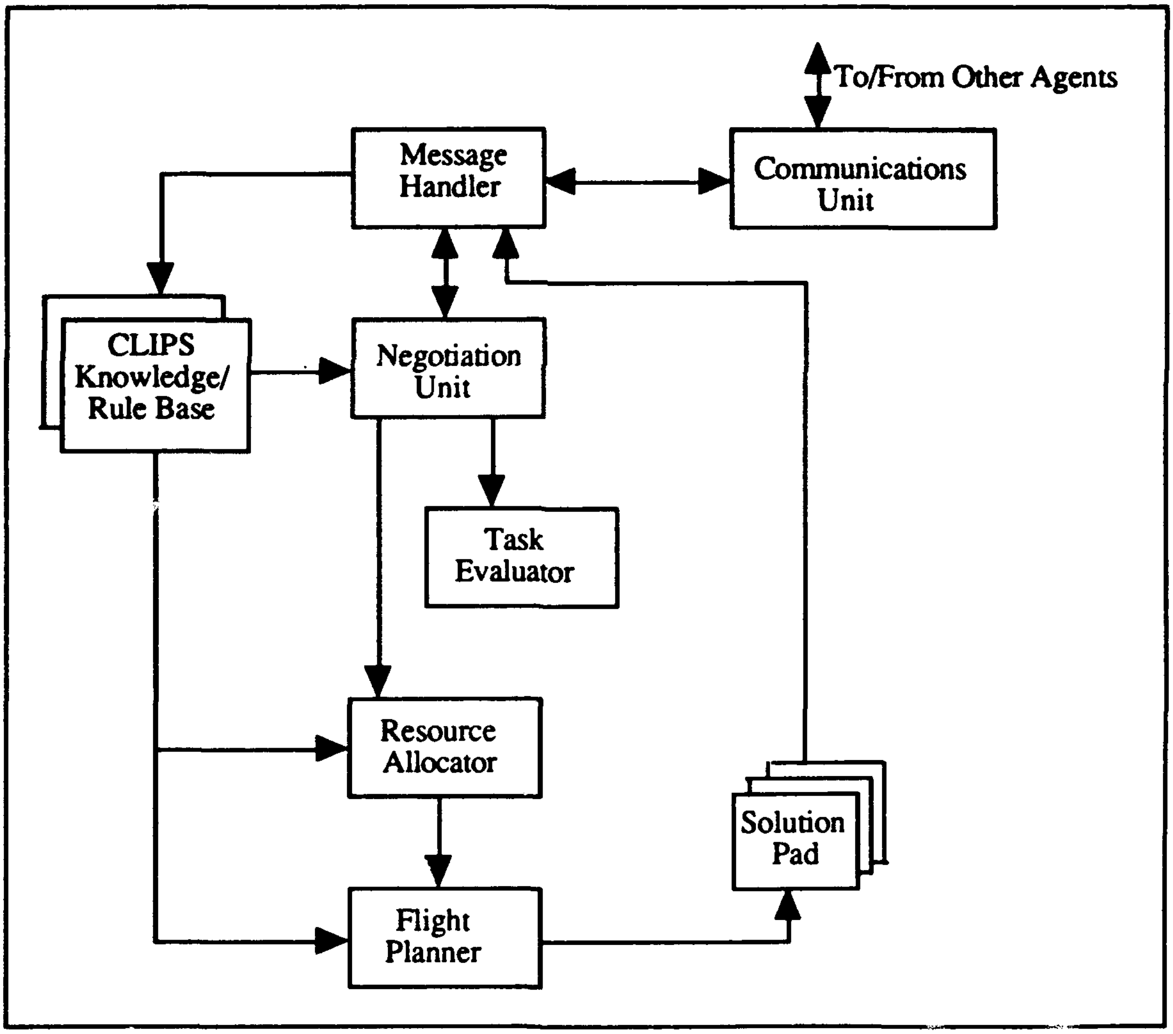

Figure 4.14. Overall architecture of the distributed problem solving module.

As shown in Figure 4.14, the following major components comprise the DPSM:

- Communications Unit — Used to communicate with other problem solving agents.
- Message Handler — Handles incoming messages from other agents and prepares messages for transmission to other agents.
- CLIPS Knowledge and Rule Base — Maintains the knowledge base containing information about the state of the world and the rule base for use in decision making.
- Negotiation Unit — Negotiates the distribution and assignment of tasks with the other agents.
- Task Evaluator — Evaluates the agent's capability to handle a particular task; i.e., computes quality measures.
- Resource Allocator — Allocates aircraft squadron resources for task accomplishment.
- Flight Planner — Constucts flight plans for aircraft allocated by the Resource Allocator. Flight plans are sequences of waypoints for aircraft to follow when accomplishing missions.
- Solution Pad — Holds the solutions for each task assigned to this agent.

4.4.1 Communications

The DPSM uses Unix sockets for communications. When an *agent* program is started on a processor, it begins by reading into memory a configuration file named *config.dat*. A sample of the information contained in this file is shown in Table 4.5. The data in *config.dat* allows each agent to know its own identity, as well as the other agents it must interact with. The *agent* program interrogates the host computer upon which it resides for the host name. Using this information, the agent knows its own name and the airfield it controls. (There is a one-to-one mapping of airfields to agents. Agent1 controls airfield #1, Agent2 controls airfield #2, and so forth.) Agent0 is a special agent under the control of the user. This agent resides on the same workstation as the BSM and the UIM. Through Agent0, the user can announce tasks to the other agents. Also, the other agents pass their task solutions to Agent0 for processing by the BSM. In other words, Agent0 acts as a go-between for the testbed and the distributed agents.

Table 4.5 Agent Configuration Data

Agent ID	Agent Name	Host Name	Port#
00	AGENT0	seine.eas.asu.edu	5000
01	AGENT1	thames.eas.asu.edu	5001
02	AGENT2	severn.eas.asu.edu	5002
03	AGENT3	volga.eas.asu.edu	5003

A standard data structure is used for sending and receiving messages between agents. This data structure is shown in Figure 4.15. All messages are stored in the same format. The message header field, hdr, allows the message handler to interpret the type of message and act accordingly. Table 4.6 lists the various message headers used by the agents. As messages may arrive at any time and in any order, the message handler routes messages to appropriate "mailboxes" which may be accessed by the agent as needed.

```
typedef struct MSG_RECORD
{
        unsigned short      agent;        /* Agent that sent message */
        unsigned short      hdr;          /* Message header */
        char                data[100];    /* Message */
        unsigned short      len;          /* Message length */
} MSG_RECORD;
```

Figure 4.15. Message data structure.

Table 4.6 Message Headers

Message Header	Value	Description
MSG_STOP	0	Informs the agents to stop executing. This message is sent by Agent0 when the user quits the testbed.
MSG_TASKS	1	Announces the availability of tasks. This message is sent by Agent0. The message consists of the name of the file containing the list of required tasks.
MSG_BIDS	2	List of bids (quality measures) for each task. The message consists of task IDs and their associated quality measures.
MSG_ASSIGN	3	Informs the agents of task assignments. Sent by Agent0. The message contains agent IDs and the IDs of the tasks for which they are responsible. (This is really just confirmation from Agent0 concerning each agent's self-appointment as a task commander.)

Table 4.6 Message Headers (Continued)

Message Header	Value	Description
MSG_RESOURCE	4	List of available resources for task accomplishment.
MSG_DONE	5	Informs agents that a particular task has been solved or cannot be solved. If task has been solved then the message will list the IDs of resources to be used to accomplish the task. This may include resources being borrowed from other agents to resolve a conflict.

4.4.2 CLIPS Knowledge and Rule Base

The CLIPS knowledge and rule base maintains information about the simulated air war environment and contains a prioritized set of rules for use in decision making respectively. Knowledge about the "air war world" is stored as a collection of CLIPS facts. The knowledge base includes information about land-based resources (types, locations, damage level, etc.), as well as aircraft resources. Each agent knows the aircraft squadrons which belong to other agents; however, agents do not know the status of those resources until informed by others during task assignment. After each cycle of simulation, the BSM provides result data to the DPSM for use in updating the knowledge base.

Each agent uses a copy of the same set of rules. The rules provide for the allocation of aircraft resources based upon the type of task and the conditions of the air war environment. Some examples of the types of rules used by the DPSM are shown in a high-level, English-like form in Table 4.7. The rules are prioritized and organized in a manner that general solutions may be refined as more rules are invoked. Thus, the more is time available for allocating resources, the more rules may be fired and, consequently, the better the resulting solution for a particular task.

Table 4.7 Example Rules

1.	If Task_Type(task) = Strategic_Aerospace_Offense Then AC_Required(Bombers)
2.	If AC_Type(squadron) = AC_Required(type) Then Valid_AC(squadron)
3.	If Status(squadron) = Available Then Valid_AC(squadron)
4.	If Range_AC(squadron) >= Distance_To_Target(target) Then Valid_AC(squadron)
5.	If Time_To_Target(squadron) <= Deadline(task) Then Valid_AC(squadron)

4.4.3 Negotiation and Task Evaluation

The Negotiation Unit and Task Evaluator deal with task evaluation and assignment as described in Sections 3.4.1 and 3.4.2. An external data file, *weights.dat*, contains specific values for the weighting factors and a threshold value used when evaluating tasks. The negotiation process begins when the air commander, agent0, sends a task announcement message to all the other agents. This message is followed by a file containing descriptive information about the available tasks (IDs, task types, task locations, lists of opposing resources, priorities, and deadlines). Each agent then has access to the tasks list and can begin computing quality measures.

The Negotiation Unit first sorts the tasks list according to distance to each task. When evaluating tasks, agents give preferences to those tasks that are nearest them. This is a heuristic relating to the efficient utilization of resources. In general, the closer the agent to the task, the faster the turn-around time of its resources used for the task. The Task Evaluator examines each task in sorted order and computes the corresponding quality measure. Using the formula shown in Section 3.4.1, quality measures are determined based upon the agent's available resources. The quality measures are then packaged into a message the format of which is

$$\text{ID-Task}_1\ \text{QM-Task}_1,\ \text{ID-Task}_2\ \text{QM-Task}_2,\ldots,\text{ID-Task}_n\ \text{QM-Task}_n$$

This bid message is then broadcast to all other agents. The Negotiation Unit receives bid messages from the other agents via the Message Handler. The bids are placed in a list data structure illustrated in Figure 4.16.

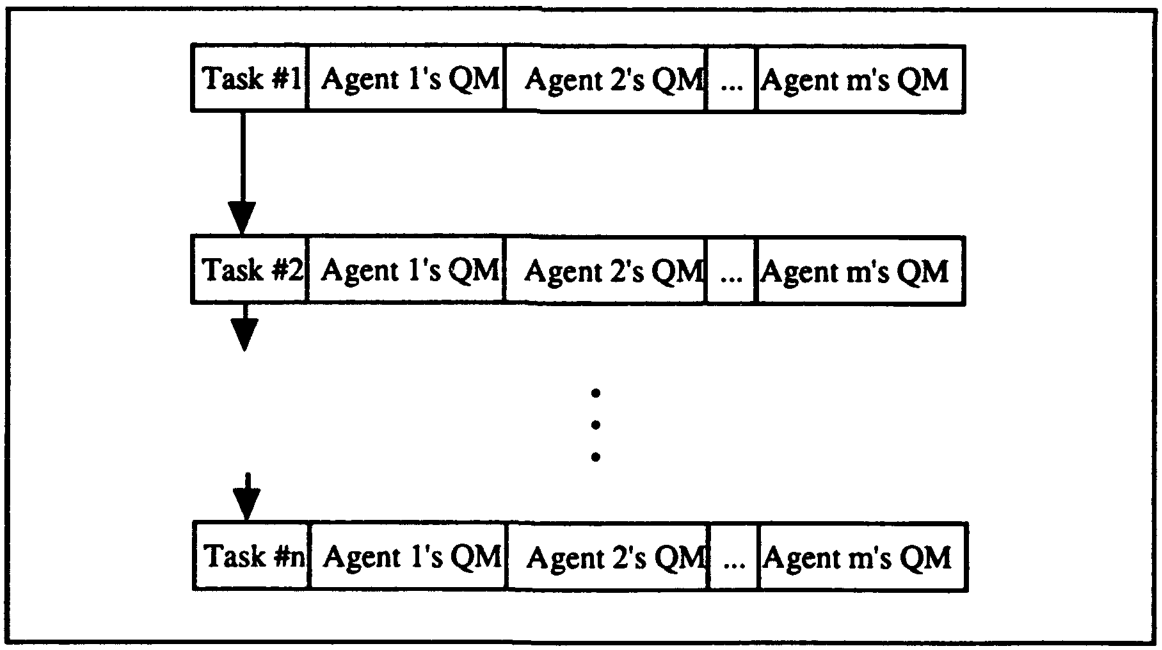

Figure 4.16. Task bids lists.

The bids (QMs) for each task are then sorted by QMs. Using the sorted lists of QMs and the threshold value, the Negotiation Unit determines which agents become task commanders. For each task, it simply examines the first QM in the list (the best) and the second QM (next best). If the first QM exceeds the second by the threshold value, the agent having submitted that QM becomes the self-appointed task commander. Those tasks which have no task commander are then re-negotiated. The Negotiation Unit also keeps track of each agent's workload (number of assigned tasks) and the previous bids. If the current round of negotiation results in the same QMs being submitted as the previous round, then this indicates that the agents cannot improve their bids and quiescence has been reached. Lastly, Agent0 confirms the task commanders by sending a task assignment message to all agents. This is a message listing each task and the corresponding task commander. In this manner, the user of the testbed acting as an air commander, may override self-appointments of tasks and specify task assignments.

4.4.4 Problem Solving

The Resource Allocator, Flight Planner, and Solution Pad operate together to solve assigned tasks. Each agent solves its assigned tasks one at a time in priority order. (After negotiation, the task list is sorted according to priority and deadline. If two tasks have the same priority, the task with the shortest deadline gets solved first.) The Resource Allocator invokes the CLIPS rule base to allocate aircraft resources needed to accomplish specific tasks. The list of resources are then passed to the Flight Planner which constructs a flight plan for the resources. The flight plan is a set of waypoints which routes a path for the aircraft to follow in order to reach the destination target and return to a friendly airfield. The task solution (resources and associated flight plans) are maintained on the solution pad. Figure 4.17 shows a data flow diagram outlining the major functions and flow of date during problem solving.

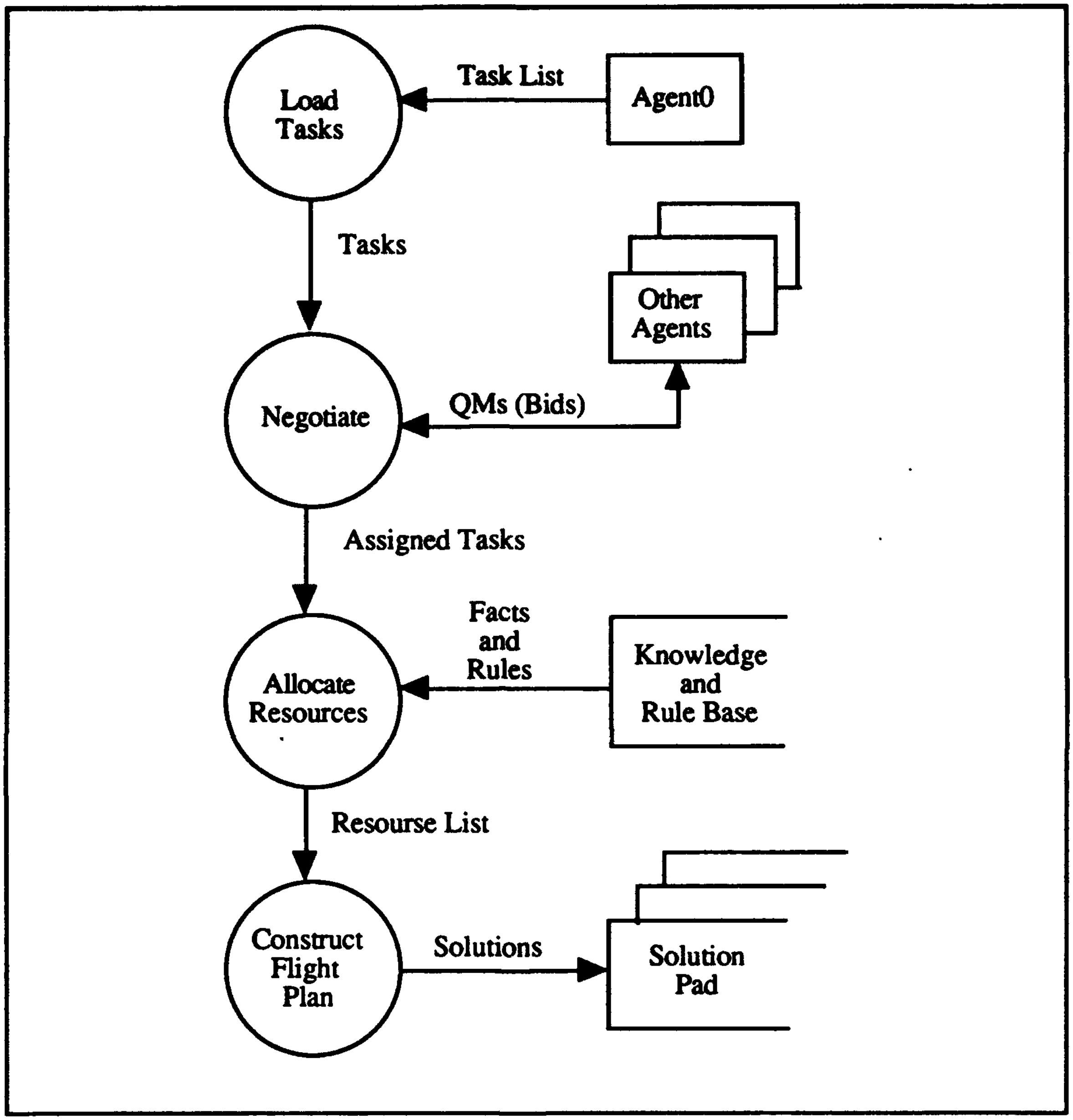

Figure 4.17. Data flow diagram of the problem solving process.

As resources have various capabilities, some resources are general in nature and can be used for different types of tasks. Other resources, however, can deal only with a specific type of task. For example, bomber aircraft can be used in an air-to-ground attack role only. On the other hand, fighter-bombers have more capabilities and can be employed for ground attacks or used for air-to-air missions such as defensive counter air and escort roles. The Resource Allocator attempts to allocate resources with limited, specific capabilities first. Furthermore, in an attempt to minimize conflicts, resources are not allocated from other agents unless an agent has no resource of its own which may be used for the particular task at hand.

Offensive tasks may be solved far in advance of when they must be accomplished. Thus, for offensive tasks, agents will generally have adequate time for problem solving. However, most defensive tasks are time critical in nature. An agent may not have enough time to construct the best possible solution but must rely on an adequate solution. For this reason, the rule base is prioritized and lets the Resource Allocator invoke additional rules as time allows to find better solutions. Each pass through the rule base attempts to refine the current solution by allocating a better mix of resources for the task at hand. Furthermore, resource conflicts are identified and resolved using the hierarchical iterative conflict resolution technique described in Chapter 3. During problem solving, an agent may be interrupted by another agent with a higher priority task notifying it of a resource conflict. This lets the agent know which resource is being borrowed. It may impact the agent's current solution for a task—the agent may now have to allocate a different resource for its task.

4.4.4.1 Solution Pad

The solution pad is a linked list structure containing solutions for each assigned task. A solution consists of a set of resources allocated for the task and a set of waypoints for each aircraft resource to follow in order to accomplish the task. At each waypoint, an action is listed for the resource. The actions may be either take-off, landing, move (flight between waypoints), combat, refuel, or none. The solutions are also in a form that may be readily accessed by the BSM. During each cycle of simulation, the BSM can obtain the appropriate part of a task solution in order to simulate the events for that task. Figure 4.18 shows a sample solution for a task.

Task #2

Type: Strategic Aerospace Offense
Location: O27
Importance: 8 Urgency: 5 Priority: 40
Deadline: 4
Allocated Resources: 9 10 15 16
Events:

Cycle #1
Sqd #9 action: Take-Off from: W27 to: S25
Sqd #10 action: Take-Off from: W27 to: S25
Sqd #15 action: Take-Off from: W27 to: S25
Sqd #16 action : Take-Off from: W27 to: S25

Cycle #2
Sqd #9 action: Combat from: S25 to: O27
Sqd #10 action: Combat from: S25 to: O27
Sqd #15 action: Combat from: S25 to: O27
Sqd #15 action: Combat from: S25 to : O27

Cycle #3
Sqd #9 action: Move from: O27 to: S25
Sqd #10 action: Move from: O27 to: S25
Sqd #15 action: Move from: O27 to: S25
Sqd #16 action: Move from: O27 to: S25

Cycle #4
Sqd #9 action: Landing from: S25 to: W27
Sqd #10 action: Landing from: S25 to: W27
Sqd #15 action: Landing from: S25 to W27
Sqd #15 action: Landing from: S25 to: W27

Figure 4.18. Sample task solution.

4.4.4.2 Time Criticality

Time criticality is modeled in the DPSM by limiting the number of inference cycles (rule firings) allowed when solving a task. A mapping is made between task urgency and allowed number of firing of rules. The lower the urgency (less time stress involved) for a task, the more rules may be fired. An external data file is used to maintain the mapping information. A user can easily modify these values for use with particular experimental runs of the testbed.

4.4.5 Operating the DPSM

An agent process is invoked by running the *agent* program on a particular workstation. (The workstation must be listed in the configuration file in order for the agent to know its identity.) Two command line options are available when starting the *agent* program. One option allows the user to set specific watch items for CLIPS. The option for this is *-w* followed by an *f* to watch facts, an *r* to watch rules, or an *a* to watch activations. With these items set, CLIPS will print information to the console when the item is effected; e.g., when facts are asserted or rules fired. The second option lets the user specify a particular CLIPS rule base to be used. To set this option, *-f* is listed on the command line followed by the name of the file containing the rule base. If this option is not used, then the standard file *aw-rules.clp* is used. In the example shown below, an agent process is started with watch items set for facts and rules.

```
agent -wfr
```

The *agent* program also displays a trace of its actions to the console during operation. This allows a user to trace through the decision making process of an agent to determine how an agent bids for tasks and how task solutions were constructed. Appendix C shows a sample trace from one problem solving session.

The *agent0*.program is run on the same workstation as the rest of the testbed (BSM and UIM). With *agent0*, the user may control the operation of the other agents. By interacting with *agent0*, a user may announce available tasks to the other agents and, thus, begin distributed problem solving. As tasks are solved, the solutions will be accessible by the BSM for simulating events in the air war environment. As the air war simulation progresses, the user can add tasks as desired. Figure 4.19 is a simple architectural view of the testbed, with agent processes functioning as part of the DPSM.

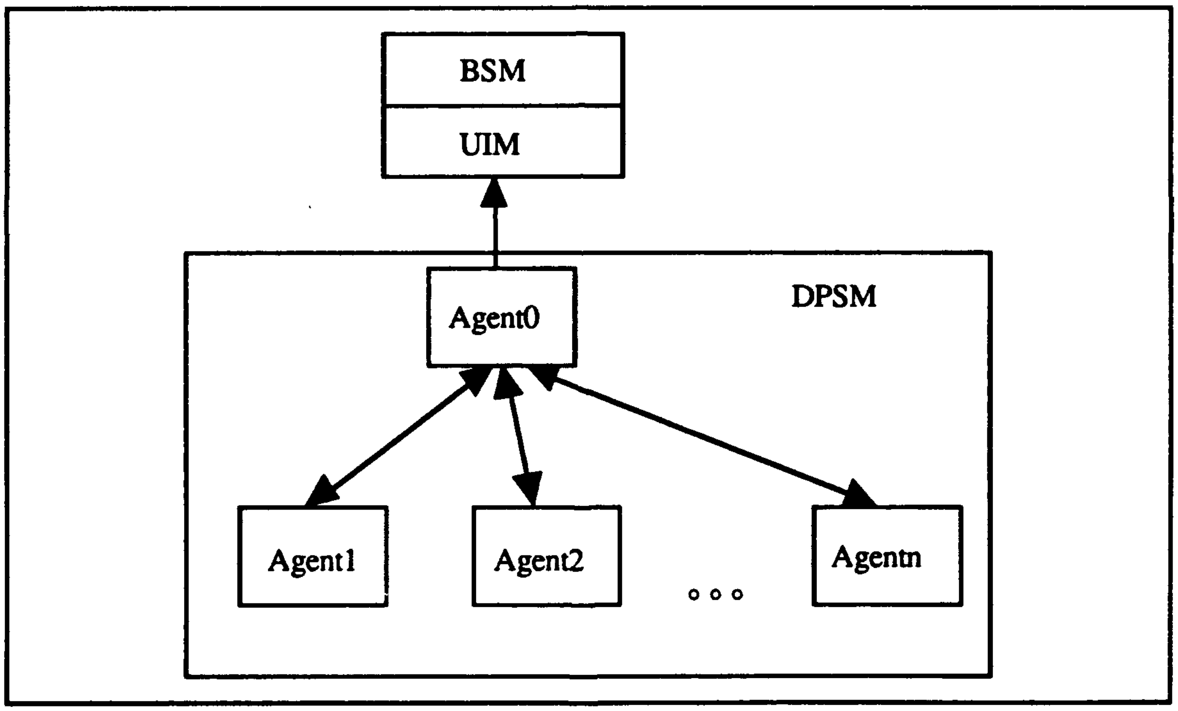

Figure 4.19. Architectural view of the testbed with agent processes.

CHAPTER 5

EMPIRICAL STUDIES

5.1 Experiment Overview

Empirical studies were conducted using the Distributed Air War testbed to investigate the DAI methodology developed in this research. Experiments consisted of air war simulations between two opposing forces. During simulations, decision makers had to allocate aircraft resources in order to respond to tasks (air missions). The resources required to handle tasks were distributed geographically in the air war environment. In the experiments, the decision makers were either:

- a group of humans;
- a group of intelligent computer agents as provided by the DPSM of the testbed; or
- a single intelligent computer agent.

Throughout the simulations, data was collected concerning the allocation of resources, level of task accomplishment (effectiveness), and resource losses. This data was then used to analyze the performance of the different decision makers.

5.1.1 Air War Scenarios

For the experiments, air war scenarios were constructed as a basis for air war simulations. The scenarios involved two opposing forces — a Blue Force and a Red Force. The forces mirrored one another, in that, both forces were equal in terms of types, amounts, and relative locations of resources. Thus, no force had an unfair advantage over the other at the start of an air war. In each scenario, both forces had one headquarters unit and three airfields a piece, as well as various other types of land-based resources. Aircraft squadrons were deployed amongst the airfields and were available for use at the start of each air war simulation. Refer to Appendix D for the list of resources used in each scenario.

The air war scenarios were constructed with the assistance of a former faculty member of the Military Art and Science Department at the United States Air Force Academy. The scenarios were based upon the types of air war simulations used to teach air power strategy to second year cadets at USAFA. As such, the ten scenarios used in this study represent the range of those simulations in terms of aircraft resources and air missions (tasks). Due to the limited availability of the domain experts, developing the scenarios was a lengthy and time consuming process conducted over a six-month period. In addition, the simulation of one scenario generates only three distinct data points (Blue aircraft losses, Red aircraft losses, and average level of accomplishment). With ten scenarios, ten data points in each of the three categories can be collected. Thus, with so few data points, it was not possible to obtain statistically significant results of the comparisons among the different approaches used in the empirical studies.

Various tasks (air missions) were pre-planned in advance for the Red Force. These consisted of offensive missions to attack specific land-based resources belonging to the Blue Force. In addition, aircraft resources were identified for defensive roles to ensure adequate protection of Red Force resources during the simulated air wars. For the Blue Force, a set of tasks was developed. Each task was assigned a static importance value and a deadline. The importance values ranged on a scale from 1 to 10, with 1 being the least important and 10 being the most important. More than one task could have the same importance value. The deadline attribute associated with a task indicated the time by which the task must be accomplished. (Each cycle of simulation in the testbed is equal to one unit of time.)

Two types of tasks were used in the experiments: Strategic Aerospace Offensive missions and Defensive Counter-Air missions. Strategic Aerospace Offensive missions involve the strategic bombing of enemy targets. To handle such tasks, a mix of resources is needed which has air-to-ground capabilities (for bombing land-based resources), air-to-air capabilities (for protection of aircraft from enemy fighters), and SEAD capabilities (for suppressing enemy SAMs). Thus, a task of this kind will require an attack formation composed of some mixture of fighters, bombers, fighter-bombers, and Wild Weasels. The second type of task used in the experiments, Defensive Counter-Air, involves defending resources from enemy attacks. Aircraft with air-to-air capabilities are needed in this case to intercept and engage opposing aircraft in order to minimize damage to friendly resources.

5.1.1.1 Aircraft Resources for the Scenarios

Table 5.1 lists the roles that specific types of aircraft were limited to in the scenarios. Each type of aircraft was not necessarily used in each scenario. Refer to Appendix D for a complete list of all aircraft resources available by scenario. Also, see Table 4.3 for a list of specific capabilities by aircraft type.

Table 5.1 Aircraft Roles

Aircraft Type	Role
F-01	Defensive counter-air (DCA) only.
F-02	Escort only (protecting other aircraft during offensive missions).
F-03	Any air-to-air mission (DCA or escort).
F-10	Any air-to-air or air-to-ground mission.
F-20	Any air-to-air or air-to-ground mission.
F-30	Any air-to-air or air-to-ground mission.
B-40	Air-to-ground (bombing) only.
B-50	Air-to-ground (bombing) only.
B-60	Air-to-ground (bombing) only.
W-11	SEAD missions only.
K-12	Refueling missions only.

The resources used in each scenario were divided among the three agents. In Scenarios 1, 2, 3, 9, and 10, the aircraft resources were divided evenly among the agents, i.e., each agent had the same types and numbers of aircraft as the other agents. Conversely, in Scenarios 4, 5, 6, 7, and 8, the distribution of aircraft resources is not divided evenly among the agents. Some agents have different amounts and types of resources than the other agents. For example, in Scenario 4, Agent1 and Agent2 are assigned 24 F-01s, 20 F-02s, 18 F-03s, 24 B-40s, 20 W-11s, and 4 K-12s each. Agent3, however, is assigned 24 F-01s, 96 F-20s, 12 B-40s, 20 W-11s, and 4 K-12s.

5.1.1.2 Tasks for the Scenarios

The tasks for the Blue Force for each scenario were developed based upon experience with air war simulations as conducted at the USAFA. The tasks for each scenario are geographically distributed throughout the region of conflict, and not centered in any one particular area of the map. A variety of Red targets were selected for offensive tasks, rather than concentrating on only one or two particular types of targets. In this manner, the scenarios could fully exercise the problem solving abilities of the decision makers. Refer to Appendix E for a list of all tasks used in each scenario.

5.1.2 Scenario Processing

In the simulations for each air war scenario, the Red Force's actions were in accordance with the pre-planned operations. The actions for the Blue Force were under the control of the appropriate decision makers. During the simulations, the decision makers were provided with a set of tasks to accomplish. The decision makers would then have to solve the tasks by allocating aircraft resources and computing flight paths for the aircraft to follow. Furthermore, the decision makers did not know in advance all of the tasks they would have to accomplish for a given scenario. At various cycles during a simulation, the decision makers would be given a subset of the tasks. Thus, the decision makers did not know in advance at which cycles of a simulation to expect tasks, either.

5.1.3 Decision Makers

In order to compare the performance of the different decision makers, all of the air war simulations were repeated employing the various decision makers. The decision makers in this research consisted of a group of three humans, a group of three intelligent computer agents provided by the DPSM of the testbed, and a single intelligent computer agent.

5.1.3.1 Human Decision Makers

The human decision makers were three faculty members from the Military Art and Science Department of the USAFA. Each individual had experience in teaching airpower theory and doctrine, as well as in running air war simulations to educate cadets about the use of airpower. Specific characteristics of the individual human decision makers is listed in Table 5.2.

Table 5.2 Characteristics of the Human Decision Makers

	Person #1	Person #2	Person #3
Military Rank	Sqd Ldr (RAF)	Capt (USAF)	Capt (USAF)
Years in Military	23	12	15
Main Military Duties	Pilot Mission Planner Instructor	Missile Ops Instructor	Navigator Instructor
Years Teaching Airpower Theory	2	3	4
Years Experience with air war simulations	2	3	4

The three individuals participating in the experiments were provided in advance with copies of the user's manual of the testbed. This was to allow them to become familiar with the operation of the testbed. Also, a half-day training session was conducted. This consisted of a demonstration of the features of the testbed, as well as a practice air war simulation.

In a real air war environment, human decision makers (commanders) must be concerned with a great deal of information (weather conditions, intelligence reports, moral and fatigue level of personnel, status of resources, leadership traits of lower level commanders, and interests of the civilian authority, to name just a few). Some automated support is also available to assist the commander when making decisions, but situations involving information overload do occur. The simulated air war environment provided by the testbed attempts to supply users with the same sensation of information overload that commanders in the real world have to deal with. Thus, users of the testbed may not be concerned with the exactly same types of information found in the real world, but the effects of information overload are still present.

The experiments with the human decision makers were conducted at USAFA. Two Sun workstations were employed for these experiments, using Mode 1 of the testbed. Events for the Red Force were input using one workstation, while the team of human decision makers used the second workstation to input data for the Blue Force.

During the studies with human decision makers, the humans were able to obtain feedback during each air war simulation. This feedback consisted of the raw data about aircraft losses and the damage levels sustained by land-based targets (level of task accomplishment). Overall averages per scenario were not provided to the human decision makers. However, it is possible that the humans used this feedback to enhance their learning and, thereby, improve their performance in subsequent scenarios.

5.1.3.2 Distributed Intelligent Agents

The experiments were repeated using a set of three distributed intelligent computer agents. Four Sun workstations were employed. The UIM, BSM, and Agent0 operated on one workstation. Each of the three other workstations ran an agent process (Agent1, Agent2, and Agent3). Every agent had the control of the resources assigned to its respective airfield. The agents as a whole had to solve the tasks specified. Each scenario was run three times with the distributed agents. In the first run of the scenarios, the DAI methodology discussed in Chapter 3 was used for agent coordination, task assignment, and conflict resolution. The agents were allowed to share resources in order to solve tasks. To lessen the propagation of resource conflicts, agents borrowed resources only when they could not solve an assigned task using their own resources. For example, a bomber aircraft is designed specifically for air-to-ground attacks and is the best resource to use for such missions. Fighter-bombers, on the other hand, are a more general purpose resource in that they may be used in either an air-to-air role or an air-to-ground role. In allocating resources for a bombing mission, an agent will first attempt to use its own bombers. If enough bombers are not available, the agent will try to allocate its own fighter-bombers. Only if the agent does not possess enough bombers and fighter-bombers, will it take a resource from another agent.

In the second run of each scenario, the agents were not to share resources. The Hierarchical Iterative Conflict Resolution feature of the testbed was disabled. In this case, each agent was limited to using only its own resources for task accomplishment.

The third run of each scenario, referred to as DPSM*, consisted of distributed computer agents but without the two techniques (Hierarchical Iterative Conflict Resolution and the Task Evaluation/Allocation method) described in Chapter 3. For task assignment, the agent with the best initial bid became the Task Commander for the respective task. No threshold value or renegotiation was employed with DPSM*. Each agent allocated from its own pool those resources required for the assigned tasks. Task priorities were not considered with this approach, either. As such, agents could not "take" resources planned to be used for other tasks by other agents as with HICR. Instead, only those resources not required by an agent for any of its tasks would be shared with others.

5.1.3.3 Single Intelligent Agent

The experiments were repeated a fourth time using a single, centralized intelligent computer agent. Two Sun workstations were employed. One workstation ran the UIM, BSM, and Agent0. One agent process (Agent1) operated on the second workstation. In these runs of the experiments, the single agent had control over all the resources belonging to the Blue Force. Therefore, all tasks were assigned to Agent1. Since no other agents were involved with solving tasks, the negotiation and hierarchical conflict resolution components of the agent process were not invoked.

5.1.4 Experimental Data

Throughout each air war simulation, data was collected for use in analyzing the performance of the decision makers. The two most important types of data collected were aircraft resource losses and level of task accomplishment. Aircraft could be lost either directly or indirectly due to combat. In general, the less adequate the allocation of aircraft for a particular task, the greater the number of aircraft that will be lost in combat. Damage to airfields and support aircraft can result indirectly in losses of other aircraft. For example, an airfield that is completely destroyed is non-operational. Aircraft on the ground at a non-operational airfield cannot take-off. In addition, aircraft cannot land at non-operational airfields. Thus, if aircraft are scheduled to land at a particular airfield that has been destroyed, those aircraft must change course to land at another airfield that is operational. If those aircraft do not have sufficient fuel to reach another airfield, and if tankers are not available for in-flight refueling, those planes will be lost. Finally, those aircraft on the ground at non-operational airfields may be considered lost since they cannot be used in the accomplishment of any tasks. Two of the subgoals of the decision makers, therefore, were to minimize the loss of their (Blue Force) aircraft and to maximize the loss of the opponent's (Red Force) aircraft.

The level of accomplishment for each task is measured as the amount of damage sustained by the intended target for offensive tasks, and the operational level of the target under attack for defensive tasks. These measures may be defined as

$$OL(TGT_i) = 100 - DL(TGT_i)$$

$$A(TSK_j, Off, TGT_i) = DL(TGT_i)$$

$$A(TSK_j, Def, TGT_i) = OL(TGT_i)$$

where $OL(TGT_i)$ is the operational level of target *i*, $D(TGT_i)$ is the damage level of target *i*, $A(TSK_j, Off, TGT_i)$ is the level of accomplishment for task *j* which is an offensive mission against target *i*, and $A(TSK_j, Def, TGT_i)$ is the level of accomplishment for task *j* which is a defensive mission protecting target *i*.

For a given set of tasks, an average weighted level of accomplishment (AWA) is defined as

$$AWA = \frac{\sum_{i=1}^{n} W_i A_i}{\sum_{i=1}^{n} W_i}$$

where n is the number of tasks, W_i is the weight associated with task *i* and A_i is the level of accomplishment for task *i*. For the experiments conducted in this research, the Importance value of a task was used for the weights.

5.2 Experiment Results

5.2.1 Observations of Human Decision Makers

Figures 5.1 and 5.2 show the percentage of aircraft lost for the Blue Force and the Red Force for each experiment employing human decision makers. One of the secondary objectives concerning resource use in combat is to minimize your losses and maximize the losses of the opposing force. Therefore, in the experiments the decision makers wished to keep Blue Force losses low and Red Force losses high.

As previously stated, the decision makers desire to minimize their (Blue aircraft) losses. To accomplish this, sufficient numbers of the proper types of aircraft need to be allocated for tasks so that the Red Force cannot overpower the Blue aircraft in terms of fire power. As shown in the graph in Figure 5.1, the Blue Force experienced high losses of aircraft (more than 50% lost) in Scenarios 1, 7, and 8. In these three scenarios, the human decision makers miscalculated the fuel status and range of aircraft flying in large formations. As a result, these aircraft ran out of fuel before reaching their home airfields and crashed. Specifically, 82 aircraft (19%) crashed in this manner in Scenario 1, 33 aircraft (8%) crashed in Scenario 7, and 77 aircraft (19%) crashed in Scenario 8. Again, these aircraft resources were lost not due to combat, but because of human error in planning the missions. Other Blue aircraft losses are the result of combat, either directly or indirectly.

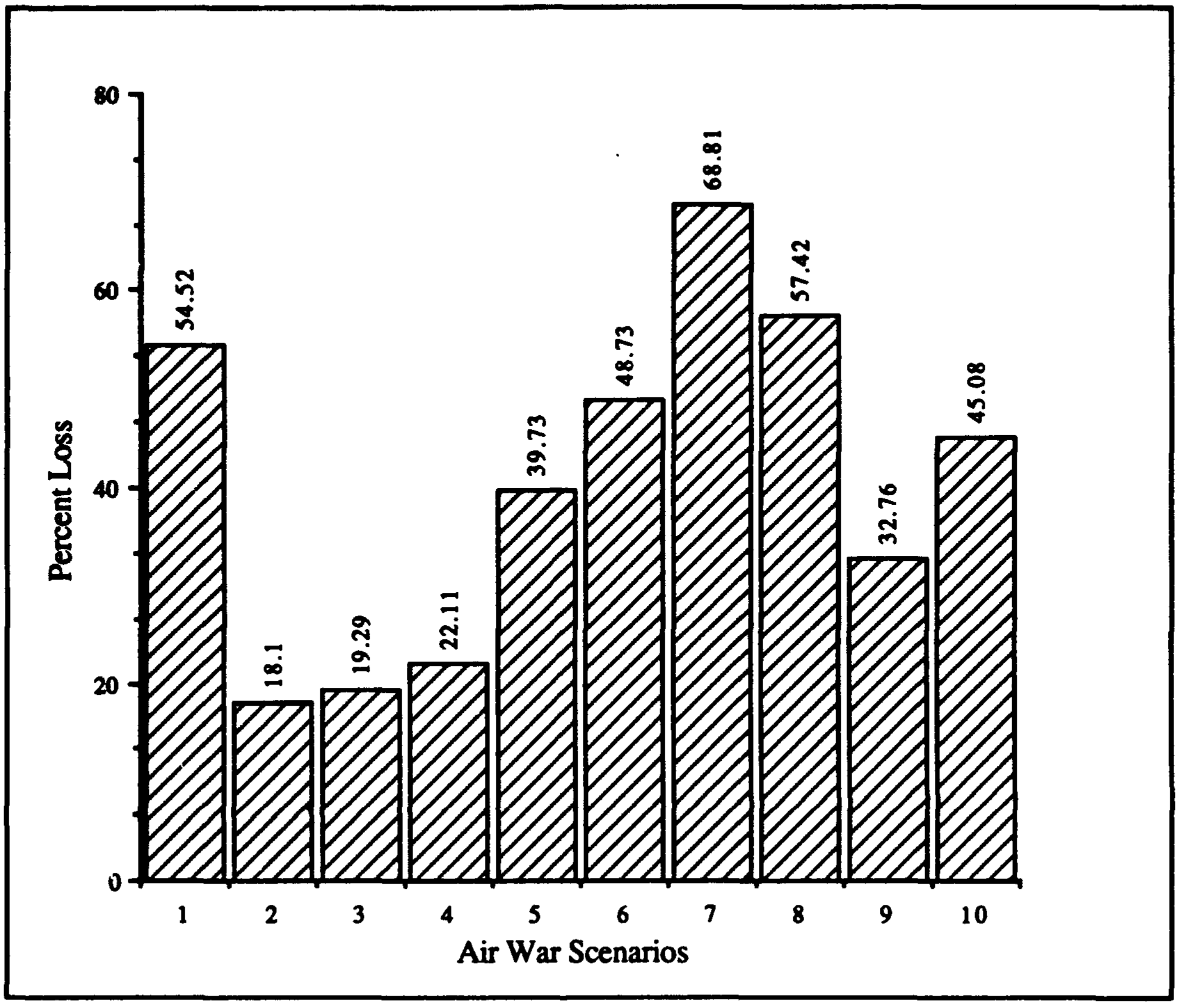

Figure 5.1. Blue aircraft losses (human decision makers).

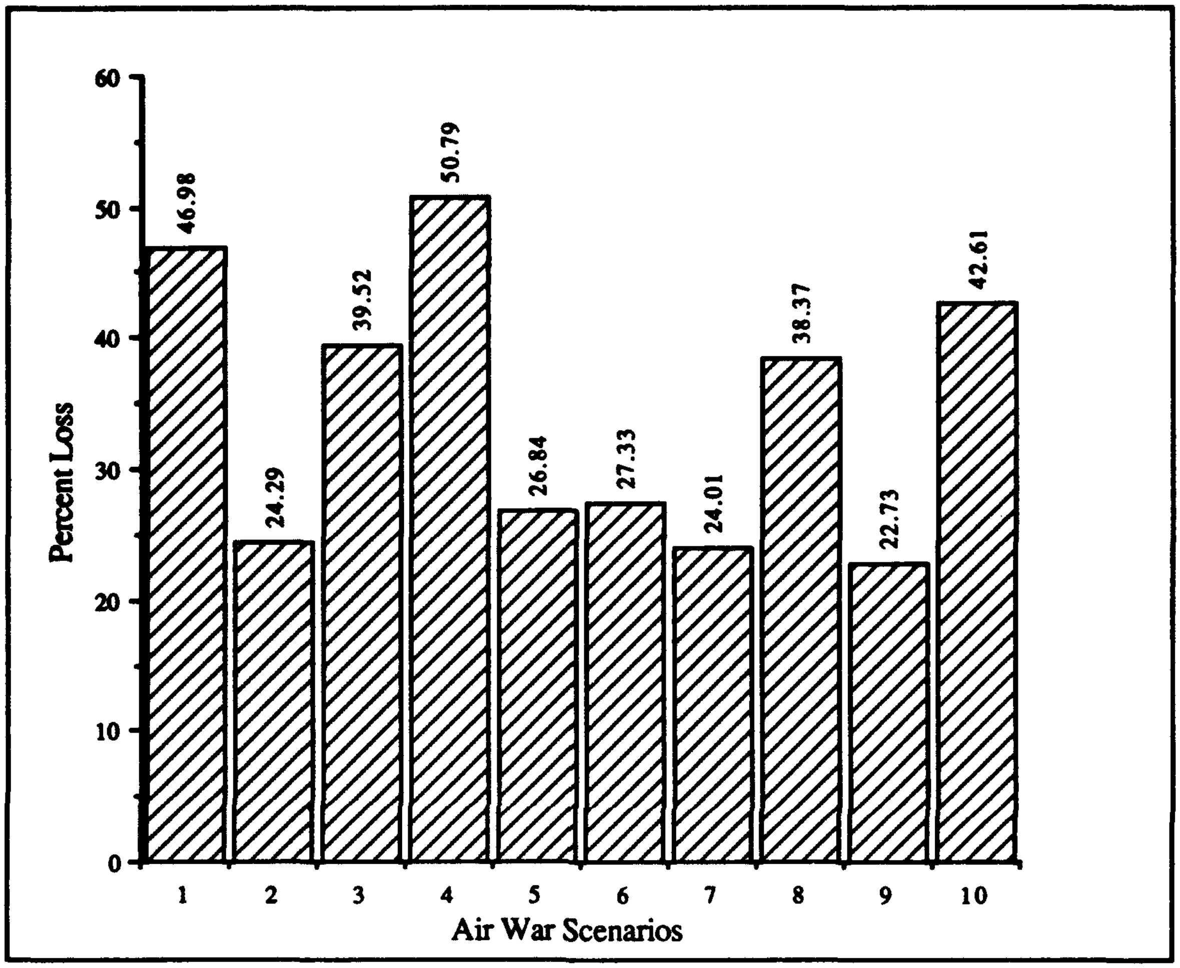

Figure 5.2. Red aircraft losses (human decision makers).

The decision makers wish to maximize the number of Red aircraft lost. In other words, the Blue Force should allocate sufficient aircraft for its tasks to inflict greater damage to the Red Force than the Red Force can inflict upon the Blue Force. As shown in Figure 5.2, the greatest percentage of aircraft lost by the Red Force occurred in Scenario 4, where the loss was 50.79%. In five of the scenarios (2, 5, 6, 7, and 9), the percentage of Red aircraft lost is under 30%. In comparing the losses for both forces, in only three scenarios (2, 3, and 4) is the number of aircraft lost by the Blue Force less than the number lost by the Red Force. In all other scenarios, the Blue Force lost a higher percentage of aircraft than the Red Force. Finally, averaging the losses over all ten scenarios results in an average loss percentage of 40.66% for the Blue Force versus 34.35% for the Red Force.

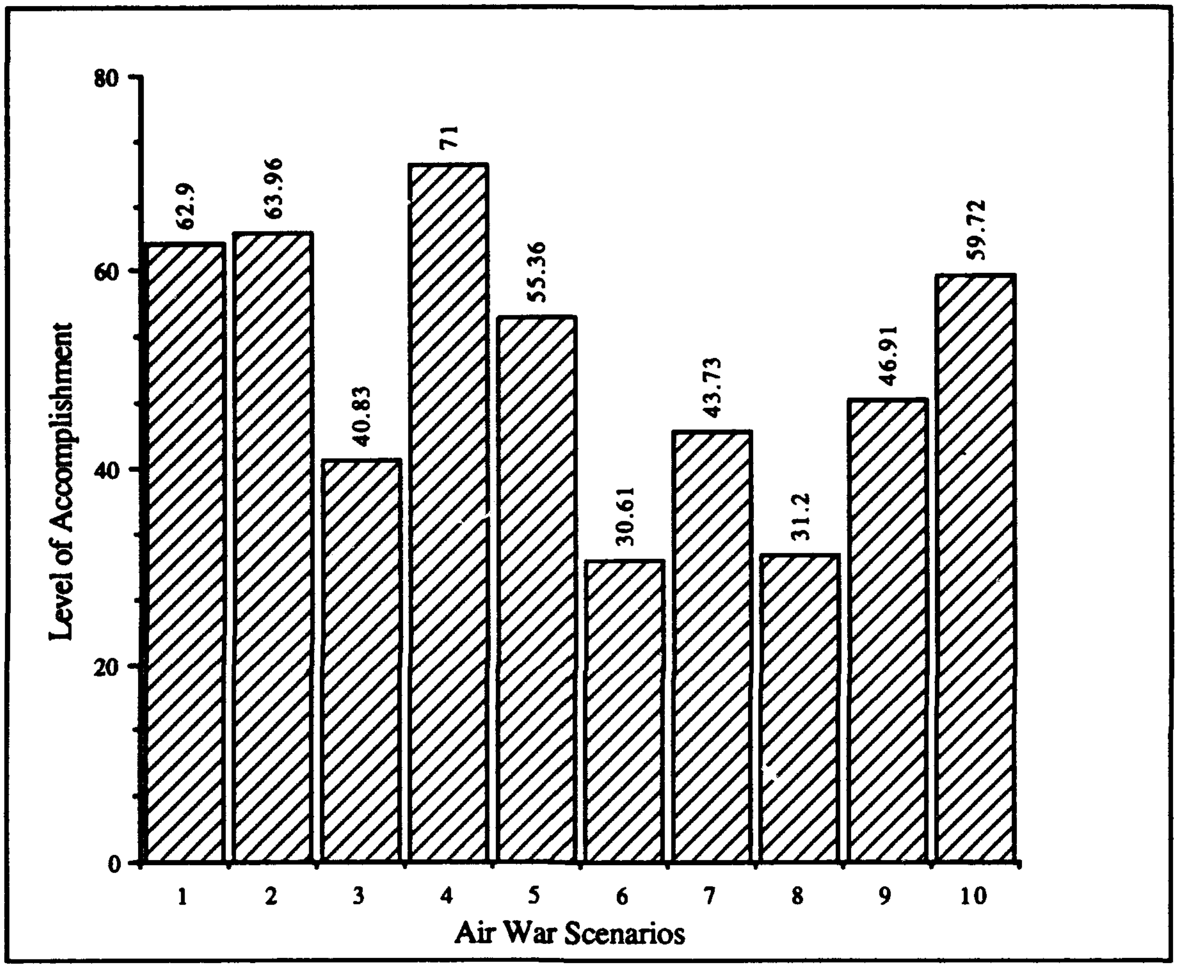

Figure 5.3. Weighted average level of accomplishment (human decision makers).

Figure 5.3 shows the average weighted level of task accomplishment for each scenario. A level of accomplishment of 100% would indicate that all tasks were completely accomplished, i.e., for offensive tasks all Red targets were 100% destroyed, and for defensive tasks all Blue targets remained undamaged (100% operational). The highest average level of accomplishment is in Scenario 4 with a value of 71%. Five of the scenarios (1, 2, 4, 5, and 10) have average levels of accomplishment exceeding 50%. The average level of accomplishment for the human decision makers across all ten scenarios is 50.62%.

5.2.1.1 Behavior of Human Decision Makers

As already shown in the graphs, the Blue Force, under the control of human decision makers, lost on the average a greater percentage of Blue aircraft than the Red Force lost of its aircraft. Additionally, the Blue Force achieved an average level of accomplishment slightly above 50%. From observing the actions of the human decision makers, two interesting explanations about their performance can be made. First, even though the simulated air war environment is a "toy" world, the decision makers must deal with a great amount of information (aircraft availability, distances to targets, fuel status, weapon levels, aircraft endurance, and time-to-target just to name a few). This information overload did result in errors being made by even the experienced humans. As a result of these errors, aircraft resources were lost under conditions not directly related to combat. Of course, the timing of these errors impacted the performance of the Blue Force. For example, at the end of Scenario 1 the Blue Force lost 82 aircraft due to human error. This was after the Blue Force had attempted to accomplish all tasks. Thus, the Blue Force had lost a high percentage of its aircraft, but still achieved a level of accomplishment of 62.9% since tasks were accomplished prior to losing those 82 aircraft. On the other hand, aircraft were lost due to human error near the middle of scenarios 7 and 8. This limited the ability of the Blue Force to accomplish future tasks, thus resulting in lower levels of accomplishment. A second interesting observation concerning the human decision makers involves communication. During the experiments, the individuals were not geographically separated. They freely collaborated among themselves. Even so, miscommunication among the human decision makers often resulted in mistakes when planning missions. On several occasions, information about resources relayed from one individual to another was misunderstood. This resulted in the wrong aircraft being allocated for tasks, or aircraft being directed along an inadequate flight path. In one specific instance, a large formation of aircraft was directed to the wrong waypoint when one individual misunderstood the map coordinate as stated by another person. Because of this mistake, the aircraft strayed too far from the actual flight path and did not have enough fuel to reach the intended target. The aircraft had to return to base without having completed its task.

5.2.2 Observations of Automated Decision Makers

The ten air war scenarios were repeated using the automated decision makers — once with distributed computer agents and resource sharing, once with distributed computer agents without resource sharing, once using a single computer agent, and once using distributed computer agents without the techniques discussed in Chapter 3 (DPSM*). In order to make direct comparisons of the four automated approaches, the fog and friction adjustment had to be removed from the BSM. Since each approach would not necessarily handle tasks in the same order, different fog and friction values could be applied to the same tasks as handled by the different automated approaches. This would add random noise to the resulting data and not allow fair comparisons to be made. Without fog and friction, combat outcomes are handled deterministically, and thus, the different approaches may be compared. (Unfortunately, the impact of fog and friction to the data was not realized until after the experiments with the human decision makers. Therefore, the results of the humans cannot be directly compared with the automated methods.)

Figure 5.4 shows the percentage of Blue Force aircraft lost for each air war scenario by the different automated decision makers. As shown by the graph, the distributed approach without resource sharing lost the most aircraft in each scenario. Without being able to share resources, this approach usually under-allocated the amount of aircraft needed for a given task. The result being that the Red Force could engage the Blue Force in combat with a stronger force, and thus, destroy more Blue aircraft. Both DPSM with resource sharing and the single agent approach performed better in terms of losing fewer Blue aircraft. This makes sense because by sharing resources, or having access to the entire inventory of aircraft, more adequate allocations of resources can be made for task accomplishment. The performance of DPSM* fell between the no-resource-sharing approach and the single agent. The percentage of Blue aircraft lost as averaged across all ten scenarios is 39.90% for DPSM without resource sharing, 29.30% for DPSM*, 25.89% for the single agent, and 24.65% for DPSM with resource sharing.

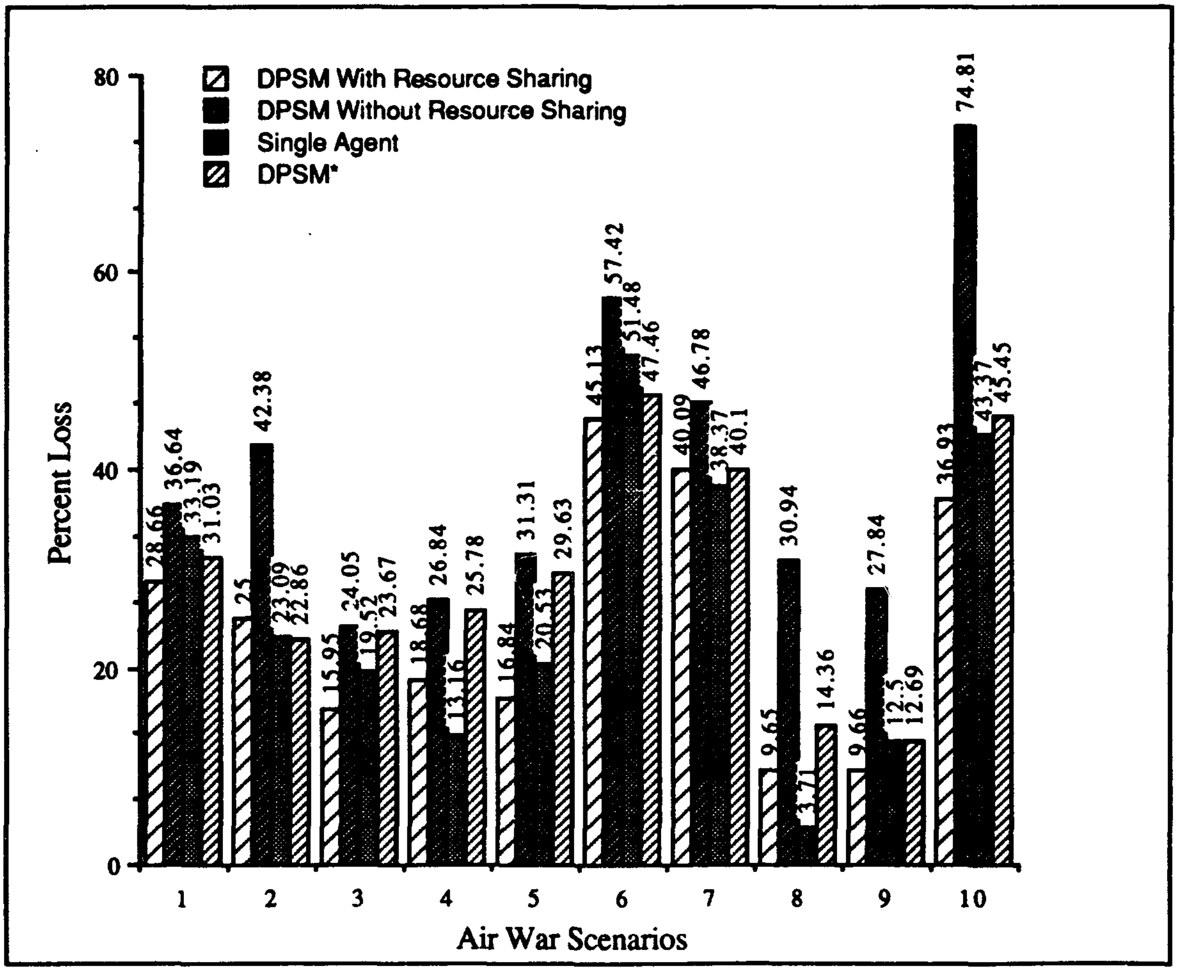

Figure 5.4. Blue aircraft losses (automated approaches).

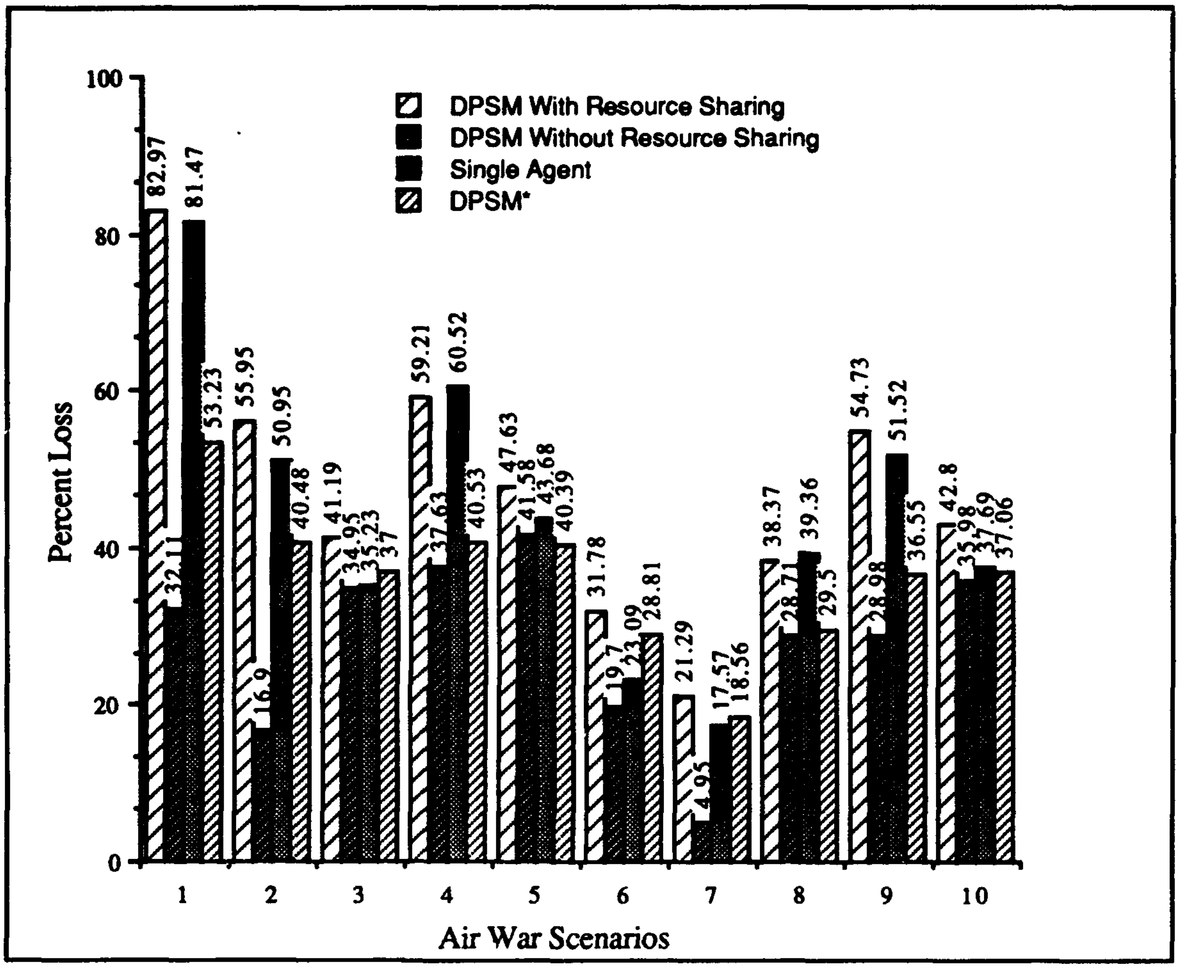

Figure 5.5. Red aircraft losses (automated approaches).

Figure 5.5 shows the percentage of Red Force aircraft losses for each scenario for each of the automated decision making approaches. Except for Scenario 5, DPSM without resource sharing performed the worst, i.e., fewer of the Red Force aircraft were lost using this method than compared to the other three methods. The sharing of resources, or having direct access to all resources, resulted in better allocations of Blue Force aircraft. In this manner, a more adequate force of Blue aircraft could engage the Red Force in combat, and thus, do more damage to the Red Force. The average percentage of Red aircraft lost across all ten scenarios is 28.25% for DPSM without resource sharing, 36.21% for DPSM*, 44.11% for the single agent, and 47.59% for DPSM with resource sharing. In comparing aircraft losses between the Blue Force and the Red Force, the Blue Force lost a greater percentage of aircraft than the Red Force under DPSM without resource sharing in seven scenarios (1, 2, 3, 6, 7, 8, and 10). For both the single agent approach and DPSM*, the Blue Force lost a greater number of aircraft than the Red Force in only three scenarios (6, 7, and 10). Finally, for DPSM with resource sharing, the Blue Force lost more aircraft than the Red Force in just two scenarios (6 and 7).

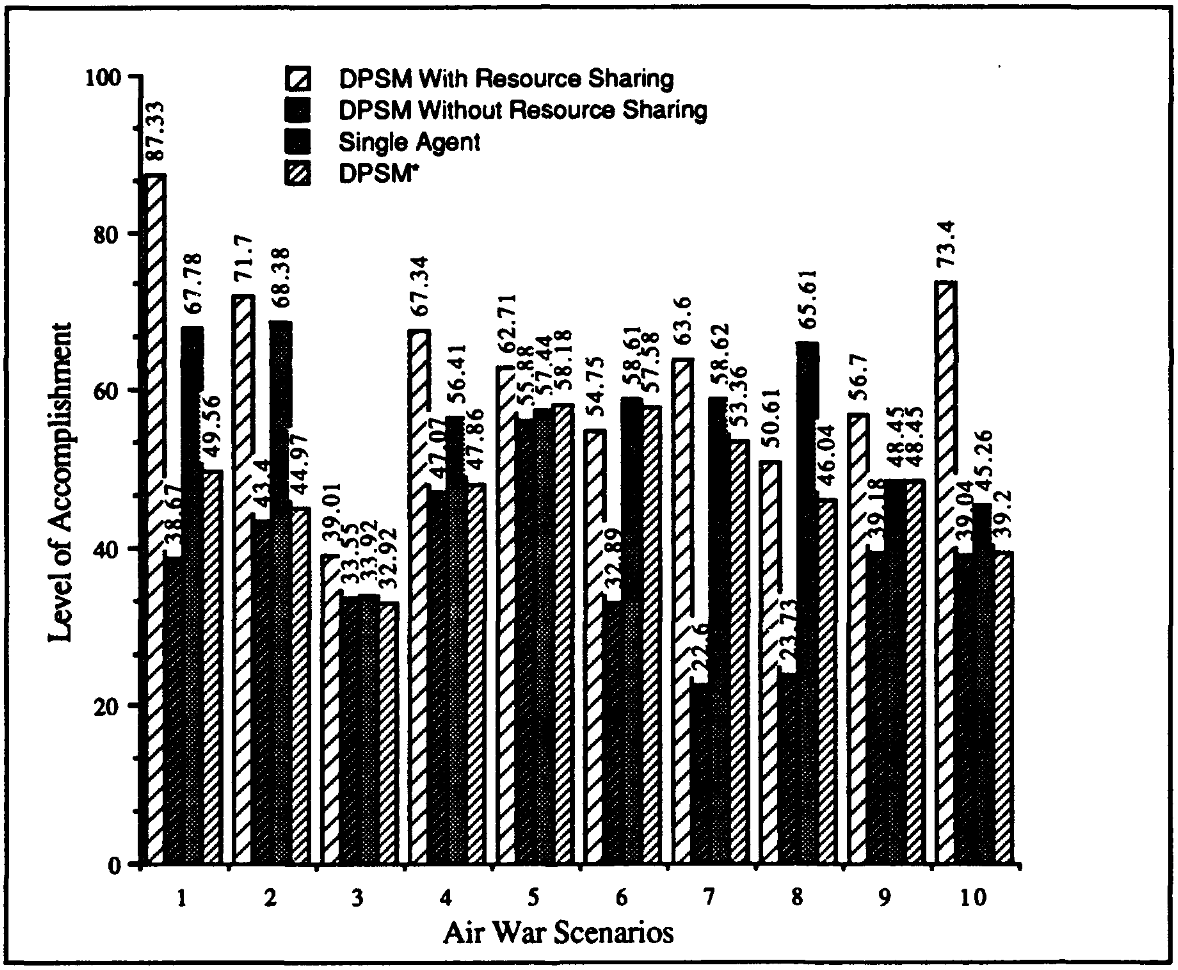

Figure 5.6. Weighted average level of accomplishment (automated approaches).

Figure 5.6 shows the weighted average level of task accomplishment for each scenario for each type of decision making method. In all cases but one (Scenario 3), DPSM without resource sharing had the lowest level of task accomplishment. The ability to share resources, or allocate from the entire inventory of resources, allows for better resource allocation. This in turn, allows for a higher level of task accomplishment. (The more adequate the allocation of resources, the better the ability to accomplish a given task.) The single agent approach did not perform as well overall as the distributed method with resource sharing, except in Scenarios 6 and 8. In most cases, DPSM* had a better level of accomplishment than DPSM without resource sharing, but a lower level than the other two approaches. An analysis of these four approaches is presented in Sections 5.3.2-5.3.4. The level of task accomplishment averaged over the ten scenarios for each technique was 38.20% for DPSM without resource sharing, 47.81% for DPSM*, 56.05% for the single agent approach, and 62.72% for DPSM with resource sharing.

5.2.3 Timing Test

To investigate the amount of time required to solve a group of tasks by each automated method, the following experiment was conducted. Five categories of test cases were developed based upon the number of available tasks to solve. This ranged from having to solve only one task, to having to solve five tasks. Within each category, 30 test cases were randomly generated. Each automated problem solving approach was then run to solve all of the tasks by category, and the time required to solve a given set of tasks was collected. Finally, the average time to solve the tasks by category was computed. This information is displayed in Figure 5.7. For the distributed approaches, four Sun workstations were employed, as in the experiments with the air war scenarios. Similarly, two workstations were used for the single agent method. The data collected was the time between Agent0 sending the task list to the other agents and Agent0 being notified of their solutions.

The graph in Figure 5.7 shows the average amount of time needed to generate a solution for the number of tasks provided. With only one task to solve, the centralized approach with only one agent used the least amount of time. This is due to the fact that the three distributed approaches require additional time for the agents to negotiate the assignment of tasks. With the single agent approach, no negotiation is needed as the one agent gets assigned all available tasks. As the number of tasks increased, the distributed approach without resource sharing provided the smallest amount of time for problem solving. Even though the distributed agents negotiated task assignments, they were able to solve groups of tasks faster than the single agent. Once tasks were assigned, agents could solve tasks concurrently rather than processing them one at a time as the single agent must do. The distributed approach with resource sharing required the most time to solve tasks. This is due to the fact that agents must sometimes re-solve their tasks when other agents borrow resources because of conflicts. With multiple tasks, DPSM* was faster than the single agent and the DPSM with resource sharing, but slower than DPSM without resource sharing. The reason for the latter finding is that DPSM* is still doing *some* resource sharing when the borrowed resources are not needed by other agents for their own tasks.

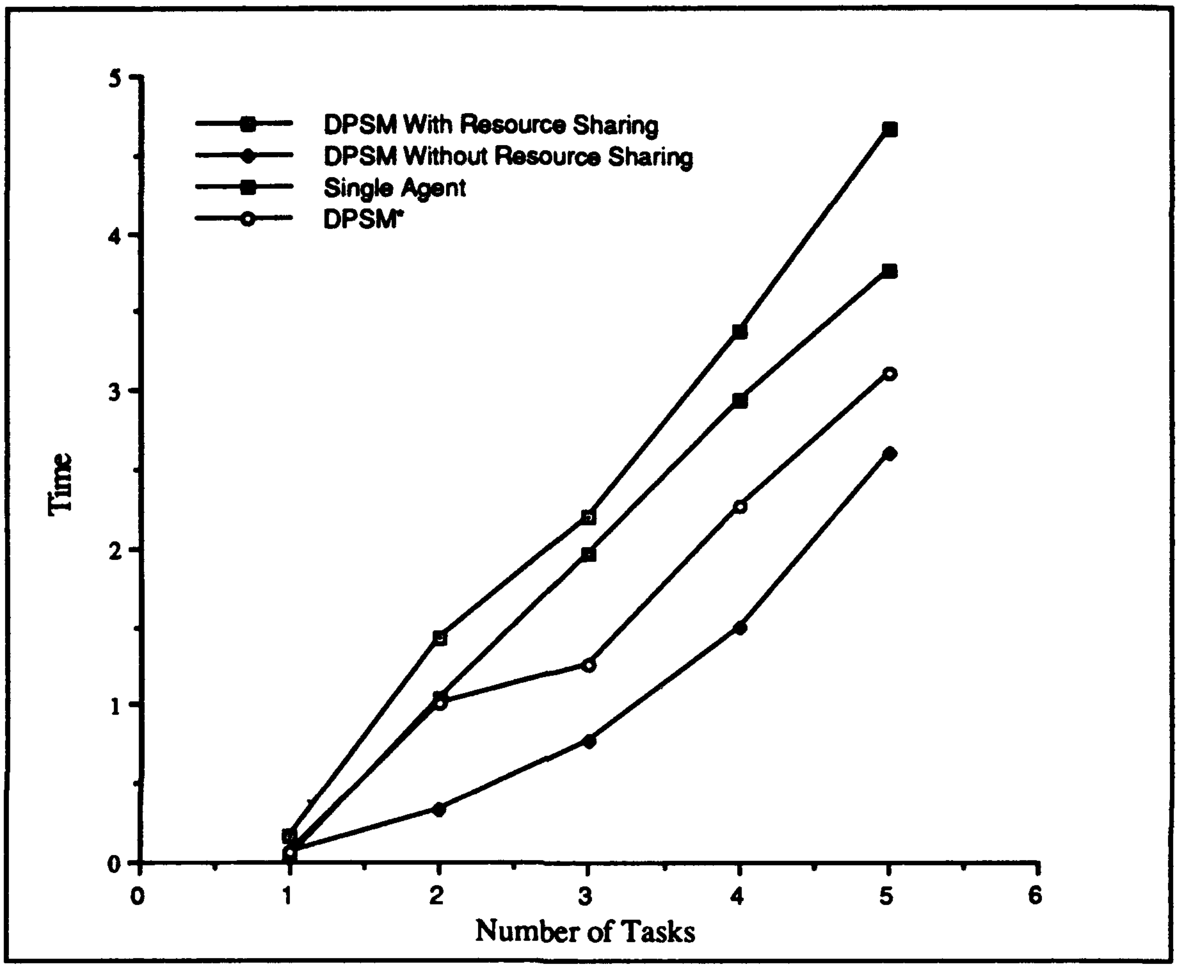

Figure 5.7. Average time to solve a given set of tasks.

5.3 Analysis of Performance and Agent Behavior

5.3.1 Human Decision Makers Compared to the Automated Approaches

The performance of the human decision makers cannot be directly compared to that of the automated approaches because of the "noise" introduced into the data by the simulation of fog and friction. Due to the limited availability of the individuals participating in the experiments, the air war scenarios without the effects of fog and friction could not be repeated with the human decision makers. On the other hand, observing the actions of the individuals during air war simulations helped to identify the types of problems that can occur with human decisions makers.

- Information Overload — When humans must make decisions based upon an abundance of information, errors are often made in the resulting solutions. This fact was exemplified when the human decision makers' task solutions contained errors based upon aircraft fuel status. This resulted in the loss of a large number of aircraft and in a decrease in the ability of the decision makers to accomplish subsequent tasks.

- Miscommunication — When humans communicate with one another, information may be misunderstood or lost. This happened on several occasions during the air war simulations. Information relayed from one individual to another was misinterpreted. When this happened, the result was a less than adequate solution for a task and inefficient use of resources (allocating the wrong aircraft squadron or directing aircraft to the wrong waypoints).

5.3.2 Distributed Agents Without Resource Sharing

When having to solve multiple tasks, distributed agents without resource sharing can develop solutions faster than the other two automated approaches. The drawback with this method, however, becomes evident in those cases when an agent responsible for a task does not have enough resources to completely accomplish that task. Without the ability to borrow resources from other agents, the resulting task solutions may not be as effective as those generated by the other automated approaches. For example, in Scenario 2 the Red Force attacks Blue airfield #2 with a formation of one squadron of F-02 fighters, two squadrons of B-60 bombers, and two squadrons of W-11 Wild Weasels. The task of defending the airfield is assigned to Agent2. At the time of the attack, Agent2 can allocate only one squadrons of F-01s. The ensuing combat engagement resulted in the Red Force losing no aircraft, while the Blue Force lost 11 fighters and airfield #2 sustained 100% damage. When presented with this same task, the other two approaches were able to obtain additional fighter aircraft from other airfields to assist with the defense. In the method with distributed agents and resource sharing, three squadrons of F-01s are used to defend against the attack (two squadrons are borrowed from Agent1). In this case, the combat resulted in the Red Force losing 20 bombers and 5 Wild Weasels. The Blue Force lost 11 fighters but airfield #2 sustained no damage. Similar situations occurred with offensive missions. Without the ability to share resources, the Blue Force, in many offensive missions, could not attack Red targets in sufficient strength to completely destroy the targets. At the beginning of an air war simulation, each agent may possess enough resources to engage the enemy with the firepower required to completely take out a target. After 2-3 combat engagements, however, the agents begin to lose aircraft resources to the degree that, without resource sharing, their effectiveness in accomplishing tasks falls below that of the other two approaches where resources can be pooled.

5.3.3 DPSM*

With DPSM*, agents negotiated for tasks, but only the initial bids were used for the assignment of tasks. In addition, agents shared only those resources which they were not planning to use for their own assigned tasks. This is in contrast to DPSM with resource sharing where, based upon priority, agents with higher priority tasks could "take" any resource required for task accomplishment. The overall results with respect to the ten air war scenarios were that DPSM*'s performance was better than DPSM without resource sharing but below that of the single agent and DPSM with resource sharing (the system resulting from this research). An example taken from Scenario 9 illustrates this difference in performances. At one point in Scenario 9, the Red Force attacks the Blue Force's airfields #2 and #3. In the approaches using task priorities (the DPSM with resource sharing and the single agent), the task of defending airfield #3 has the highest priority. As such, adequate resources are allocated as needed for the proper defense of that airfield, i.e., aircraft were taken from other agents to handle the highest priority task. The outcome is such that airfield #3 is protected while airfield #2 received 100% damage. With the DPSM*, agents 2 and 3 allocated from their own pool of resources the aircraft to defend their respective airfields. Neither agent had enough of their own resources to mount an adequate defense. In addition, neither agent had additional aircraft to share with one another. The end result was that both airfields were destroyed (100% damage). Thus, in one case only one airfield was lost, while in the other case (DPSM*) both airfields were lost.

5.3.4 DPSM With Resource Sharing Versus the Single Agent Approach

The initial expectation with these two approaches was that their performance in solving tasks would be very similar. Each agent in the distributed approach has access to the entire inventory of aircraft resources by their ability to "take" resources from other agents. The single agent method automatically has access to all aircraft resources. Both approaches used the same rule base for resource allocation and the same algorithm for generating flight paths. (The single agent approach simply used the DPSM with one agent.) Even though the two approaches were similar, the distributed method performed better in terms of resource losses and level of accomplishment. The reason for this had to do with the heuristic used for resource sharing.

In the distributed approach, the borrowing of resources from agents can easily propagate additional conflicts. Therefore, to minimize the impact of conflicts, agents borrowed resources only when they could not solve an assigned task using *any* of their own resources. Here is a brief example to illustrate this heuristic. Agent1 has an assigned task which requires the allocation of aircraft with air-to-ground capability. The best aircraft to use for such a mission are bombers. However, Agent1 has only two squadrons of bombers, but to successfully accomplish this task three squadrons are needed. Before taking a bomber squadron from another agent, Agent1 will use a more general resource from its own inventory (such as a fighter bomber squadron). Only if it has no aircraft that can be used, will Agent1 attempt to borrow from another agent. The single agent approach, having access to all the aircraft, will always exhaust the allocation of aircraft with specific capabilities before allocating the more general aircraft. In this example, the distributed agent would use two squadrons of bombers and one squadron of fighter-bombers from its own inventory. The single agent approach would allocate three squadrons of bombers. However, due to the locations of aircraft, the single agent method would normally not be able to accomplish the task in the same amount of time as the distributed approach. By using its own aircraft, Agent1 could reach the intended target faster than the single agent approach. This is because one of the bombers allocated by the single agent may be farther away than the other bombers, and thus, task accomplishment is delayed until the arrival of that bomber squadron. Both approaches would accomplish their task by the deadline, but the distributed approach would accomplish the task before the single agent approach. The impact of arriving at the target later may be that the environment has changed to such a degree that the Blue Force performs poorly in combat. (The Red Force has more resources available and can engage the Blue bombers with more fighters.) This "side effect" occurred often in the experiments, i.e., the single agent method would develop task solutions which took longer to accomplish than the solutions generated by the distributed agents. This would provide the Red Force with more time to prepare for the impending attacks. Additionally, the distributed approach would sometimes develop solutions whereby multiple Red targets were attacked at the same time. This would cause the Red Force to divide its defensive fighters. In many instances, the single agent solutions would involve attacks occurring one after the other. This would allow the Red Force to mass its defenses for each attack, rather than having to divide its forces.

CHAPTER 6

CONCLUSION

6.1 Summary

A major issue of concern in DAI involves coordination and cooperation of distributed problem solving agents. Given a set of tasks and a group of geographically distributed agents, how can the agents as a group assign tasks among themselves? Secondly, under conditions involving limited resources, how can the agents resolve resource conflicts in order to best accomplish a given set of tasks? These two general DAI problems provided the focus for this research.

To solve the first problem, a negotiation technique was developed based upon the basic ideas of the contract net. Agents evaluate their abilities to accomplish tasks and then exchange "bids" with one another. Agents with the best bids exceeding a certain threshold are assigned those associated tasks. Tasks with no best bids are re-negotiated until bids are improved or a level of quiescence is reached. This method of negotiation allows the agents as a group to determine task assignments, rather than relying on one agent as a contractor to assign tasks.

This research modified the basic Contract Net approach for negotiating about tasks. Our approach takes into account the knowledge an agent obtains when tasks are bid. By knowing which tasks were awarded to other agents, an agent may be able to provide a better bid for a task which was not assigned to any other agent. (Should an agent know in advance that it will lose a bid for a particular task then it could provide a better bid for some other task. The use of the threshold value and renegotiation allows agents to do just this — improve their bids for outstanding tasks.) The motivation behind this modification of the Contract Net was to allow for a better distribution of tasks among agents.

Resource sharing occurs in a distributed system when one agent cannot satisfy (accomplish) an assigned task using only the resources under its jurisdiction. To solve such a task, the agent must borrow resources from other agents. A resource conflict can happen when more than one agent attempts to allocate the same limited resource for use during the same time frame. To solve this DAI problem, a technique called Hierarchical Iterative Conflict Resolution was used. With this technique, agents with higher priority tasks (where priority is determined using an algorithmic method) may "take" resources belonging to agents with lower priority tasks. Borrowing resources in this manner may create additional conflicts but at lower priority levels. Conflicts are gradually resolved in this manner; thus, resource conflicts are resolved iteratively, based upon a hierarchy of task priorities. Either all tasks are solved, or if some tasks remain unsolved due to the lack of resources, the unsolved tasks will be those of lowest priority.

In many real-world domains (air war, fire fighting, law enforcement, search & rescue and finance, to name just a few), some tasks may be considered more important than others. For example, in the search & rescue domain, the task of rescuing a person from a capsized vessel is more important than rescuing an individual who is safe in a vessel that has simply run out of fuel. Thus, by prioritizing tasks, we can ensure that the higher priority tasks are given preference for resources. This ensures that the higher priority tasks are accomplished before those of lower priority. With fully cooperative agents solving tasks based upon priority, communications costs may be reduced because the agents will not have to interact as much. Rather than negotiating (requesting) resources from other agents, an agent knows that it may "take" resources from others if it has the higher priority task.

The two techniques were implemented in a DAI testbed. The testbed simulates features of the air war domain which has the necessary characteristics of a distributed problem solving environment — geographically distributed agents (Wing Commanders), limited resources (aircraft squadrons), and distributed tasks (air missions). Empirical studies were conducted using the testbed. The studies consisted of air war simulations between two opposing forces. Tasks involved air missions in which limited aircraft resources had to be allocated by geographically separated agents. The experiments examined the performance of different decision making entities: human decision makers, distributed computer agents with resource sharing, distributed computer agents without resource sharing, distributed computer agents without the techniques described in Chapter 3 (DPSM*), and a single computer agent. In these experiments, the performance of the distributed agents without resource sharing was the worse (in terms of resource losses and level of task accomplishment). Without the ability to share resources, agents could not allocate adequately the resources needed to accomplish many tasks. Distributed agents with resource sharing performed the best. Using Hierarchical Iterative Conflict Resolution, agents could share resources to ensure that the higher priority tasks were accomplished before those of lesser priority. The single agent approach would attempt to always allocate the best resources for a given task. In many cases, however, it would allocate a resource located far from the task. This would result in increasing the amount of time for task accomplishment, and also allowed the opposing force more time to defend against offensive attacks. Thus, the performance of the single agent fell between that of distributed agents with resource sharing and distributed agents without resource sharing. The approach with DPSM* performed better than distributed agents without resource sharing, but worse than the other two techniques. In this approach, agents shared only those resources for which they did not plan to use for their own assigned tasks. Finally, the experiments with human decision makers served to re-enforce the reasons behind human errors in problem solving — information overload and miscommunication.

6.2 Specific Accomplishments

The Distributed Air War testbed is a flexible system for investigating the behavior and performance of distributed problem solving agents within a volatile environment. As part of this research, the following significant accomplishments have been achieved:

- Implementation of an effective method for allocating tasks among a group of intelligent computer agents. This negotiation technique builds upon the idea of a contract net but allows agents to become self-appointed task commanders (responsible for a task). By exchanging quality measures with one another, a group of agents can know which of the agents is the best for handling a particular task. In addition, it is ensured that only agents with the best "bids" exceeding a threshold value will become task commanders. The use of the threshold value lets agents re-evaluate and re-negotiate quality measures for unassigned tasks in an attempt to improve previous bids. It also helps in overcoming the difficulties that would be caused by decisions based on inaccurate assessments.

- Implementation of a method for resolving resource conflicts among a group of distributed agents. This technique, called Hierarchical Iterative Conflict Resolution, provides a systematic method for determining which tasks should be allocated the limited resources. Resource conflicts are resolved iteratively based upon a hierarchy of task priorities. This allows higher priority tasks to be accomplished before those of lower priority.

- Construction of a testbed for the study of distributed problem solving within a dynamic, volatile environment. The testbed contains a Distributed

Problem Solving Module which employs the techniques described in this research for agent coordination and cooperation. Furthermore, the testbed provides a simulated air war environment in which various problem solving situations (air war scenarios) may be created in a high-level manner, in order to study the behavior and performance of intelligent agents. Finally, the testbed has a graphical user interface which allows the user to easily interact with and monitor the system.

- Performed empirical studies to evaluate the distributed problem solving concepts examined in this research.

The testbed was designed with flexibility and robustness in mind. The three main modules (UIM, BSM, and DPSM) can operate independently of one another. This aspect of the testbed also allows the various parts to be modified independently of the other parts. For example, a command line, text-based user interface could be developed to replace the current graphical user interface for use on systems without a graphics capability. Similarly, the BSM could be replaced with a different simulator module or the DPSM with a different problem solving module. Further, external files contain the various parametric values for use in the testbed. By editing these files, a user can change such features as aircraft characteristics, weighting factors, target attributes, and so forth.

The results of the empirical studies conducted in this research has provided the following information:

- Distributed agents can solve tasks independently of each other, within a volatile domain in which resources can be lost. However, the ability of sharing resources will improve the level of task accomplishment. The techniques for negotiating task assignments and Hierarchical Iterative

Conflict Resolution promote inter-agent coordination and cooperation in such an environment. In the ten air war scenarios simulated in this research, distributed problem solving without resource sharing had an average level of accomplishment of 38.20%, DPSM* had an average level of accomplishment of 47.81%, and distributed problem solving with resource sharing had an average level of accomplishment of 62.72%.

- A centralized approach using a single agent with access to all resources can ensure that the most appropriate resources are allocated for tasks. However, as seen in the studies, the locations of resources and tasks can affect the timing of task accomplishment. This, in turn, can impact the level of accomplishment within a dynamically changing environment. In the ten air war scenarios, the single agent approach averaged a level of accomplishment of 56.05%.

- The experiments with human decision makers had an average level of accomplishment of 50.62%. These experiments served to show how information overload and miscommunication can easily result in human errors during the problem solving process.

- In terms of resource losses, distributed agents with resource sharing performed the best. As shown in Table 6.1, the distributed approach with resource sharing minimized its (Blue Force) losses and maximized the Red Force losses better than the other three automated approaches.

Table 6.1 Resource Losses Averaged Over the Ten Scenarios

Method	Blue Aircraft Lost	Red Aircraft Lost
DPSM With Resource Sharing	24.65%	47.59%
Single Agent	25.89%	44.11%
DPSM*	29.30%	36.21%
DPSM Without Resource Sharing	39.90%	28.25%

- Concerning the time required to solve tasks, the single agent approach is faster when only one task is given. As the number of tasks increase, the use of distributed agents enhances the solution of a group of tasks faster than a single agent (distributed agents without resource sharing versus a single agent). Finally, since resolving conflicts may result in the need to re-do the solutions for some tasks, distributed agents with resource sharing may require more time to solve a group of tasks, as compared to a single agent. If possible, the need for resource sharing should be kept to a minimum. However, in environments such as an air war, resource losses cannot be predicted with certainty. As more tasks are accomplished, more resources may be lost, and hence, the need for resource sharing may have to increase for subsequent tasks.

Resource conflicts over time and space are common features in most domains. The techniques for promoting multi-agent cooperation used in this research provide a methodical way of sharing resources to ensure that the more important tasks are accomplished before those of lesser importance (priority). These techniques are appropriate for those environments in which the following are true:

- Agents are fully cooperative.

- Tasks have varying degrees of importance and may be prioritized in some manner.

- The expectation for the loss of resources is high and, hence, resources must be shared to a large degree to accomplish tasks.

- It is desirable to minimize the interaction (communication) between agents.

- It is necessary to consider changes in both factors of the priority function; i.e., not only does the urgency level change monotonically but the importance level may need to be updated whenever the environment changes or more/better information about it becomes available.

Besides their intuitive appeal, the techniques described in Chapter 3 performed better overall than the other three automated approaches for the ten air war scenarios. The reasons for better performance include:

- Better assignment of tasks to agents. DPSM* (the distributed approach without the techniques introduced in this research) assigned tasks to agents after only the initial bids were provided. On the other hand, the DPSM with resource sharing allowed bids to be renegotiated, whereby agents could improve bids for outstanding tasks based upon the knowledge about previous bids.

- Use of a priority scheme. This ensured the accomplishment of significant (higher priority) tasks before those of lower priority.

- Use of a systematic approach for resource sharing. This allowed agents to "take" resources based upon priority to allow efficient accomplishment of a given set of tasks.

6.3 Contributions

This research has produced contributions to both the DAI research community and the military community. For the former, the following has been accomplished:

- *An examination of an approach for multi-agent coordination and cooperation:* The general problem is two-fold. Within a distributed environment (geographically separated agents, resources and tasks), how do

  - *m* agents assign and distribute a set of *n* tasks among themselves so that the "best" match is between agents and tasks?

  - *m* agents allocate a set of *n* resources to a set of *o* tasks? The resources are associated with the agents but they may or may not be shared (borrowed).

  This research has employed techniques for resolving the above issues in multi-agent coordination and cooperation. These ideas were studied with reference to an air war domain, but the basic ideas are general enough to be applied to other homomorphic distributed domains as well.

- *Empirical studies using the approach:* The empirical studies show the benefit obtained when agents are allowed to share resources with one another. While a single agent approach also provided better results than distributed agents without resource sharing, its level of accomplishment was lower when compared to the distributed approach with resource sharing. One must also bear in mind that the single agent approach has a major drawback in a volatile environment — single point of failure, that is a low level of robustness. If the single agent fails, no tasks can be solved. With the distributed approach, the failure of one agent may slow down task accomplishment, but it does not stop the accomplishment of all tasks.

- *A testbed for conducting studies in multi-agent coordination and cooperation:* The flexible testbed constructed as a part of this research provides a tool which can be used for further studies on issues of multi-agent decision making in dynamic, volatile environments.

For the military community, the following has been contributed:

- *A distributed approach for allocating aircraft resources during an air war:* Current Air Force research in this area has focused upon centralized techniques concerning air war planning. This research has provided an avenue to examine a distributed approach.

- *A research tool:* The testbed may be used by the Air Force to conduct studies on resource allocation within the air war environment. As previously stated, the testbed has the flexibility to be easily modified to meet varying requirements.

- *An educational tool:* The testbed itself may be used as an educational tool for teaching aspects of airpower strategy and doctrine. It allows users to create "what if" situations to examine the outcomes of combat under varying conditions of the air war environment.

6.4 Future Research

In the empirical studies conducted, both forces possessed full knowledge of the opponent's capabilities (resources but not intended actions). In real life, this information about the opposing force may be only partially available, or known with less than 100% certainty. The testbed could be modified to model this aspect of uncertainty for further studies. This will also necessitate modifying the rule base in the DPSM to deal with information that may not be known with complete accuracy. Fuzzy logic [Zadeh, 1983] and/or a truth maintenance system [Doyle, 1979] could be introduced in the knowledge and rule base to work with uncertain information.

As resources are depleted in a distributed environment, the need for resource sharing rises. At some point, it would be beneficial to re-deploy resources (reassign control) among the various agents. Studies could be conducted to determine an approach for deciding when and how resources should be re-deployed to minimize the need for resource sharing.

At present, the testbed allows one force to be controlled by a set of networked computers (intelligent agents), while the other force must be under human control. The testbed could be modified to allow both forces to operate under computer control. This would allow simulations in which different decision making strategies are under computer control for both forces. Also, the simulations could then be processed faster without the need for human user input.

REFERENCES

Abrett, G., Deutsch, S., & Downes-Martin, S. (1990, September). Two AI languages for simulation. Transactions of the Society for Computer Simulation, 7, 229-250.

Anderson, M. & Granger, A. (1985, Fall). Target area mission planning. Fighter Weapons Review, pp. 1-7.

Andriole, S. J. & Hopple, G. W. (1984). Tactical planning and replanning. In S. J. Andriole & G. W. Hopple (Eds.), Defense Applications of Artificial Intelligence: Progress and Prospects (pp. 225-242). Lexington, MA: Lexington Books.

Applegate, C., Elsaesser, C. & Sanborn, J. (1990). An architecture for adversarial planning. IEEE Transactions on Systems, Man, and Cybernetics, 20, 186-194.

Azarewicz, J., Falla, G., Hayslip, I. & Heithecker, C. (1987). Multi-agent plan recognition in an adversarial domain. Proceedings of the Third Annual Expert Systems in Government Conference, 188-193.

Beam, Walter R. (1986). Automated support systems in military command and control: The problem. Proceedings of the 1986 Expert Systems in Government Symposium, 357-361.

Benoit, J. R., Bonasso, R. P., & Davidson, J. R. (1986). ALLIES: An experiment in cooperating expert systems for command and control. Proceedings of the 1986 Expert Systems in Government Symposium, 372-380.

Benoit, J. W., Elsaesser, C., MacMillan, T. R. & Tallis, H. C. (1990). Planning for Conflicts in Multi-Agent Domains (Technical Report No. MTR-90W00149). Washington DC: MITRE Corp.

Bonasso, R. P. (1988, Fall). What AI can do for battle management. The AI Magazine, pp. 77-83.

Cammarata, S., McArthur, D. & Steeb, R. (1983). Strategies of cooperation in distributed problem solving. Proceedings of the Eighth International Joint Conference on Artificial Intelligence, 767-770.

Casavant, T. L. & Kuhl, J. G. (1990). A communicating finite automata approach to modeling distributed computation and its application to distributed decision-making. IEEE Transactions on Computers, 39, 628-639.

Charniak, E. & McDermont, D. (1985). Introduction to artificial intelligence. Reading, MA: Addison-Wesley.

Corkill, D. D. & Lesser, V. R. (1983). The use of meta-level control for coordination in a distributed problem solving network. Proceedings of the Eighth International Joint Conference on Artificial Intelligence, 748-756.

Decker, K. S. (1987). Distributed problem-solving techniques: A survey. IEEE Transactions on Systems, Man and Cybernetics, 17, 729-740.

Demazeau, Y. & Muller, J. P. (1990). Decentralized artificial intelligence. In Y. Demazeau & J. P. Muller (Eds.), Decentralized A. I.: Proceedings of the First European Workshop on Modelling Autonomous Agents in a Multi-Agent World (pp. 3-13). Netherlands: Elsevier Science.

Doyle, J. (1979). A truth maintenance system. Artificial Intelligence, 12(3), 231-272.

Durfee, E. H. & Montgomery, T. A. (1990). A hierarchical protocol for coordinating multiagent behaviors. Proceedings of the National Conference on Artificial Intelligence, 86-93.

Dodhiawala, R., Sridharan, N. S., Raulefs, P. & Pickering, C. (1989). Real-time AI systems: A definition and an architecture. Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, 256-261.

Downes-Martin, S., Deutch, S., & Abrett, G. (1992). Managing uncertainty in time-critical plan evaluation. International Journal of Man-Machine Studies, 36, 337-356.

Dunnigan, J. F. (1988). How to make war. New York: Quill.

Durfee, E. H. & Lesser, V. R. (1988). Incremental planning to control time-constrained blackboard-based problem solver. IEEE Transactions on Aerospace and Electronic Systems, 24, 647-662.

Engelmore, R. S., Morgan, A. J., & Nii, H. P. (1988a). Introduction. In R. S. Engelmore & A. J. Morgan (Eds.), Blackboard Systems (pp. 1-22). Workingham, England: Addison-Wesley.

Engelmore, R. S., Morgan, A. J., & Nii, H. P. (1988b). Hearsay-II. In R. S. Engelmore & A. J. Morgan (Eds.), Blackboard Systems (pp. 25-29). Workingham, England: Addison-Wesley.

Entner, R. S. & Tosh, D. E. (1986). Expert system architectures for battle management. Proceedings of the 1986 Expert Systems in Government Symposium, 16-21.

Erman, L. D., Hayes-Roth, F., Lesser, V. R., & Reddy, D. R. (1988). The Hearsay-II speech-understanding system: Integrating knowledge to resolve uncertainty. In R. S. Engelmore & A. J. Morgan (Eds.), Blackboard Systems (pp. 31-86). Workingham, England: Addison-Wesley.

Findler, N. V. & Lo, R. (1986). An examination of distributed planning in the world of air traffic control. Journal of Parallel and Distributed Computing, 3, 213-231.

Findler, N. V. & Lo, R. (1991). A distributed artificial intelligence approach to air traffic control. IEE Proceedings Part D, Control Theory and Applications, 138, 515-524.

Findler, N. V. (1990). Contributions to a computer-based theory of strategies. Berlin: Springer-Verlag.

Firby, R. J. (1987). An investigation into reactive planning in complex domains. Proceedings of the Sixth National Conference on Artificial Intelligence, 202-206.

Fraichard, T. & Demazeau, Y. (1990). Motion planning in a multi-agent world. In Y. Demazeau & J. P. Muller (Eds.), Decentralized A. I.: Proceedings of the First European Workshop on Modelling Autonomous Agents in a Multi-Agent World (pp. 137-153). Netherlands: Elsevier Science.

Gadsden, J.A. (1988). Knowledge-based planning and replanning in Naval command and control. The IEEE Fourth Conference on Artificial Intelligence Applications, 286-292.

Galliers, J. R. (1990). The positive role of conflict in cooperative multi-agent systems. In Y. Demazeau & J. P. Muller (Eds.), Decentralized A. I.: Proceedings of the First European Workshop on Modelling Autonomous Agents in a Multi-Agent World (pp. 33-46). Netherlands: Elsevier Science.

Gasser, L., Bragamza, C. & Herman N. (1987). MACE: A flexible testbed for distributed AI research. In M. N. Huhns (Ed.), Distributed Artificial Intelligence (pp. 119-152). Los Altos, CA: Pitman, Morgan, Kaufman.

Gasser, L. & Huhns, M. N. (1989). Distributed Artificial Intelligence Vol. II. Los Altos, CA: Pitman, Morgan, Kaufman.

Gasser, L. (1991). Social conceptions of knowledge and action: DAI foundations and open systems semantics. Artificial Intelligence, 47, 107-138.

Ge, Q. & Findler, N. V. (1988). Distributed planning and control for manufacturing operations. Proceedings of the 2nd Australian Joint Artificial Intelligence Conference, 432-448.

Graham, M. H. (1992). Issues in real-time data management. Real-Time Systems, 3, 185-202.

Hewitt, C. (1991). Open information systems semantics for distributed artificial intelligence. Artificial Intelligence, 47, 79-106.

Hopple, G. W. (1984). Knowledge-based applications in military problem solving. In S. J. Andriole & G. W. Hopple (Eds.), Defense Applications of Artificial Intelligence: Progress and Prospects (pp. 61-75). Lexington, MA: Lexington Books.

Huang, J., Stankovic, J.A., Ramamritham, K., & Towsley, D. (1989). Experimental evaluation of real-time transaction processing. IEEE Real-Time Systems Symposium, 144-153.

Huhns, M. N. (1987). Distributed Artificial Intelligence. Los Altos, CA: Pitman, Morgan, Kaufman.

Krieger, C. R. (1987, Summer). Fighting the air war: A wing commander's perspective. Airpower Journal, 21-31.

Laskowski, S. J. & Hofmann, E. J. (1987). Script-based reasoning for situation monitoring. Proceedings of the Sixth National Conference on Artificial Intelligence, 819-823.

Lakin, W. L., Miles, J. A. H., & Byrne, C. D. (1988). Intelligent data fusion for Naval command and control. In R. S. Engelmore & A. J. Morgan (Eds.), Blackboard Systems (pp. 443-458). Workingham, England: Addison-Wesley.

Leedom, D. K. (1984). Knowledge representation in artificial intelligence: Moving AI technology to the organizational level. In S. J. Andriole & G. W. Hopple (Eds.), Defense Applications of Artificial Intelligence: Progress and Prospects (pp. 61-75). Lexington, MA: Lexington Books.

Lehner, P. E. (1989). Artificial intelligence and national defense: Opportunity and challenge. Blue Ridge Summit, PA: TAB Books.

Lehner, P. E. & Adeleman, L. (1984). Senior battle staff decision aiding: A case study. In S. J. Andriole & G. W. Hopple (Eds.), Defense Applications of Artificial Intelligence: Progress and Prospects (pp. 153-166). Lexington, MA: Lexington Books.

Lehnert, K. J, & Sullivan, M. (1989). Development and evaluation of artificial intelligence techniques for tactical decision support systems. Proceedings of the Annual AI Systems in Government Conference, 133-140.

Lesser, V. R. & Erman, L. D. (1988). A retrospective view of the Hearsay-II architecture. In R. S. Engelmore & A. J. Morgan (Eds.), Blackboard Systems (pp. 87-121). Workingham, England: Addison-Wesley.

Lesser, V. R. & Corkill, D. D. (1983, Fall). The distributed vehicle monitoring testbed: A tool for investigating distributed problem solving networks. The AI Magazine, pp. 15-33.

Lizotte, M. & Moulin, B. (1990). A temporal planner for modelling autonomous agents. In Y. Demazeau & J. P. Muller (Eds.), Decentralized A. I.: Proceedings of the First European Workshop on Modelling Autonomous Agents in a Multi-Agent World (pp. 121-136). Netherlands: Elsevier Science.

Lo, R. (1988). A distributed planning system for air traffic control. Unpublished doctoral dissertation, Arizona State University.

Lo, R. & Findler, N. V. (1991). A distributed approach to air traffic control — I. theoretical and experimental studies. Manuscript submitted for publication.

Martial, F. (1990). Interactions among autonomous planning agents. In Y. Demazeau & J. P. Muller (Eds.), Decentralized A. I.: Proceedings of the First European Workshop on Modelling Autonomous Agents in a Multi-Agent World (pp. 105-119). Netherlands: Elsevier Science.

Masui, S., McDermont, J. & Sobel, A. (1983). Decision making in time-critical situations. Proceedings of the Eighth International Joint Conference on Artificial Intelligence.

Nii, H. P. (1986a, Summer). Blackboard systems: The blackboard model of problem solving and the evolution of blackboard architectures. The AI Magazine, pp. 38-53.

Nii, H. P. (1986b, Fall). Blackboard systems: The blackboard model of problem solving and the evolution of blackboard architectures (part two). The AI Magazine, pp. 82-106.

Ow, P. S., Smith, S. F. & Thiriez, A. (1988). Reactive plan revision. Proceedings of the Seventh National Conference on Artificial Intelligence, 77-82.

Parunak, H. V. D. (1990). Distributed AI and manufacturing control: Some issues and insights. In Y. Demazeau & J. P. Muller (Eds.), Decentralized A. I.: Proceedings of the First European Workshop on Modelling Autonomous Agents in a Multi-Agent World (pp. 81-101). Netherlands: Elsevier Science.

Pearson, G. (1988). Mission planning within the framework of the blackboard model. In R. S. Engelmore & A. J. Morgan (Eds.), Blackboard Systems (pp. 433-442). Workingham, England: Addison-Wesley.

Peterson, J. L. & Silberschaty, A. (1985). Operating Systems Concepts (2nd Ed.). Massachusetts: Addison-Wesley.

Ramamritham, K. & Stankovic, J. A. (1989). Distributed scheduling of tasks with deadlines and resource requirements. IEEE Transactions on Computers, 38, 1110-1123.

Rich, E. & Knight, K. (1991). Artificial intelligence (2nd Ed.). New York: McGraw-Hill.

Schalkoff, R. J. (1990). Artificial Intelligence: An Engineering Approach. New York: McGraw-Hill.

Schmidt, C. F., Marsella, S. C., Goodson, J. L. & Bresina, John L. (1989). Reactive planning using a 'situation space'. Proceedings of the Annual AI Systems in Government Conference, 50-55.

Sengupta, U. (1991). A distributed planning system for dynamic resource allocation. Unpublished doctoral dissertation, Arizona State University.

Shekhar, S. & Dutta, S. (1989). Minimizing response times in real time planning and search. Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, 238-242..

Silverman, B. G., Chang, J. S.& Feggos, K. (1989). Blackboard system generator (BSG): An alternative distributed problem-solving paradigm. IEEE Transactions on Systems, Man and Cybernetics, 19, 334-355.

Smith, R. G. (1980). The contract net protocol: high-level communication and control in a distributed problem solver. IEEE Transactions on Computers, 29, 1104-1113.

Smith, S. F. & Ow, P. S. (1985). The use of multiple problem decompositions in time constrained planning tasks. Proceedings of the Ninth International Joint Conference on Artificial Intelligence, 1013-1015.

Stankovic, J. A. & Zhao, W. (1988). On real-time transactions. SIGMOD Record 1, 4-18.

Stapp, J. M. (1990). The use of distributed processors in traffic engineering signals to optimize traffic flow. Unpublished master's thesis, Arizona State University.

Steeb, R., Cammarata, S., Hayes-Roth, F. A., Thorndyke, P. W. & Wesson, R. B. (1981). Distributed intelligence for air fleet control (Rand Report No. R-2728-ARPA). Los Angeles: Rand Corp.

Steels, L. (1990). Cooperation between distributed agents through self-organization. In Y. Demazeau & J. P. Muller (Eds.), Decentralized A. I.: Proceedings of the First European Workshop on Modelling Autonomous Agents in a Multi-Agent World (pp. 175-196). Netherlands: Elsevier Science.

Stuart, C. (1985). An implementation of a multi-agent plan synchronizer. Proceedings of the Ninth International Joint Conference on Artificial Intelligence, 1031-1033.

Sycara, K. P. (1989). Argumentation: Planning other agents' plans. Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, 517-523.

Sycara, K., Roth, S., Sadeh, N. & Fox, M. (1990). An investigation into distributed constraint-directed factory scheduling. Proceedings of the IEEE Sixth Conference on Artificial Intelligence Applications, 94-100.

Tachmindji, A. J. & Bonasso, R. P. (1984). Airland battle management. In S. J. Andriole & G. W. Hopple (Eds.), Defense Applications of Artificial Intelligence: Progress and Prospects (pp. 167-188). Lexington, MA: Lexington Books.

United States Air Force. (1992). Basic aerospace doctrine of the United States Air Force (Air Force Manual 1-1). Washington, DC: U. S. Government Printing Office.

United States Air Force Academy. (1990). Airpower theory and doctrine (Course No. PMS-220). Colorado Springs, CO: United States Air Force Academy.

Warden, J. A. III (1988). The air campaign: planning for combat. Washington, DC: National Defense University.

Werner, E. (1990). Distributed cooperation algorithms. In Y. Demazeau & J. P. Muller (Eds.), Decentralized A. I.: Proceedings of the First European Workshop on Modelling Autonomous Agents in a Multi-Agent World (pp. 17-31). Netherlands: Elsevier Science.

Wood, S. (1983), Dynamic world simulation for planning with multiple agents. Proceedings of the Eighth International Joint Conference on Artificial Intelligence, 69-71.

Zadeh, L. A. (1983, October). Commonsense knowledge representation based on fuzzy logic. Computer, 61-65.

APPENDIX A

DISTRIBUTED AIR WAR TESTBED USERS MANUAL

A.1 Introduction

The Distributed Air War (DAW) testbed provides a flexible environment for the study of problem solving for resource allocation for air wars. The testbed consists of three main components: a graphical user interface, a battle simulator, and a distributed problem solver. This manual describes how to set-up and execute air war simulations using the testbed.

A.2 System Requirements

The testbed operates on Sun workstations with X-windows (Sun's Open Windows). If distributed agents will be used, then one Sun workstation per agent is required. These workstations must be networked together. The directory from which the testbed will be run should contain the following files:

aw	The executable program for the testbed
aircraft.dat	Aircraft characteristics data file
blue.ac	Blue force aircraft squadrons data file
blue.tgt	Blue force land-based resources data file
blue.sam	Blue force SAMs data file
red.ac	Red force aircraft squadrons data file
red.tgt	Red force land-based resources data file
red.sam	Red force SAMs data file

config.dat	Network configuration data file
sides.dat	Configuration file to indicate the workstations upon which each force operates
cycle.dat	Simulation cycle data file (keeps track of cycle numbers)
weights.dat	Weighting factors configuration file
rules.clp	CLIPS rule-base used by the problem solving module

The blue force and red force data files need not be present to start the testbed. If those files are absent, however, the user must go through the set-up and deploy operations to create the files.

A.3 Testbed Operation

To start the testbed, at the system prompt simply type:

aw

After system initialization, the user interface screen is displayed. This consists of the air war map, which fills most of the computer screen, and a menu bar at the top of the map. The map is a standard hexagonal grid as used with military simulation systems. The columns are lettered from A to KK and the rows are numbered from 1 to 51. Thus, map locations are identified by a letter-number combination, such as K3. Blue and red icons representing aircraft and land-based resources may be displayed on the map as well. (The absence of the blue force and red force data files will result in an empty map being displayed.) Figure A.1 shows a portion of this screen as presented to the user.

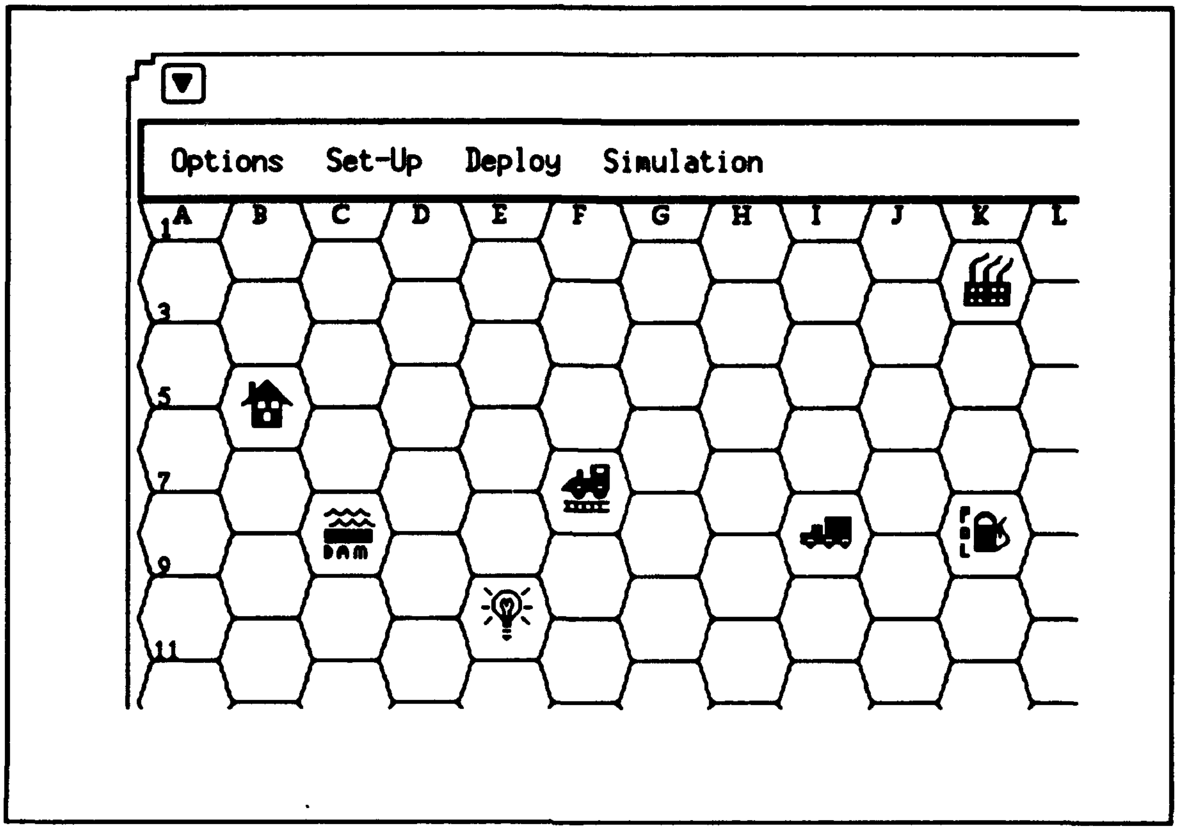

Figure A.1. Main user interface.

The user interface is mouse driven. To select a function from the menu bar, simply move the mouse so that the pointer (arrow) is on the appropriate function name. Next, click the left-most mouse button. This will result in a pull-down menu being displayed. Use the same process to make a selection from the pull-down menu.

A.3.1 Options Menu

The Options Menu is shown in Figure A.2. This menu provides functions for setting the mode of the testbed, toggling the use of sound effects, and quitting the system. Clicking on *Mode* in this menu allows you to change the operating mode of the testbed. Figure A.3 shows the Mode Selection window which appears when this choice is selected.

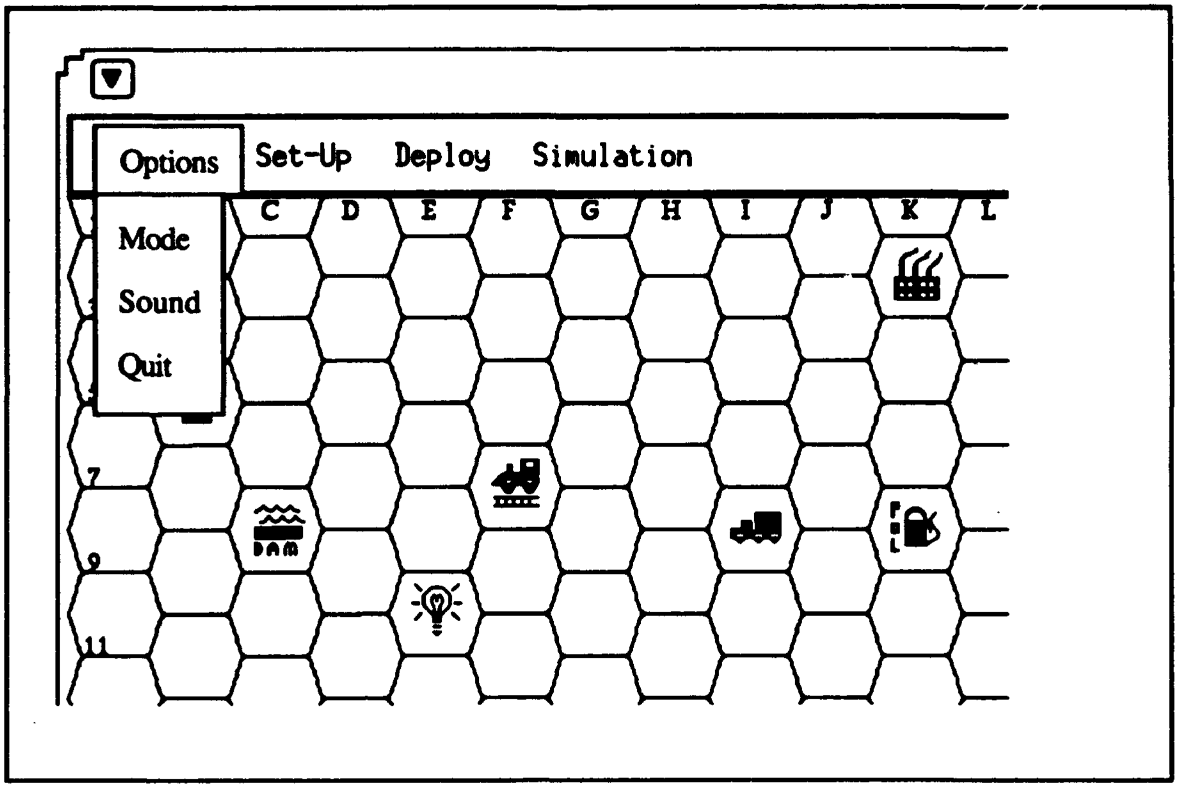

Figure A.2. Options menu.

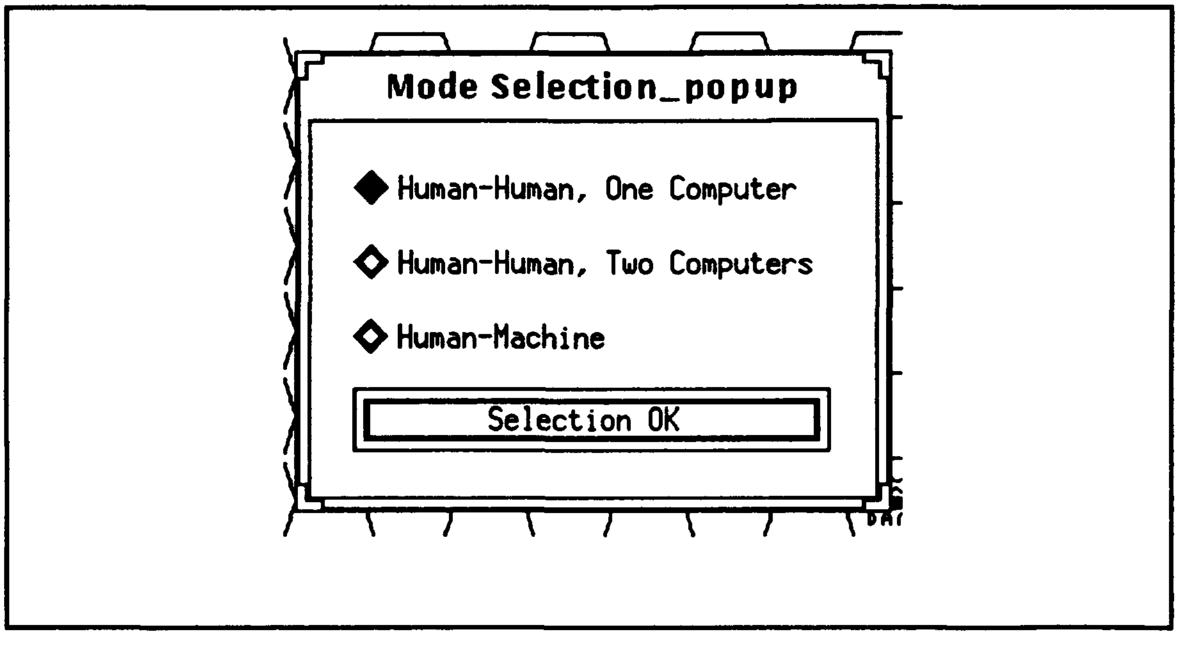

Figure A.3. Mode selection popup window.

To specify the desired mode, the user need only click on the appropriate diamond-shaped radio button. Then after setting the mode, click on the *Selection OK* button. As can be seen by Figure A.3, the testbed can operate in one of three modes:

Mode 0: Human-human, one computer. In this mode, decisions concerning the actions of the blue and red forces are determined by human users. During air war simulations, actions for these forces are input by the human users on one workstation. The system will prompt for input for each force as necessary.

Mode 1: Human-human, two computers. In this mode, one human user controls the actions of the blue force using one workstation, while another user controls the red force using a separate workstation. The computers communicate with each other as needed to exchange data during simulations.

Mode 2: Human-machine. In this mode, a human user controls the actions of one force, while the distributed problem solving module of the testbed makes decisions for the other force. The human operates on one workstation, while the problem solving module is run on one or more other workstations.

Choosing *Sound* from the Options Menu allows you to turn on and off sound effects. If the sound is turned on, then during simulation of combat, appropriate sound effects are played on the workstation. Figure A.4 shows the Sound Selection window. Choices in this window are made in the same manner as the Mode Selection window. Finally, selecting *Quit* from the Options Menu will let you exit the program and terminate your session with the testbed.

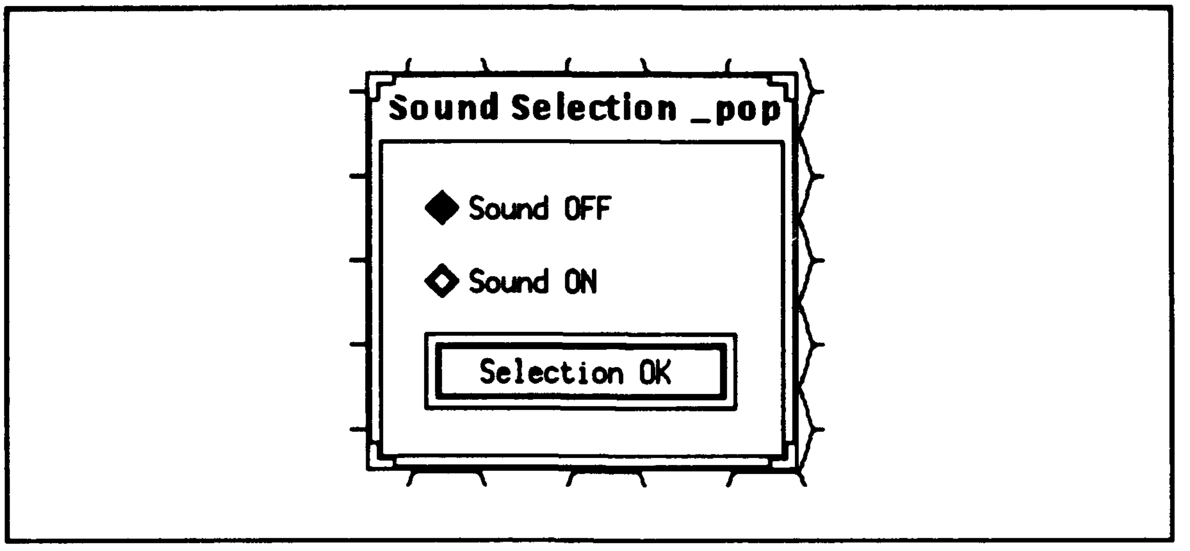

Figure A.4. Sound selection window.

A.3.2 Establishing Resources

The *Set-Up* function, shown in Figure A.5, enables you to create aircraft and Surface-to-Air Missile (SAM) units to be used during simulations. Simply select the appropriate choice from this menu to establish the desired resources. Once these resources have been set-up, you will need to use the Deploy function to place them at their initial positions on the map.

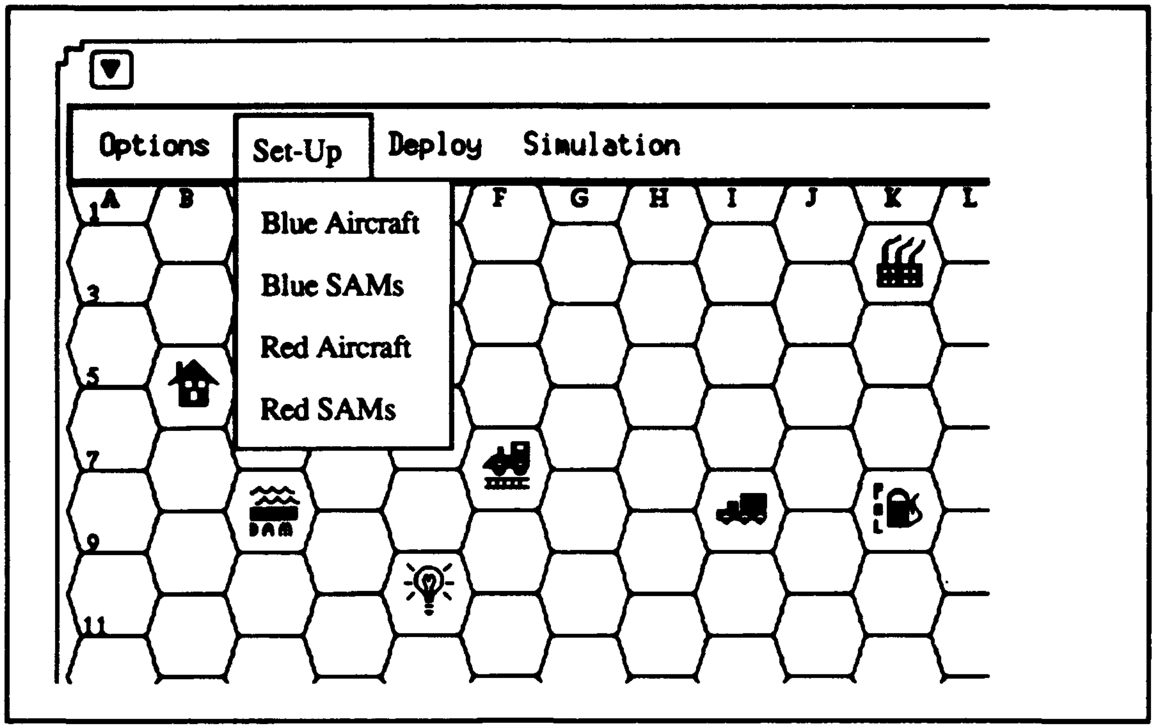

Figure A.5. Set-up menu.

A.3.2.1 Aircraft Resources

The *Set-Up Blue Aircraft* and *Set-Up Red* Aircraft functions let you specify the features of aircraft squadrons to be used in the testbed. Currently, each force is limited to a maximum of 30 aircraft squadrons a piece. Each squadron may have from one to twenty-four aircraft. Table A.1 lists the different types of aircraft available for use in the testbed. Each type of plane has certain characteristics:

- Combat Factor — This indicates the relative effectiveness of an aircraft during combat. For example, a plane with a combat factor of 10 is twice as effective as a plane with a combat factor of 5. Depending on the capabilities of the aircraft, a plane may have combat factors for air-to-air (A-to-A) and/or air-to-ground (A-to-G) combat.

- Endurance — This is related to fuel consumption and indicates the maximum number of simulation cycles an aircraft can remain airborne before running out of fuel.

- Speed — The maximum number of hex locations an aircraft can move during any one cycle of simulation.

- Maximum Engagements — This indicates the maximum number of times that an aircraft can engage the enemy in combat before running out of bullets or bombs. During air war simulations, aircraft must land to re-arm with new weapons.

Table A.1 Aircraft Types and Features

A/C Type	Combat Factor A-to-A	Combat Factor A-to-G	Endurance	Speed	Maximum Engagements A-to-A	Maximum Engagements A-to-G
F-01	5	0	3	8	3	0
F-02	10	0	4	8	3	0
F-03	15	0	5	10	3	0
FB-10	3	5	3	8	2	1
FB-20	6	8	4	8	2	1
FB-30	10	12	5	10	2	1
B-40	0	15	3	10	0	2
B-50	0	30	4	8	0	2
B-60	0	40	5	8	0	2
W-11	0	10	5	8	0	2
K-12	0	0	10	6	0	0

As shown in Table A.1, aircraft designated with an F are fighters and only have A-to-A capability. Aircraft designated with an FB are fighter-bombers and have both A-to-A and

A-to-G capability. Bombers, designated with a B, only have A-to-G capability. The remaining two types of planes have special features. Wild Weasels, indicated by a W, are SAM suppression aircraft. Tankers, designated by a K, are able to refuel other aircraft while in-flight.

A.3.2.2 Aircraft Set-Up

As already stated, the *Set-Up Aircraft* functions allow you to create aircraft resource data files After selecting the appropriate command from the pull-down menu, you will be warned that a new aircraft squadron data file will be created. (Existing files will be destroyed and replaced with the new data that you enter.) If this is what you want to do, then click on the OK button at the bottom of the warning window. If you do not want to create the new data file, then click on the CANCEL button. Figure A.6 shows the warning window that will be displayed.

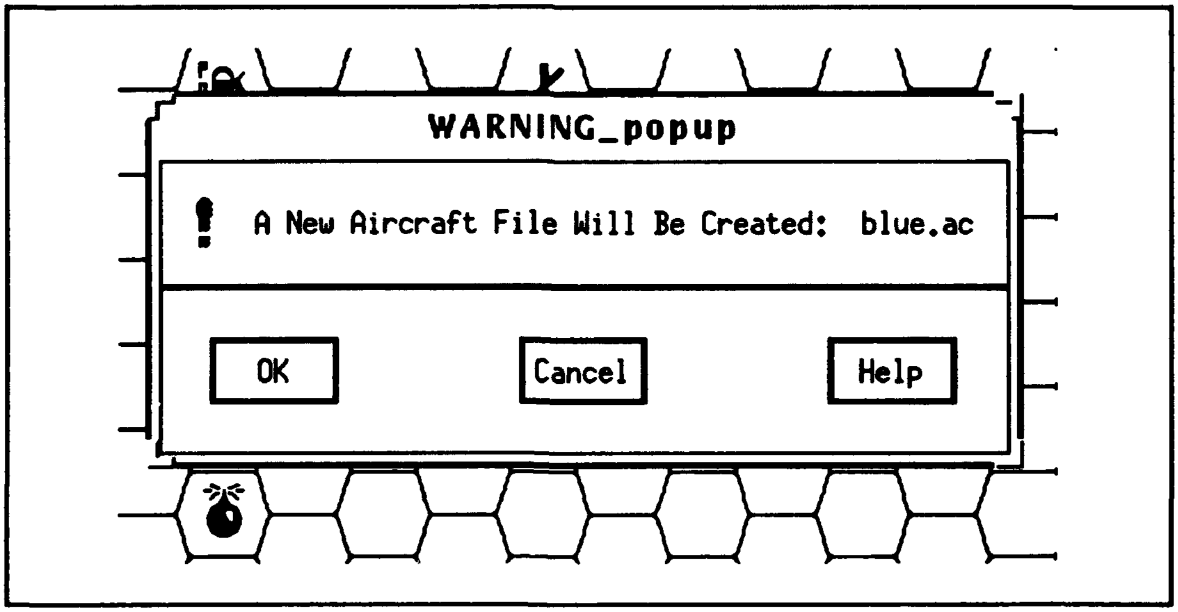

Figure A.6. Warning window.

If you select the OK button to build a new aircraft data file, then you will be taken through a series of pop-up windows which allow you to specify the characteristics for each aircraft squadron. The first window prompt you will see will be for aircraft type as shown in Figure A.7. The narrow box located to the left of the list of aircraft types is a scroll widget. By placing the mouse pointer on one of the arrow heads and pressing the left mouse button, the list of aircraft types can be scrolled up or down. Point to the aircraft type you desire and click the left mouse button. Your choice will then appear in the selection box. Finally, to confirm this selection, click on the OK button in the pop-up window. Clicking on the CANCEL button will abort data entry for the current squadron.

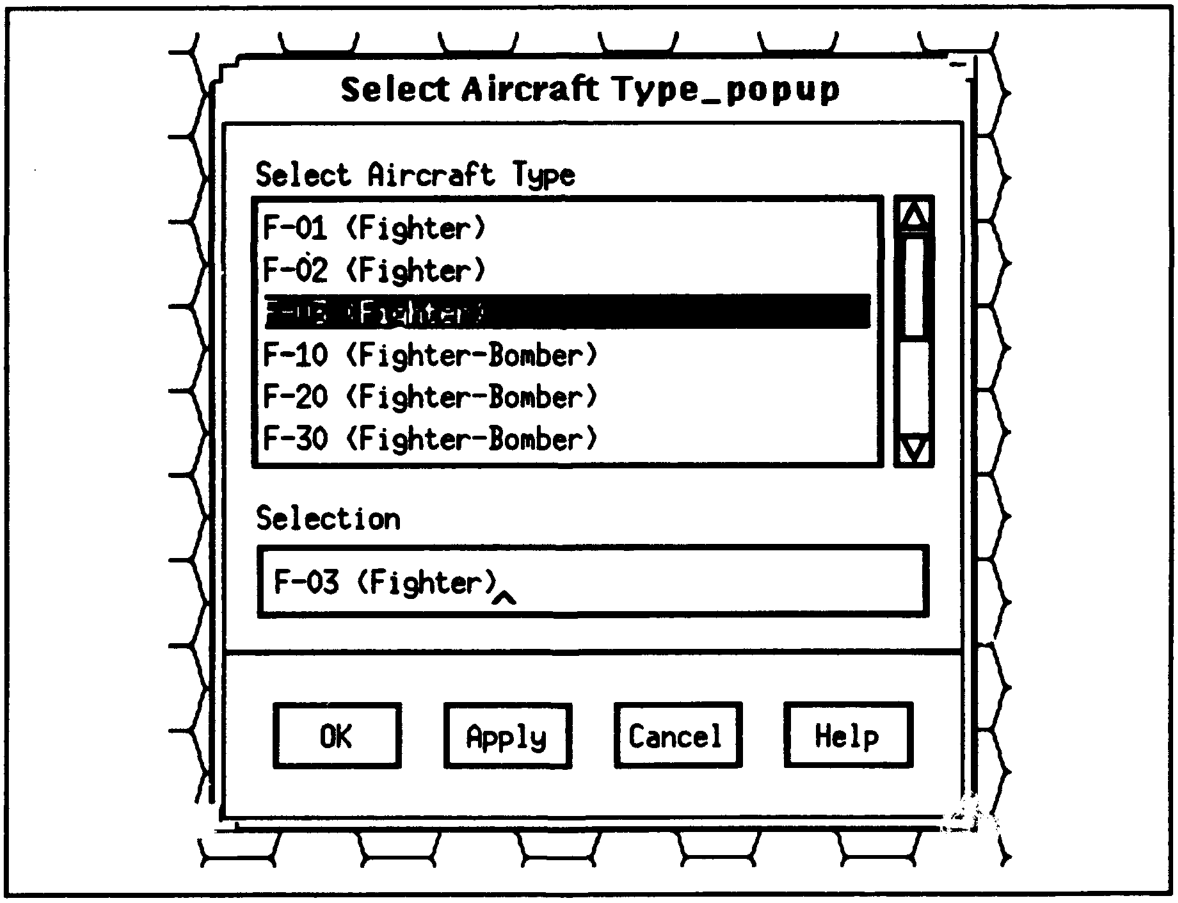

Figure A.7. Aircraft type selection window.

After choosing the type of aircraft, you will be prompted for the number of aircraft in the squadron. The squadron size selection window as shown in Figure A.8 will be displayed on the screen. To set the number of aircraft, place the mouse pointer over the scale indicator. Now, while holding the left mouse button down, you can move the scale indicator left or right by moving the mouse left or right. As the scale indicator is moved, its current numeric value is displayed. Once you have selected the desired number, click on the OK button to accept that value. Use the CANCEL button to abort data entry for this squadron.

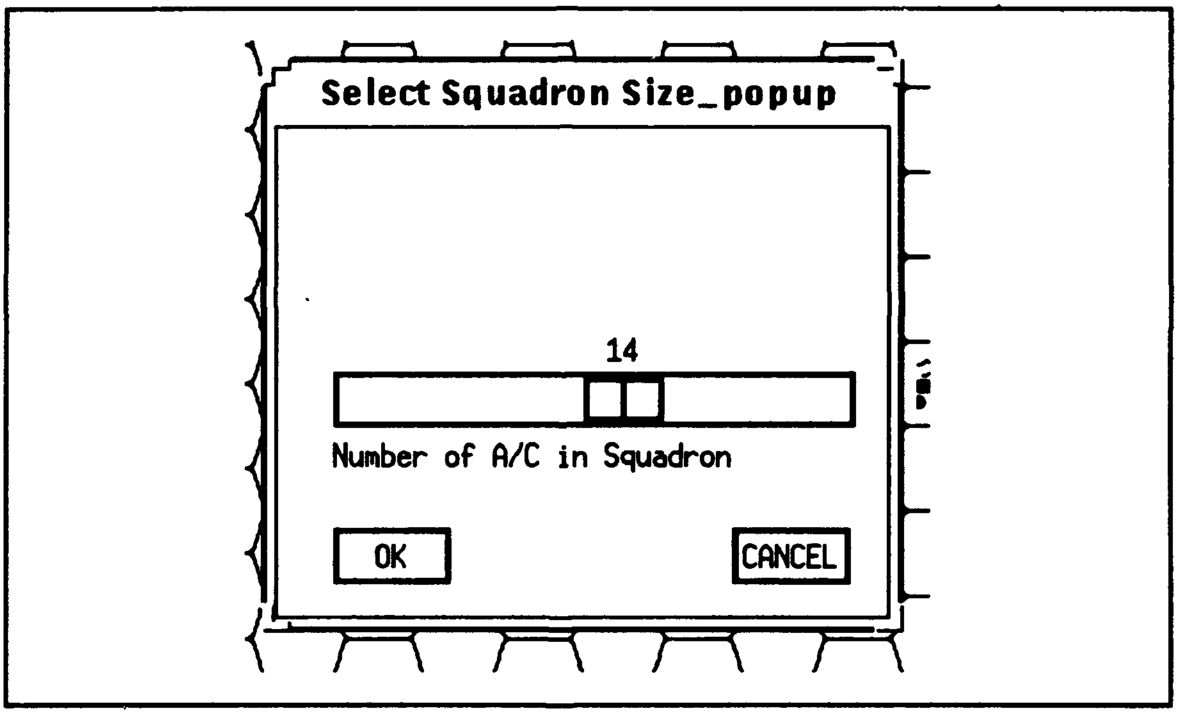

Figure A.8. Squadron size selection window.

After selecting the squadron size, the crew selection window will be displayed. This window, shown in Figure A.9, prompts you for the type of crew. A novice crew will decrease the effectiveness of a squadron by 10%, while an experienced crew increases the effectiveness by 10%. A standard crew has no change on the effectiveness of the squadron. As before, after making your selection, click on OK to accept the currently selected item or CANCEL to abort input. Next, you will be prompted for the weapon type to use with this squadron via the weapon selection window (Figure A.10). Dumb weapons decrease aircraft effectiveness by 15%, while smart weapons increase effectiveness by 15%. Standard weapons do not change combat effectiveness. Make your selection in the same manner as for crew type.

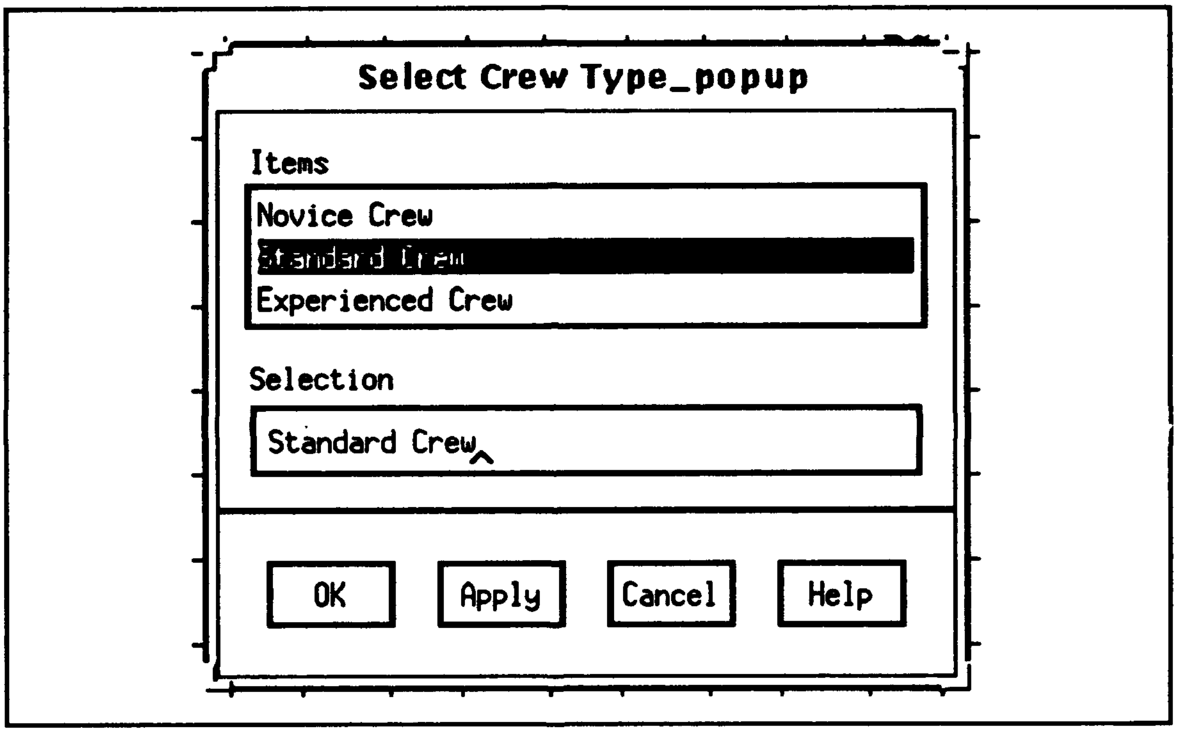

Figure A.9. Crew type selection window.

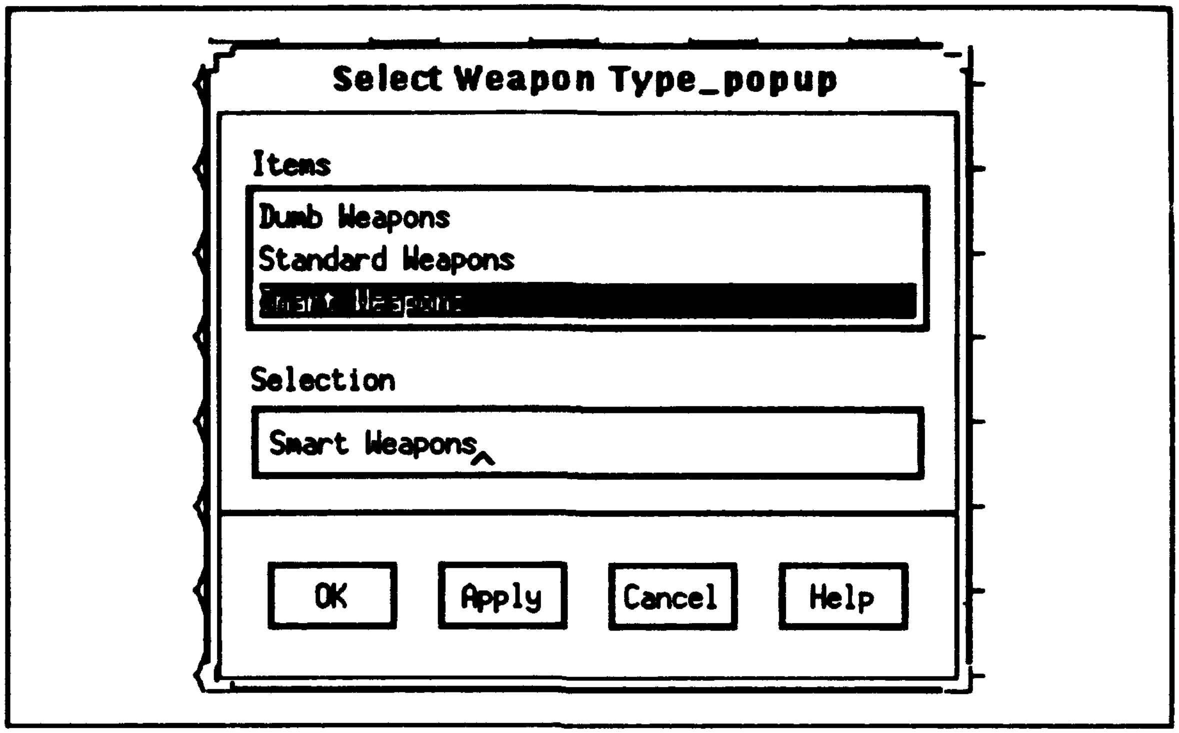

Figure A.10. Weapon type selection window.

After completing inputs for the squadron, you will be asked if you want to create another squadron. You will see a prompt like that shown in Figure A.11. Simply click on YES to build another aircraft squadron, or NO to end input. If you select YES, the process of specifying an aircraft squadron will be repeated in the same manner as before. In this way, you can set-up all the required aircraft squadrons for each force to be used in a simulation. Furthermore, during input you will see a text window at the top of the screen. This window will contain a list of all aircraft squadrons that you have established. Each time you create a new squadron, it is added to this list. Finally, if you select NO to the "Create Another Squadron" prompt, you will be asked if you want to save the new data. This prompt is shown in Figure A.12. Clicking on OK will save the data to the indicated file. Clicking on CANCEL will discard the new data and the original data file, if it exists, will be retained.

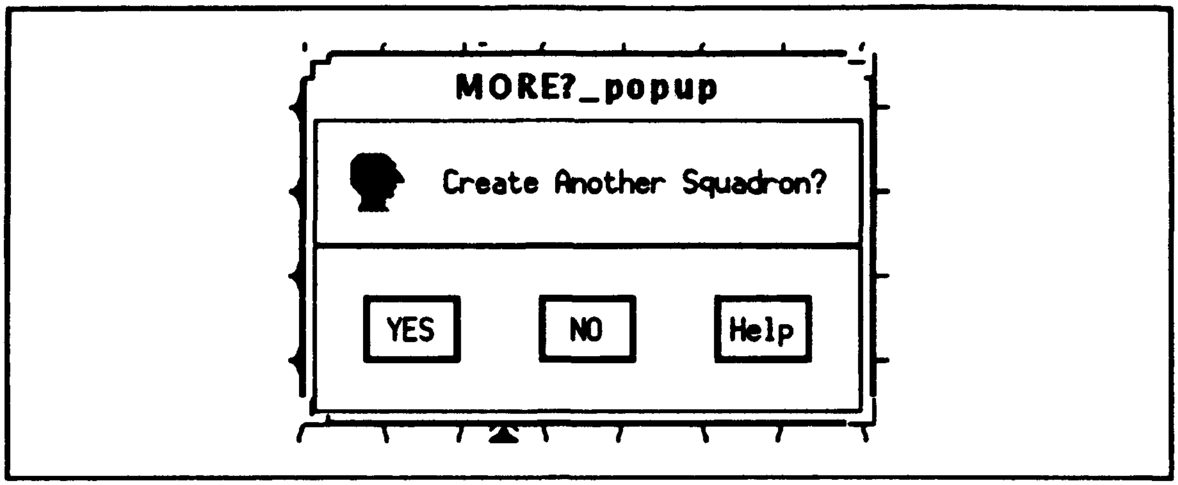

Figure A.11. Prompt for more aircraft squadrons.

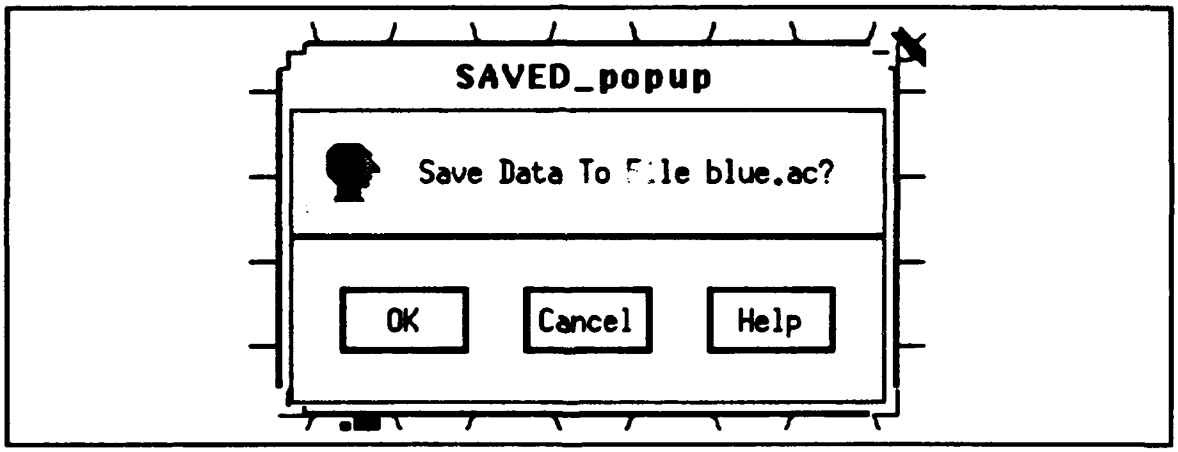

Figure A.12. Save data prompt.

A.3.2.3 SAM Resources

The testbed allows a maximum of 30 SAM units per force. Each SAM unit may contain a maximum of 24 launchers and 50 spares. During airwar simulations, the spares are used to replenish the launchers after SAMs have been fired. Two types of SAMs are available in the testbed—S-01s and S-02s. The effectiveness of a SAM in shooting down an aircraft depends on the speed of the plane. In general, the slower the plane, the greater the chance that a SAM can shoot it down. In addition, Wild Weasel aircraft are immune to SAMs. Table A.2 lists the effectiveness of SAMs based upon aircraft speed. Effectiveness is stated as the number of SAMs that must be fired to destroy one aircraft. For example, 1 in 3 means that 3 SAMs must be fired in order to shoot down one aircraft.

Table A.2. SAM Effectiveness

	Aircraft Speed (hexes/cycle)		
SAM	6	8	10
S-01	1 in 3	1 in 6	None
S-02	1 in 2	1 in 3	1 in 4

A.3.2.4 SAM Set-Up

The *Set-Up SAM* functions let you create new SAM resource data files. After you select the desired command from the *Set-Up* pull-down menu, you will be warned that a new SAM data file will be created. The SAM warning window that is displayed on the screen is very similar to the aircraft warning window shown in Figure A.6. Clicking on the OK button will allow you to continue with creating a new SAM data file. Clicking on the CANCEL button will abort this operation.

As with the aircraft set-up function, you will be taken through a series of pop-up windows prompting you for information concerning SAM units. The first prompt you will see will be the SAM type selection window. This lets you specify the type of SAMs for the unit. Figure A.13 shows this selection window. Just use the mouse to select the SAM type and then click on the OK button. You can abort data entry at this point by clicking on the CANCEL button.

If you clicked on OK, then you will be prompted to enter the number of launchers and spares for this unit. The prompts will be in the form of windows containing scalar widgets as used for aircraft squadron size selection. These two windows are shown in Figures A.14 and A.15. Simply use the mouse to move the scales to the desired values and click on OK. As before, you may click on CANCEL to abort input.

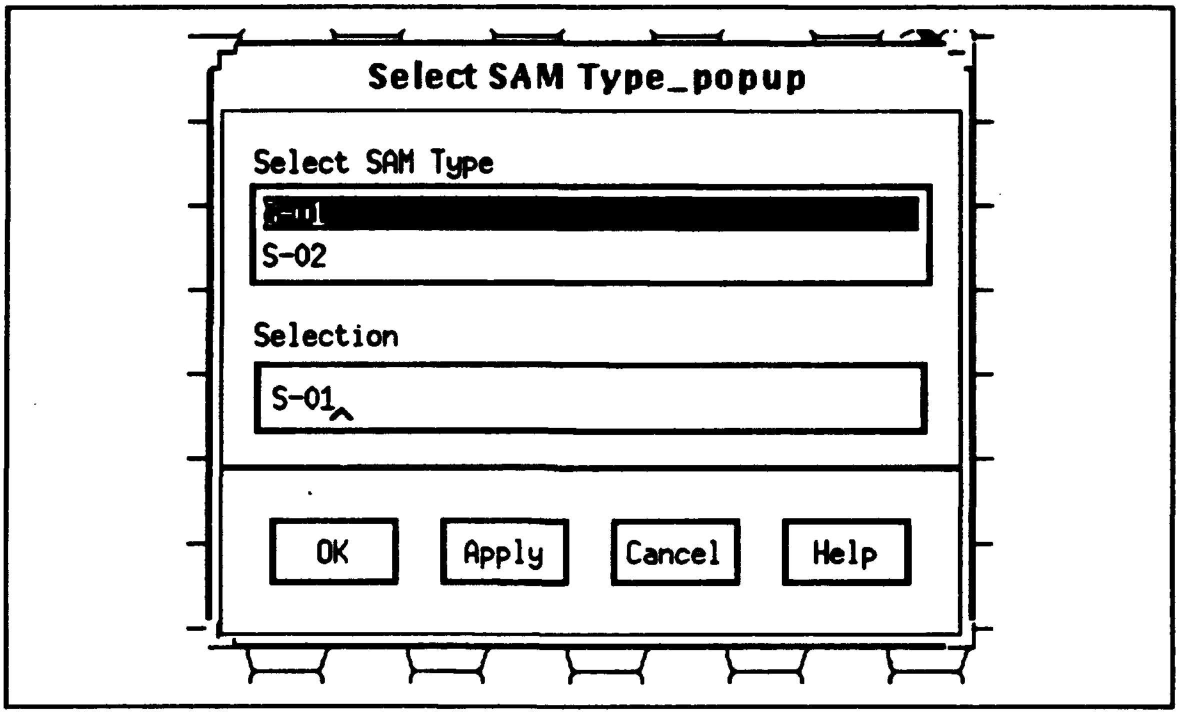

Figure A.13. SAM type selection window.

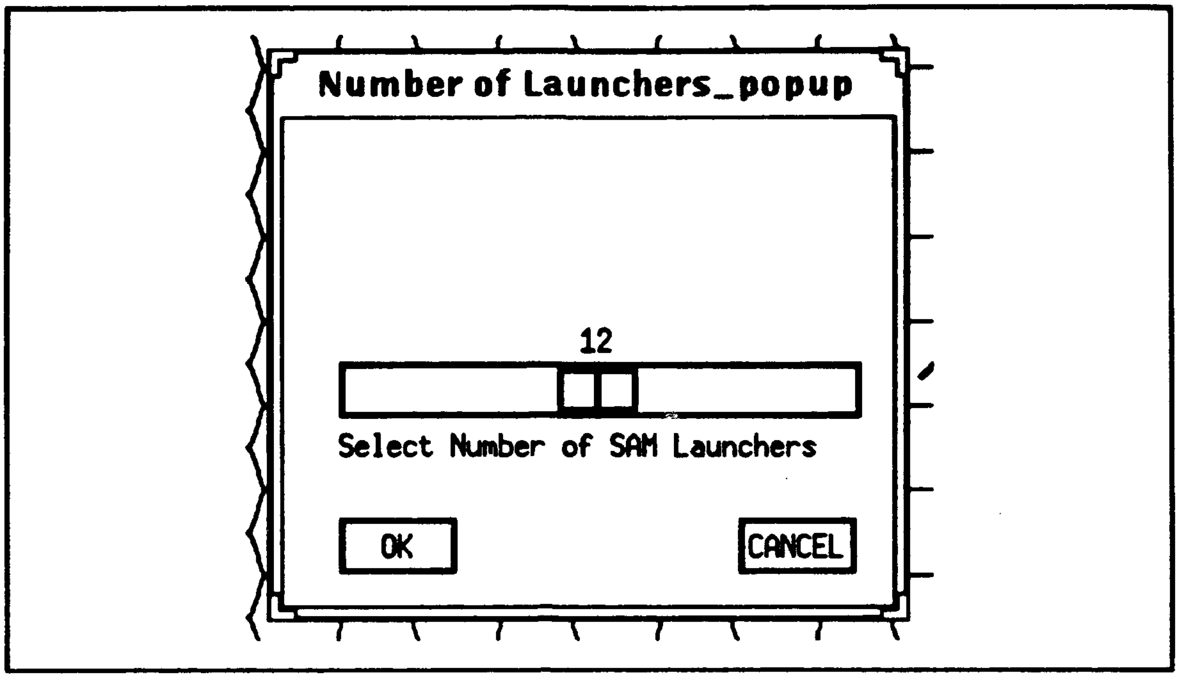

Figure A.14. SAM launcher size selection.

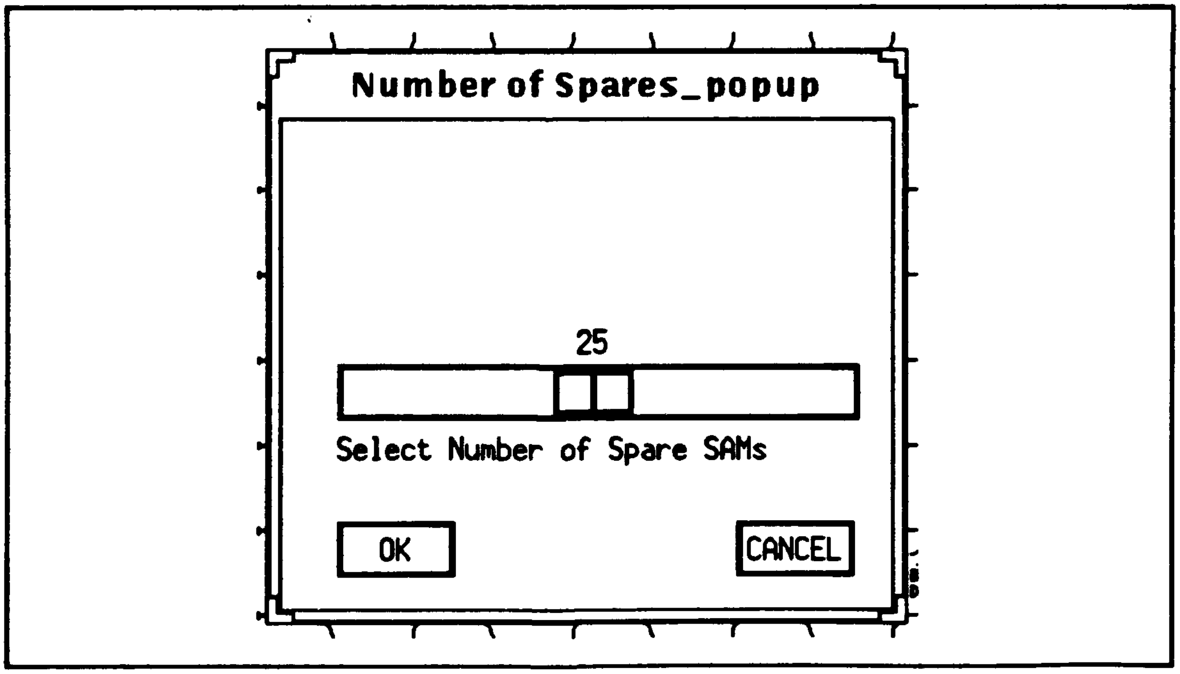

Figure A.15. Number of spare SAMs selection.

After completing inputs for the current SAM unit, you will be asked if you want to create another new unit. You will see a prompt similar to that shown in Figure A.11. If you select YES, the process of creating a SAM unit will be repeated. As with aircraft set-up, a text window will be displayed at the top of the screen, listing all units that you have established. When you have created all units that you desire, just click on the NO button when asked if you want to created another one. You will then see a window similar to that in Figure A.12 asking if you want to save the data to a file. Selecting OK will save the data the to indicated file, while selecting CANCEL will discard the new data.

A.3.3 Deploying Resources

Once all of the resource data files have been constructed, you will need to deploy them to their initial positions on the map. To do this, begin by selecting the *Deploy* function from the main menu bar. At this point, you will see the *Deploy* pull-down menu as shown in Figure A.16.

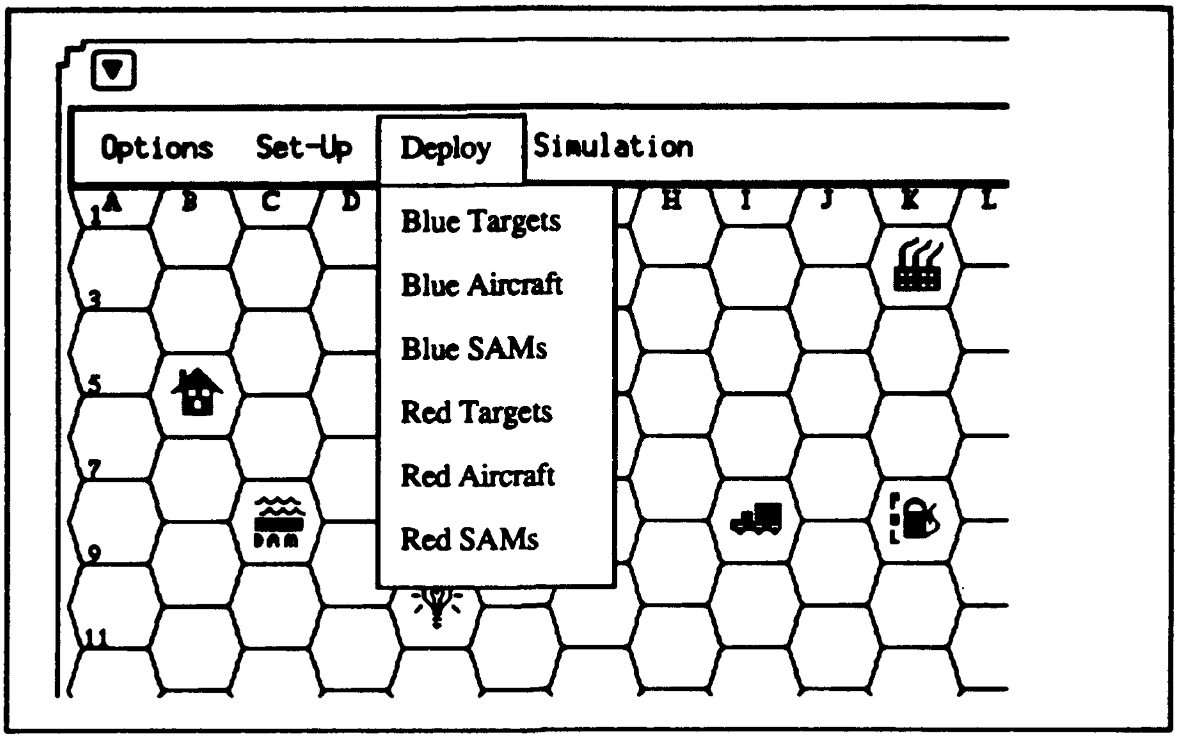

Figure A.16. Deploy pull-down menu.

A.3.3.1 Target Deployment

To deploy land-based targets, select the appropriate function from the *Deploy* pull-down menu. You will then be presented with a target deployment window like the one in Figure A.17. As can be seen, this window lists the available target types. Simply use the mouse to select the desired type of target and click on OK. The target deployment window will then disappear from the screen, allowing you to position the target on the map. To do this, just use the mouse to point to the center of the hex in which you want the target to be located and click the left-most mouse button. An icon representing that target will then appear at that location. You can continue deploying targets of the currently selected type in the same manner. When you are through with deploying this type of target, click the right-most mouse button. This will return the target deployment window. You can now select another target type, or click on the CANCEL button to end the deployment operation.

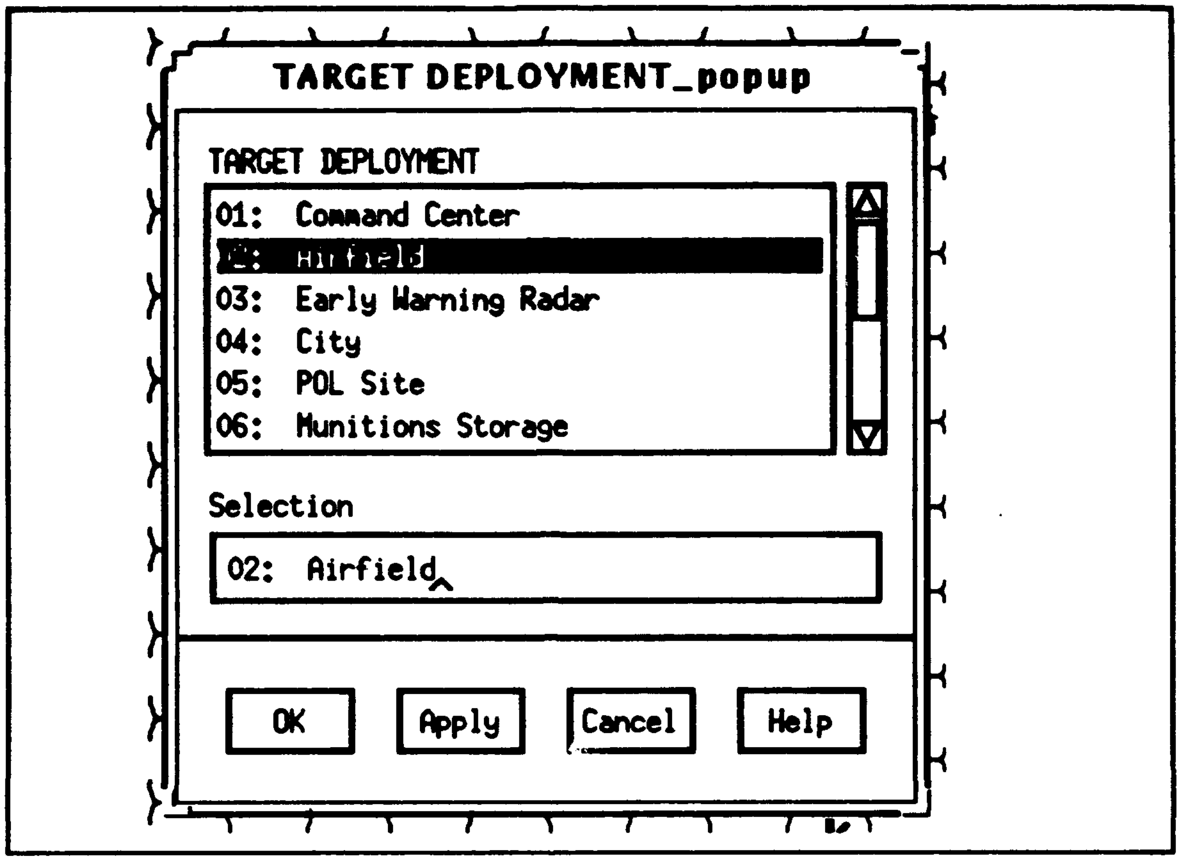

Figure A.17. Target deployment window.

The following restrictions apply concerning land-based targets. Each force is limited to a maximum of 60 targets a piece. Each force must have one, and only one, command center. This target must be deployed first. The airfields should be deployed second, and there needs to be one airfield per problem solving agent to be used with the distributed problem solving module of the testbed. There are no other restrictions concerning the remaining types of targets. Figure A.18 lists each of the available target types and shows the corresponding icons used by the testbed.

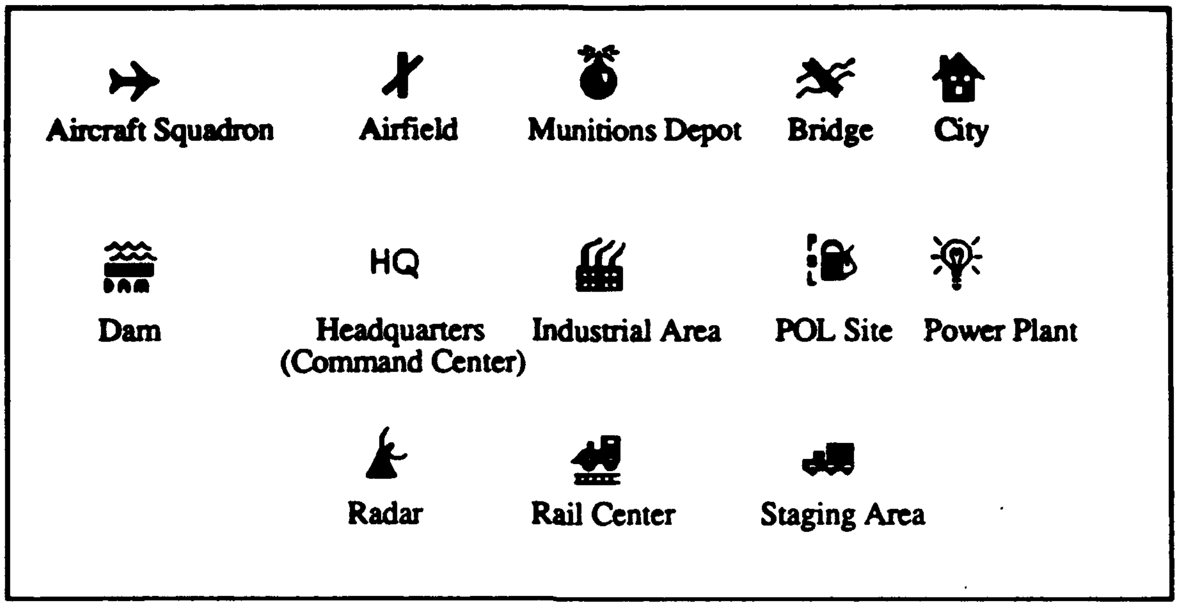

Figure A.18. Land-based targets.

A.3.3.2 Aircraft and SAM Deployment

Aircraft and SAM deployment works in a manner very similar to target deployment. Again, select the desired command from the *Deploy* pull-down menu. For aircraft, you will be presented a window containing a list of aircraft units from the aircraft data file. (These are the aircraft squadrons you established as a part of the *Set-Up* operation.) Use the mouse to select the air unit to deploy and click on OK. The aircraft deploy window will disappear from the screen. Now, use the mouse to point to the airfield (map icon) at which the aircraft squadron should be deployed. Clicking on the left-most mouse button will deploy the squadron. If you attempt to deploy the unit at an illegal location, the terminal will beep. Clicking on the right-most mouse button will cancel the deployment operation for this unit. In either case, the aircraft deployment window will reappear and you can select another unit to deploy. Proceed in this manner until you have deployed all aircraft units. When finished, click on the CANCEL button in the aircraft deployment window.

SAM deployment works the same way. Just select the SAM unit to deploy and click the left-most mouse button on the target the SAM unit is to defend. SAMs may be placed at any location which has a land-based target. No more than one SAM may defend any land-based target, i.e., only one SAM per location

A.3.4 Simulation

The *Simulation* pull-down menu is shown in Figure A.19. It provides two functions—*Next Cycle* and *Status*. The *Next Cycle* function lets you run the next cycle of simulation for an air war. The *Status* function allows you to see the status of any resource for the current air war simulation.

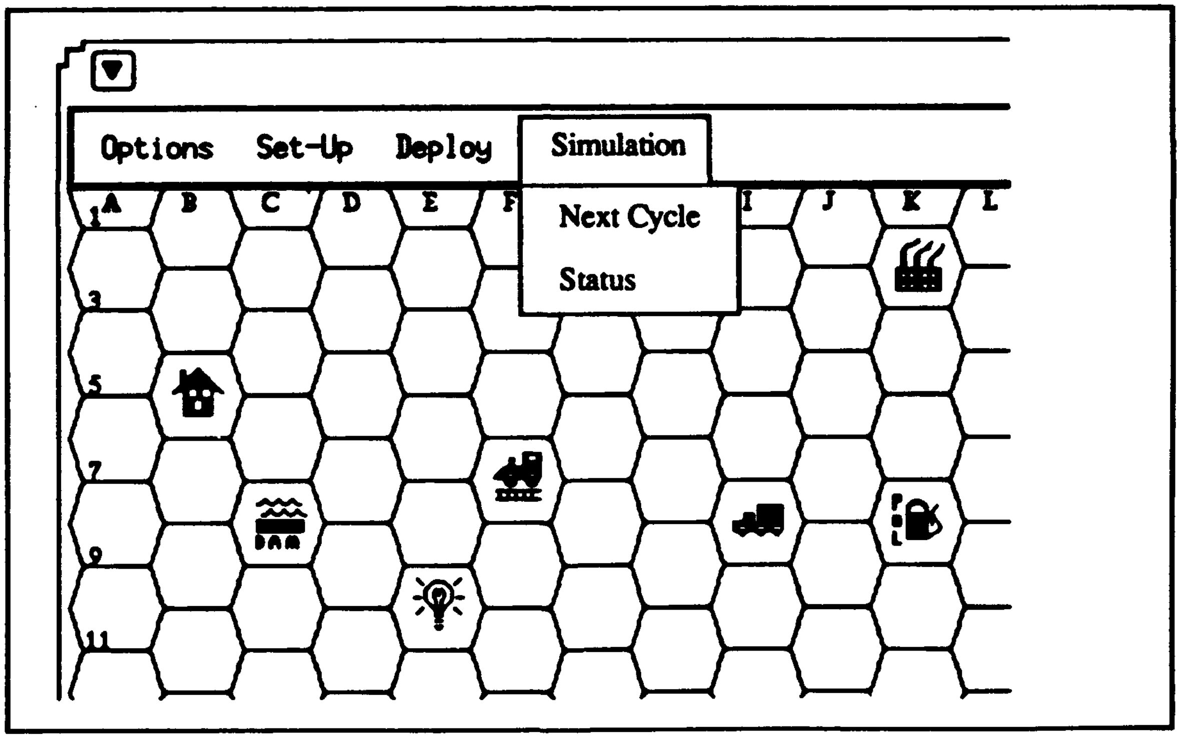

Figure A.19. Simulation pull-down menu.

A.3.4.1 Next Cycle

Selecting the *Next Cycle* option results in the testbed prompting you for the next sequence of events for the air war. Depending on the testbed mode of operation, this information will be provided interactively by the user (for Mode 0 and Mode 1) or by the distributed problem solving module (for Mode 2). Cycles of simulation occur in two distinct phases: offensive operations and defensive operations. During the offensive phase, aircraft may land, take-off, move to new locations, combat the enemy, and perform other missions such as in-flight refueling. During the defensive phase, only defensive aircraft are affected. These aircraft can take-off, move to new locations, and engage in combat. This type of information is expected by the testbed's simulation module.

Information provided interactively by the user is obtained through a sequence of pop-up windows. This first of these is the landings window shown in Figure A.20. If aircraft are currently airborne, then this window will be displayed. The window will list all airborne aircraft. Notice that for each aircraft in the list, the following information is provided: squadron ID, aircraft type, number of aircraft in the squadron, number of A-to-A and A-to-G engagements left, fuel status, and present map location. To land an aircraft, use the mouse to select a squadron from the list and click on OK. The landings window will disappear. Now point to the airfield at which to land the squadron and click the left-most mouse button. (If an invalid location is selected, the terminal will beep.) After selecting the airfield, the landings window will return, allowing you to land another squadron. Click on the CANCEL button when you are finished with landing operations.

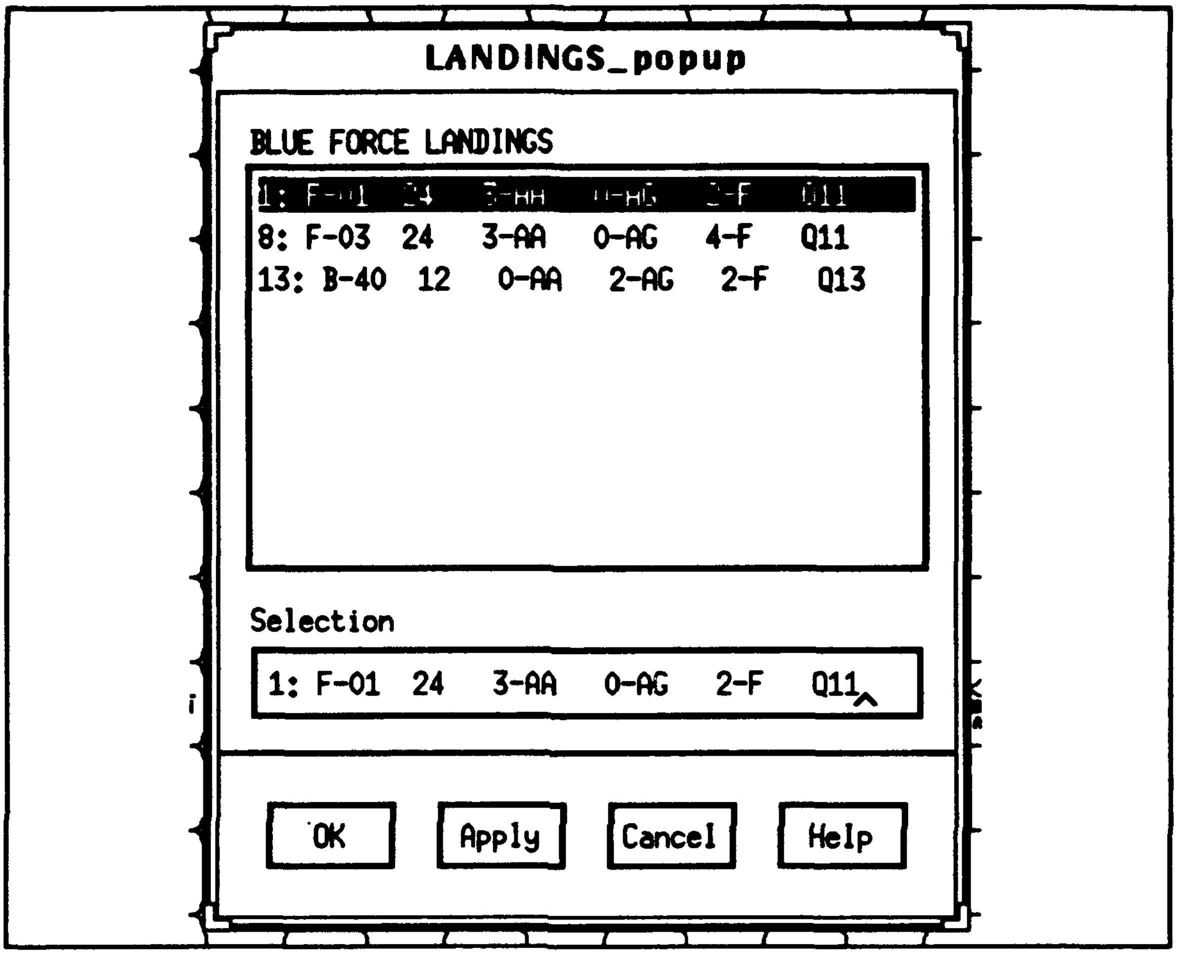

Figure A.20. Landings window.

The next window to appear on the screen will be the offensive moves window. An example of this window is shown in Figure A.21. Aircraft which are available to move are listed here. Select squadrons to move in the same way as landings. After selecting a squadron, you can point to a location on the map to which the aircraft should move during this cycle. When you have selected all aircraft that you want moved this cycle, click on the CANCEL button in the offensive moves window.

In-flight refueling information is requested next by the testbed. Note that for in-flight refueling to occur, a tanker squadron and squadron to refuel must occupy the same map location. You will be asked for refueling data via the three windows shown in Figures A.22, A.23, and A.24. Clicking on the CANCEL button in any of these windows will abort data input for refuelings.

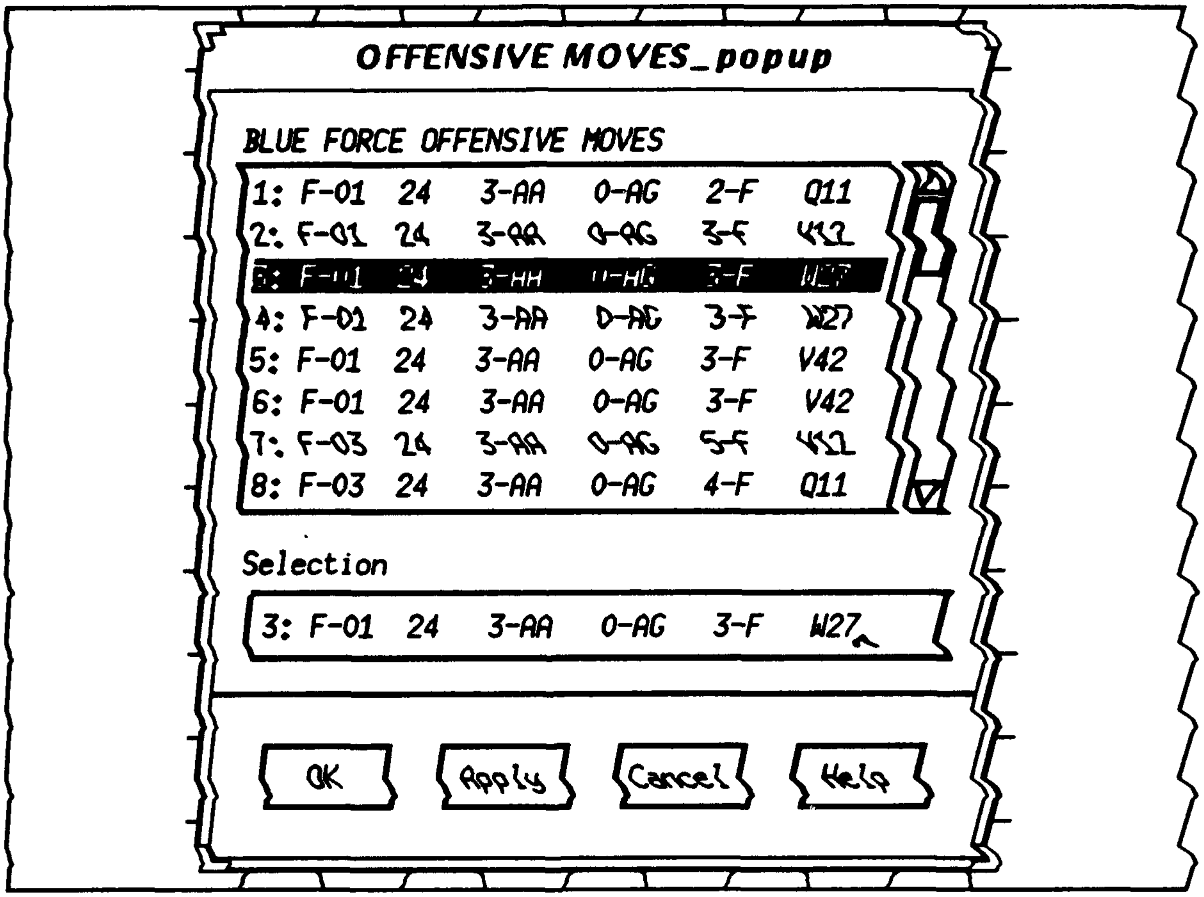

Figure A.21. Offensive moves window.

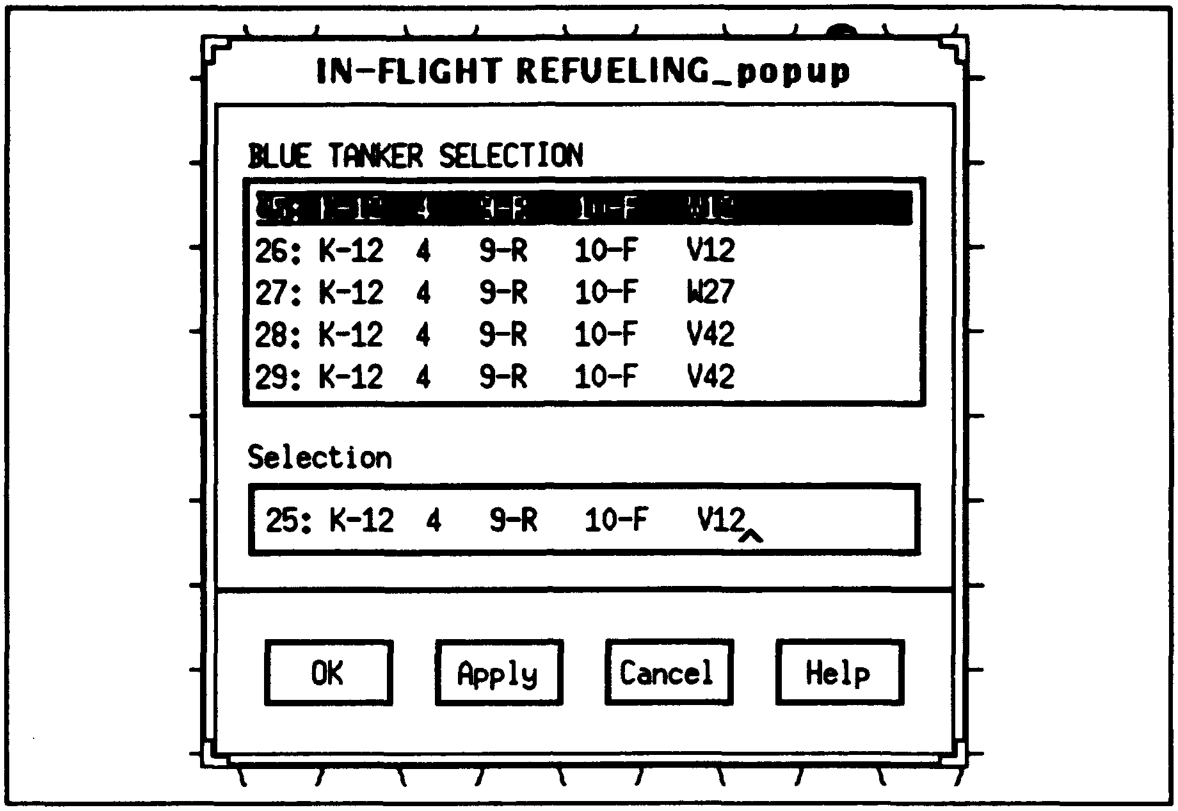

Figure A.22. Tanker selection window.

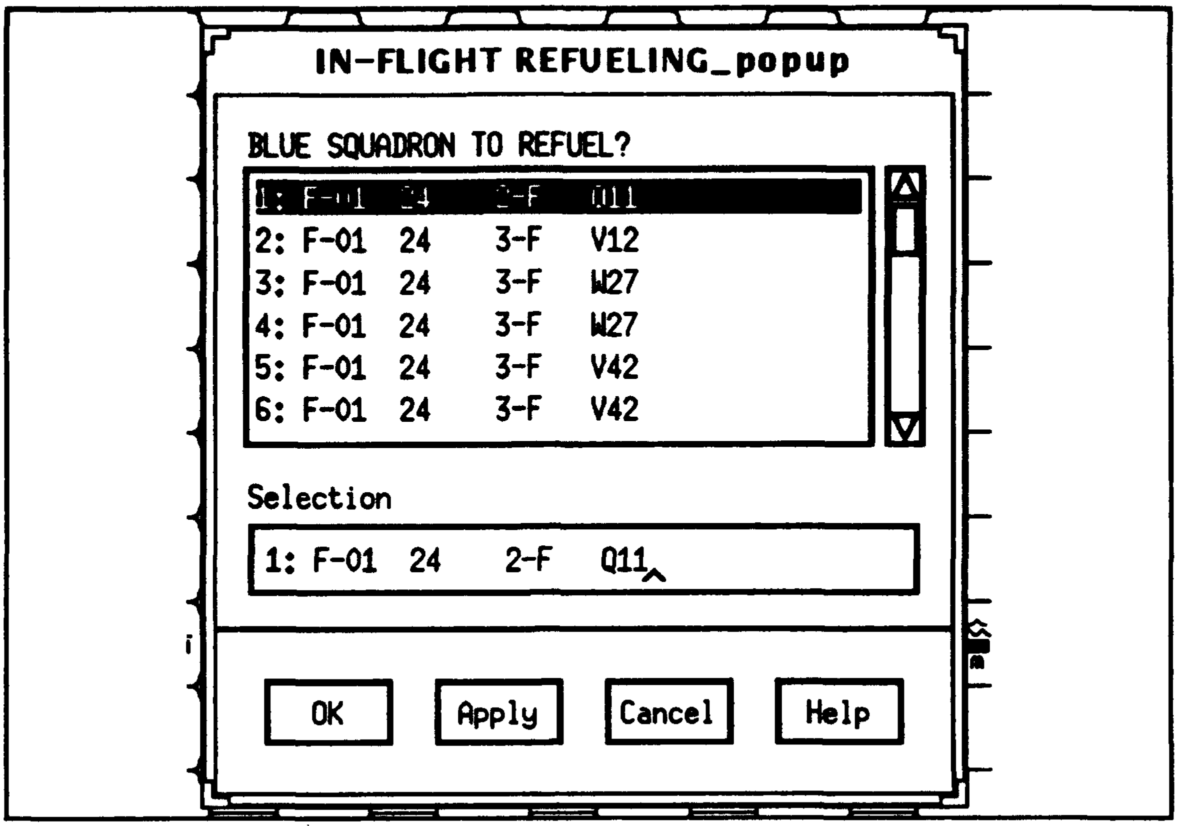

Figure A.23. Refuel squadron selection window.

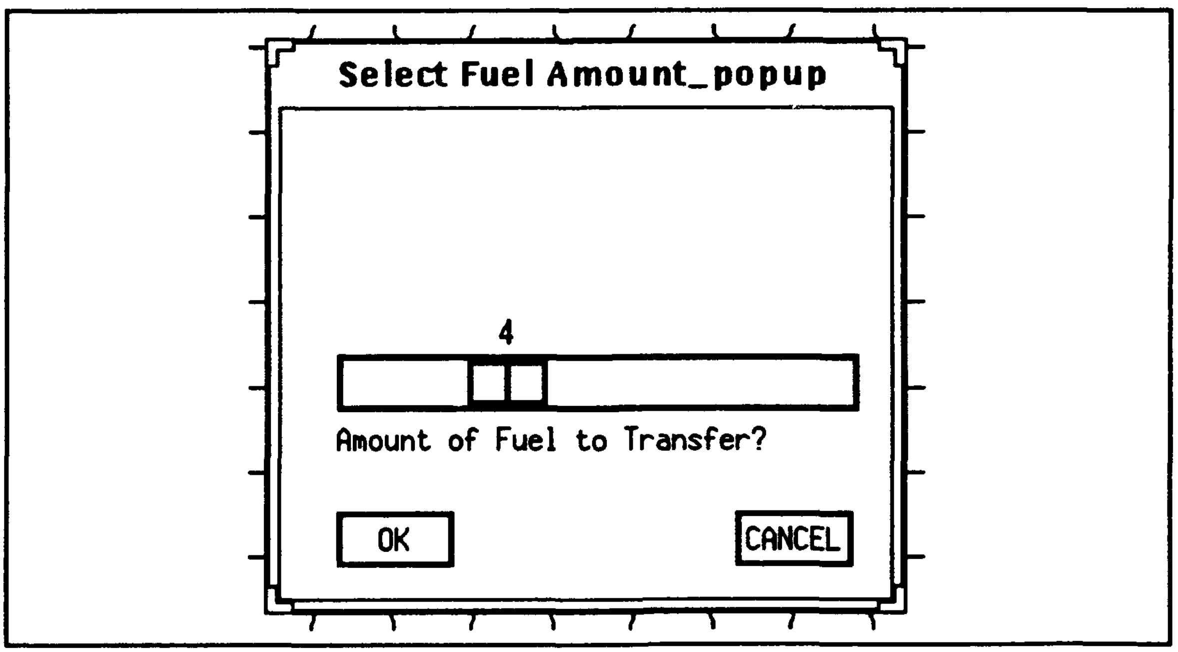

Figure A.24. Fuel amount selection window.

The refueling data selection marks the end of the offensive phase. Once data for both the blue and red forces have been entered, aircraft locations will be updated on the screen. Keep in mind that only airborne aircraft are displayed on the map. In addition, units occupying the same location are indicated by two icons being overlaid on one another, with one icon being offset from the other. At this point, you may query the testbed to receive status information about resources. (This process will be explained in the next section.)

With the completion of the status operation, the defensive phase begins. This will consist of one window as shown in Figure A.25. This is the defensive moves window. It lists all defensive aircraft that have not moved during this cycle. Select and move aircraft in the same manner as for the offensive phase. At the conclusion of data selection for defensive operations, the cycle will be simulated (movement, crashes, combat, landings, refuelings, etc.) A standard report file will be created listing all events which occurred this cycle. The file will be named *results.n* where *n* is the cycle number.

A.3.4.2 Status

The *Status* function may be used at any point between simulation cycles and between offensive and defensive phases within a simulation cycle. To use this feature, select *Status* from the *Simulation* pull-down menu. (This function is automatically invoked by the testbed after an offensive phase.) An information window will pop-up, informing you that this function is now active. Click on the OK button in this information window to make the window clear from the screen. With the *Status* function active, you may point to any location on the map and click the left-most mouse button. A status window will then appear, showing you the current status of all resources at that location. Figure A.26 is a sample status window. Clicking on the OK button in this window lets you pick another location for status information. Clicking on the CANCEL button deactivates the status feature.

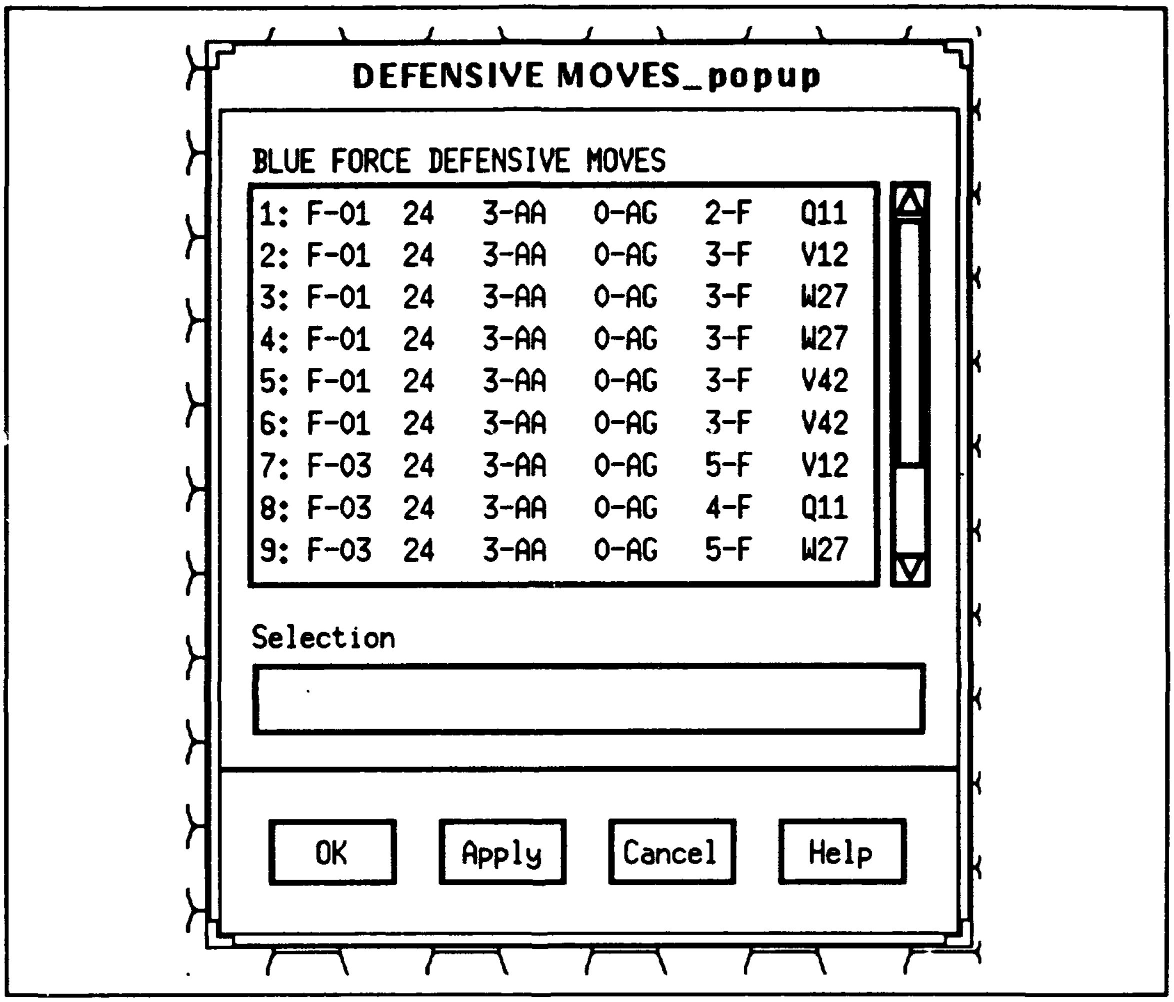

Figure A.25. Defensive moves window.

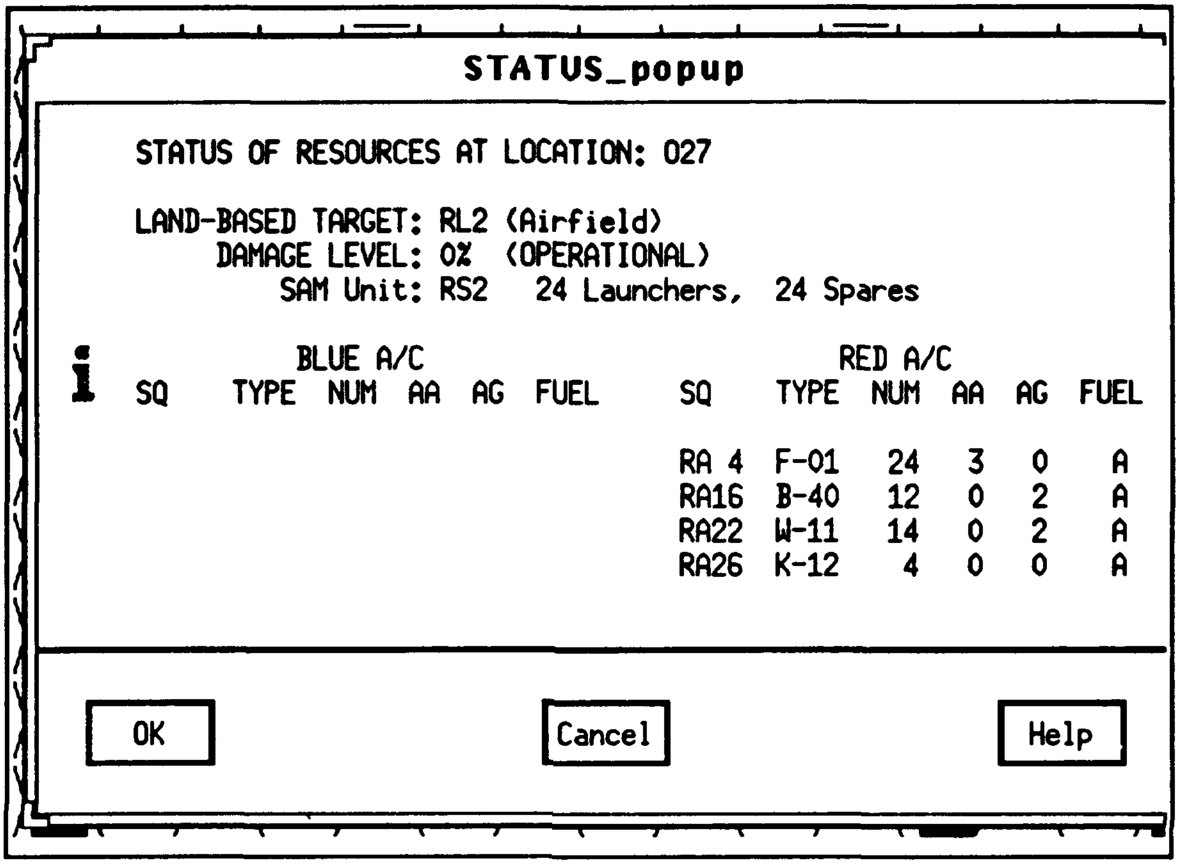

Figure A.26. Status window.

A.4 Support Programs

Two support programs are available for use with the testbed. These programs are described below:

- *mirror* — This program allows you to create a mirror image of a force. This is useful if you want to develop an air war scenario in which both forces are equal in terms of types, numbers, and relative locations of resources. To use this program, create the aircraft and SAM data files for the red force. Then, deploy the targets, aircraft, and SAMs for the red force. Finally, run the program *mirror*. It will then create all of the necessary data files for the blue force. This file will mirror the red force data files.

- *report* — This program produces two report files. The first is a standard report put in the file *standard.rpt*. This report contains standard information about aircraft and land-based targets features. The second report is put into a file called *status.rpt*. This is a summary of the current status of all resources for both forces.

APPENDIX B

SAMPLE SIMULATION RESULTS FILE

SIMULATION CYCLE #3

LANDINGS -- BLUE FORCE

LANDINGS -- RED FORCE

RA5 AT RL1

BLUE AIRCRAFT OFFENSIVE MOVEMENT

BA4 MOVED TO E17 (2)
BA5 MOVED TO E17 (2)
BA7 MOVED TO E17 (3)
BA12 MOVED TO E17 (1)
BA14 MOVED TO E17 (3)
BA18 MOVED TO E17 (3)
BA20 MOVED TO E17 (3)
BA22 MOVED TO E17 (1)
BA23 MOVED TO E17 (1)
BA24 MOVED TO E17 (3)
BA25 MOVED TO E17 (3)
BA30 MOVED TO M19 (9)
BA1 MOVED TO M19 (1)
BA2 MOVED TO M19 (1)
BA3 MOVED TO M19 (1)
BA28 TAKE-OFF TO LOCATION O47 (10)
BA6 MOVED TO O47 (2)
BA8 MOVED TO O47 (3)
BA11 MOVED TO O47 (1)
BA15 MOVED TO O47 (2)
BA16 MOVED TO O47 (2)
BA26 MOVED TO O47 (3)
BA27 MOVED TO O47 (3)

RED AIRCRAFT OFFENSIVE MOVEMENT

RA6 MOVED TO U49 (2)
RA10 MOVED TO U49 (1)
RA11 MOVED TO U49 (1)
RA15 MOVED TO U49 (2)
RA16 MOVED TO U49 (2)
RA26 MOVED TO U49 (3)
RA27 MOVED TO U49 (3)
RA4 MOVED TO S11 (2)
RA14 MOVED TO S11 (3)
RA22 MOVED TO S11 (1)
RA23 MOVED TO S11 (1)
RA24 MOVED TO S11 (3)

```
BLUE FORCE INFLIGHT REFUELINGS

     BA11 REFUELED BY BA28 (2)

RED FORCE INFLIGHT REFUELINGS

BLUE AIRCRAFT DEFENSIVE MOVEMENT

     BA10 MOVED TO U49 (1)

RED AIRCRAFT DEFENSIVE MOVEMENT

     RA2  MOVED TO E17 (1)
     RA3  MOVED TO E17 (1)
     RA7  MOVED TO E17 (3)
     RA1  MOVED TO M19 (1)
     RA9  MOVED TO M19 (3)
     RA13 MOVED TO M19 (3)
     RA8  MOVED TO O47 (3)
     RA12 MOVED TO E17 (1)

BLUE FORCE CRASHES


RED FORCE CRASHES


ENGAGEMENT 1


BLUE AIRCRAFT (AIRBORNE): BA1 BA2 BA3 BA30

 RED AIRCRAFT (AIRBORNE): RA1 RA9 RA13

       LAND TARGETS:

BLUE AIRCRAFT: FIGHTERS= 51  FIGHTER-BOMBERS= 0
BLUE A/C SQUADRONS: FIGHTERS= 3  BOMBERS= 0  OTHERS= 1

RED AIRCRAFT:  FIGHTERS= 35  FIGHTER-BOMBERS= 20
RED A/C SQUADRONS:  FIGHTERS= 3  BOMBERS= 0  OTHERS= 0

     FIGHTER-TO-FIGHTER COMBAT
```

RA13 HAS NO BULLETS

COMBAT FACTORS

BLUE= 318
RED= 518

FOG/FRICTION ADJUSTMENT

BLUE= 254
RED= 424

RA1 LOST 4 F-01
BA1 LOST 3 F-01
BA1 NO LONGER EXISTS

BA2 LOST 4 F-01
FIGHTER-TO-OTHER COMBAT

RA13 HAS NO BULLETS

COMBAT FACTORS

BLUE= 1
RED= 1

ENGAGEMENT 2

BLUE AIRCRAFT (AIRBORNE): BA4 BA5 BA7 BA12 BA14 BA18 BA20
BA22
BA23 BA24 BA25

RED AIRCRAFT (AIRBORNE): RA2 RA3 RA7 RA12

LAND TARGETS: RL0

BLUE AIRCRAFT: FIGHTERS= 58 FIGHTER-BOMBERS= 40
BLUE A/C SQUADRONS: FIGHTERS= 5 BOMBERS= 4 OTHERS= 2

RED AIRCRAFT: FIGHTERS= 62 FIGHTER-BOMBERS= 20
RED A/C SQUADRONS: FIGHTERS= 4 BOMBERS= 0 OTHERS= 0

FIGHTER-TO-FIGHTER COMBAT

RA12 JETTISONED BOMBS!

BA12 HAS NO BULLETS
BA14 HAS NO BULLETS
RA12 HAS NO BULLETS

```
     COMBAT FACTORS

BLUE= 875
 RED= 637

     FOG/FRICTION ADJUSTMENT

BLUE= 813
 RED= 433

RA2 LOST 7 F-01
BA4 LOST 8 F-02
     WEASELS-TO-SAMS

     BLUE FORCE ATTACKING

WEASEL COMBAT FACTOR = 199

     FOG/FRICTION ADJUSTMENT

RS6 LOST 8 S02

     GROUND-TO-AIR

12 SAM(s) fired.  4 hit(s), 8 miss(es).

BA18 LOST 4 B-60
BA18 NO LONGER EXISTS


     AIR-TO-GROUND

BLUE FORCE ATTACKING

BA12 HAS NO BOMBS!
BA14 HAS NO BOMBS!

     COMBAT FACTORS

A/C= 450     TARGET= 500

     FOG/FRICTION ADJUSTMENT

A/C= 270     TARGET= 500

RLO RECEIVED 54% DAMAGE
TOTAL DAMAGE NOW IS 54%


ENGAGEMENT 3
```

BLUE AIRCRAFT (AIRBORNE): BA6 BA8 BA11 BA15 BA16 BA26 BA27 BA28

RED AIRCRAFT (AIRBORNE): RA8

LAND TARGETS:

BLUE AIRCRAFT: FIGHTERS= 43 FIGHTER-BOMBERS= 20
BLUE A/C SQUADRONS: FIGHTERS= 3 BOMBERS= 2 OTHERS= 3

RED AIRCRAFT: FIGHTERS= 13 FIGHTER-BOMBERS= 0
RED A/C SQUADRONS: FIGHTERS= 1 BOMBERS= 0 OTHERS= 0

FIGHTER-TO-FIGHTER COMBAT

BA11 HAS NO BULLETS

COMBAT FACTORS

BLUE= 674
RED= 243

FOG/FRICTION ADJUSTMENT

BLUE= 633
RED= 235

RA8 LOST 6 F-03
BA6 LOST 1 F-02

ENGAGEMENT 4

BLUE AIRCRAFT (AIRBORNE): BA10

RED AIRCRAFT (AIRBORNE): RA6 RA10 RA11 RA15 RA16 RA26 RA27

LAND TARGETS:

BLUE AIRCRAFT: FIGHTERS= 0 FIGHTER-BOMBERS= 20
BLUE A/C SQUADRONS: FIGHTERS= 1 BOMBERS= 0 OTHERS= 0

RED AIRCRAFT: FIGHTERS= 21 FIGHTER-BOMBERS= 40
RED A/C SQUADRONS: FIGHTERS= 3 BOMBERS= 2 OTHERS= 2

FIGHTER-TO-FIGHTER COMBAT

BA10 JETTISONED BOMBS!

COMBAT FACTORS

BLUE= 100
RED= 462

FOG/FRICTION ADJUSTMENT

BLUE= 86
RED= 328

BA10 LOST 14 F-10

APPENDIX C

SAMPLE AGENT OUTPUT DURING PROBLEM SOLVING

```
Host: volga, My Name: AGENT2, ID: 2

Rules file: aw-rules.clp
Watch Facts Off
Watch Rules Off
Watch Activations Off
Watch All Off
Resource Sharing On

Ready for next set of tasks....

Received a LOAD_TASKS message.
Got name of task file: otask1.dat
Tasks data loaded.

Dist to task #1 = 11
Dist to task #2 = 8
Dist to task #7 = 11

Sorted tasks:

Task #2
Task #1
Task #7

Iteration #1:

QM of Task #2 = 8
QM of Task #1 = 11
QM of Task #7 = 11

Sent my bids.  Waiting on bids from other 2 agents.
Got the bids from other agents.

Received bids from Agent 1: T1 B = 6 T2 B = 11 T7 B = 13
Received bids from Agent 3: T2 B = 11 T7 B = 12 T1 B = 18
Bid data as received:
Task #1:  A2 B = 11, A1 B = 6, A3 B = 18,
Task #2:  A2 B = 8, A1 B = 11, A3 B = 11,
Task #7:  A2 B = 11, A1 B = 13, A3 B = 12,

Agent #1 gets Task #1 with a bid of 6
Agent #2 gets Task #2 with a bid of 8
Task #7 not assigned a commander.  Best = 11, Next Best = 12,
Thres = 2
Bid data after sorting for best:
Task #1:  A1 B = 6, A2 B = 11, A3 B = 18,
Task #2:  A2 B = 8, A1 B = 11, A3 B = 11,
Task #7:  A2 B = 11, A3 B = 12, A1 B = 13,
```

```
Quiessence has not been achieved yet.
I am agent 2.  My workload is 1

All tasks have not been assigned.
Workload of agent #1 is 1
Workload of agent #2 is 1
Workload of agent #3 is 0


Iteration #2:

QM of Task #7 = 16

Sent my bids.  Waiting on bids from other 2 agents.
Got the bids from other agents.

Received bids from Agent 1: T7 B = 18
Received bids from Agent 3: T7 B = 12
Bid data as received:
Task #1:  A1 B = 6, A2 B = 11, A3 B = 18,
Task #2:  A2 B = 8, A1 B = 11, A3 B = 11,
Task #7:  A2 B = 16, A1 B = 18, A3 B = 12,

Agent #1 gets Task #1 with a bid of 6
Agent #2 gets Task #2 with a bid of 8
Agent #3 gets Task #7 with a bid of 12

Bid data after sorting for best:
Task #1:  A1 B = 6, A2 B = 11, A3 B = 18,
Task #2:  A2 B = 8, A3 B = 11, A1 B = 11,
Task #7:  A3 B = 12, A2 B = 16, A1 B = 18,

All tasks have been assigned.
Workload of agent #1 is 1
Workload of agent #2 is 1
Workload of agent #3 is 1

Weighting factors:  W1= 10 W2= 1 W3= 1 W4= 5 T= 2
Prioritized Tasks:

Task #1: P = 40, D = 3, C = 1
Task #7: P = 36, D = 3, C = 3
Task #2: P = 32, D = 3, C = 2

Solving Task #2
Time-to-target needed is 3.  Target hardness is 1000.
Allocating my bombers for air-to-ground role.  2 bombers
allocated.
Additional air-to-ground firepower needed.
Allocating my fighter-bcmbers for air-to-ground role.  0
fighter-bombers allocated.
```

```
Additional air-to-ground firepower needed.
Allocating bombers for air-to-ground role from other agents.
2 bombers allocated.
Allocated A-G firepower is 1200, target hardness is 1000.
Allocating my fighters for escort role.  2 fighters
allocated.
Allocating my Wild Weasels for SEAD role.  2 Wild Weasels
allocated.

Allocated squadrons are 9, 10, 15, 16, 17, 18, 21, 22,

Received a TASK_SOLVED message from Agent #1
Resource Conflict:  Agent #1 has allocated Sqd #17 which I
also want to allocate.  Agent #1 has priority.
Redoing my solution for Task #2

Time-to-target needed is 3.  Target hardness is 1000.
Allocating my bombers for air-to-ground role.  2 bombers
allocated.
Additional air-to-ground firepower needed.
Allocating my fighter-bombers for air-to-ground role.  0
fighter-bombers allocated.
Additional air-to-ground firepower needed.
Allocating bombers for air-to-ground role from other agents.
2 bombers allocated.
Allocated A-G firepower is 1100, target hardness is 1000.
Allocating my fighters for escort role.  2 fighters
allocated.
Allocating my Wild Weasels for SEAD role.  2 Wild Weasels
allocated.

Allocated squadrons are 9, 10, 14, 15, 16, 18, 21, 22,

Task #2

  Type: Strategic Aerospace Offense
  Location: 027
  Importance: 8 Urgency: 4 Priority: 32
  Deadline: 3
  Allocated Resources: 14 15 16 18 10 9 22 21
  Solution:
```

```
CYCLE #1
ACTION: Combat
   Sqd #14 action: Take-Off  from:   V12 to:   O27
   Sqd #15 action: Take-Off  from:   W27 to:   O27
   Sqd #16 action: Take-Off  from:   W27 to:   O27
   Sqd #18 action: Take-Off  from:   V12 to:   O27
   Sqd #10 action: Take-Off  from:   W27 to:   O27
   Sqd # 9 action: Take-Off  from:   W27 to:   O27
   Sqd #22 action: Take-Off  from:   W27 to:   O27
   Sqd #21 action: Take-Off  from:   W27 to:   O27

CYCLE #2
ACTION: Move
   Sqd #14 action: Landing   from:   O27 to:   V12
   Sqd #15 action: Landing   from:   O27 to:   W27
   Sqd #16 action: Landing   from:   O27 to:   W27
   Sqd #18 action: Landing   from:   O27 to:   V12
   Sqd #10 action: Landing   from:   O27 to:   W27
   Sqd # 9 action: Landing   from:   O27 to:   W27
   Sqd #22 action: Landing   from:   O27 to:   W27
   Sqd #21 action: Landing   from:   O27 to:   W27


Task #1 is solved.
Task #7 is solved.
Task #2 is solved.

Ready for next set of tasks....
```

APPENDIX D

LIST OF RESOURCES FOR THE AIR WAR SCENARIOS

This appendix lists the aircraft squadrons, SAM units, and land-based resources used for each air war scenario. For the aircraft squadrons, the following information is shown: squadron identifier (SQD), the agent which controls the resource (OWNER), the type of aircraft in the squadron (TYPE), the number of aircraft in the squadron (NUM), and the map coordinate indicating the location of the squadron at the start of the air war (LOCATION). For SAM units, the information consists of: unit identifier (ID), type of SAMs in the unit (TYPE), location of the unit (LOC), number of launchers (NUM LAUNCHERS), and number of spares (NUM SPARES). Finally, the following information is presented for land-based resources: identifier (ID), type of resource (TYPE), identifier of SAM unit protecting this resource (SAMs), map location (LOC), damage level (DAMAGE), and hardness factor associated with the resource (HARDNESS).

Scenario #1

AIRCRAFT SQUADRON RESOURCES

BLUE FORCE

SQD	OWNER	TYPE	NUM	LOCATION
1	1	F-01	24	V12
2	1	F-01	24	V12
3	2	F-01	24	W27
4	2	F-01	24	W27
5	3	F-01	24	V42
6	3	F-01	24	V42
7	1	F-03	24	V12
8	1	F-03	24	V12
9	2	F-03	24	W27
10	2	F-03	24	W27
11	3	F-03	24	V42
12	3	F-03	24	V42
13	1	B-40	12	V12
14	1	B-40	12	V12
15	2	B-40	12	W27
16	2	B-40	12	W27
17	3	B-40	12	V42

SQD	OWNER	TYPE	NUM	LOCATION
18	3	B-40	12	V42
19	1	W-11	14	V12
20	1	W-11	14	V12
21	2	W-11	14	W27
22	2	W-11	14	W27
23	3	W-11	14	V42
24	3	W-11	14	V42
25	1	K-12	4	V12
26	1	K-12	4	V12
27	2	K-12	4	W27
28	3	K-12	4	V42
29	3	K-12	4	V42

RED FORCE

SQD	OWNER	TYPE	NUM	LOCATION
1	1	F-01	24	P12
2	1	F-01	24	P12
3	2	F-01	24	O27
4	2	F-01	24	O27
5	3	F-01	24	P42
6	3	F-01	24	P42
7	1	F-03	24	P12
8	1	F-03	24	P12
9	2	F-03	24	O27
10	2	F-03	24	O27
11	3	F-03	24	P42
12	3	F-03	24	P42
13	1	B-40	12	P12
14	1	B-40	12	P12
15	2	B-40	12	O27
16	2	B-40	12	O27
17	3	B-40	12	P42
18	3	B-40	12	P42
19	1	W-11	14	P12
20	2	W-11	14	O27
21	1	W-11	14	P12
22	2	W-11	14	O27
23	3	W-11	14	P42
24	3	W-11	14	P42
25	1	K-12	4	P12
26	2	K-12	4	O27
27	1	K-12	4	P12
28	3	K-12	4	P42
29	3	K-12	4	P42

STATUS OF SAMS

BLUE FORCE

ID	TYPE	LOC	NUM LAUNCHERS	NUM SPARES
1	S-02	V12	24	24
2	S-02	W27	24	24
3	S-02	V42	24	24
4	S-02	EE25	24	24
5	S-02	DD14	24	24
6	S-02	DD36	24	24
7	S-02	AA9	24	24
8	S-02	BB46	24	24
9	S-02	Z28	24	24
10	S-02	JJ24	24	24

RED FORCE

ID	TYPE	LOC	NUM LAUNCHERS	NUM SPARES
1	S-02	P12	24	24
2	S-02	O27	24	24
3	S-02	P42	24	24
4	S-02	G25	24	24
5	S-02	H14	24	24
6	S-02	H36	24	24
7	S-02	K9	24	24
8	S-02	J46	24	24
9	S-02	L28	24	24
10	S-02	B24	24	24

LAND-BASED TARGET STATUS

BLUE TARGETS

ID	TYPE	SAMs	LOC	DAMAGE	HARDNESS
0	Command Center	4	EE25	0	500
1	Airfield	1	V12	0	1000
2	Airfield	2	W27	0	1000
3	Airfield	3	V42	0	1000
4	Radar	5	DD14	0	300
5	Radar	6	DD36	0	300
6	City	0	JJ6	0	300
7	City	0	JJ48	0	300
8	POL Site	7	AA9	0	200
9	POL Site	9	Z28	0	200

ID	TYPE	SAMs	LOC	DAMAGE	HARDNESS
10	POL Site	8	BB46	0	200
11	Ammo Depot	0	Z18	0	600
12	Ammo Depot	0	Y39	0	600
13	Rail Center	0	FF8	0	200
14	Rail Center	0	BB40	0	200
15	Rail Center	0	II43	0	200
16	Dam	0	U7	0	200
17	Dam	0	X32	0	200
18	Dam	0	II9	0	200
19	Dam	0	CC19	0	200
20	Staging Area	0	EE43	0	200
21	Staging Area	0	CC9	0	200
22	Staging Area	0	Y23	0	200
23	Power Plant	10	JJ24	0	300
24	Power Plant	0	GG11	0	300
25	Industrial Area	0	HH50	0	400
26	Industrial Area	0	GG31	0	400
27	Industrial Area	0	AA3	0	400
28	Bridge	0	V8	0	100
29	Bridge	0	Y33	0	100
30	Bridge	0	EE19	0	100

RED TARGETS

ID	TYPE	SAMs	LOC	DAMAGE	HARDNESS
0	Command Center	4	G25	0	500
1	Airfield	1	P12	0	1000
2	Airfield	2	O27	0	1000
3	Airfield	3	P42	0	1000
4	Radar	5	H14	0	300
5	Radar	6	H36	0	300
6	City	0	B6	0	300
7	City	0	B48	0	300
8	POL Site	7	K9	0	200
9	POL Site	9	L28	0	200
10	POL Site	8	J46	0	200
11	Ammo Depot	0	L18	0	600
12	Ammo Depot	0	M39	0	600
13	Rail Center	0	F8	0	200
14	Rail Center	0	J40	0	200
15	Rail Center	0	C43	0	200
16	Dam	0	Q7	0	200
17	Dam	0	N32	0	200
18	Dam	0	C9	0	200
19	Dam	0	I19	0	200
20	Staging Area	0	G43	0	200
21	Staging Area	0	I9	0	200
22	Staging Area	0	M23	0	200

ID	TYPE	SAMs	LOC	DAMAGE	HARDNESS
23	Power Plant	10	B24	0	300
24	Power Plant	0	E11	0	300
25	Industrial Area	0	D50	0	400
26	Industrial Area	0	E31	0	400
27	Industrial Area	0	K3	0	400
28	Bridge	0	P8	0	100
29	Bridge	0	M33	0	100
30	Bridge	0	G19	0	100

Scenario #2

AIRCRAFT SQUADRON RESOURCES

BLUE FORCE

SQD	OWNER	TYPE	NUM	LOCATION
1	1	F-02	24	V12
2	2	F-02	24	W27
3	3	F-02	24	V42
4	1	F-30	20	V12
5	2	F-30	20	W27
6	3	F-30	20	V42
7	1	F-01	18	V12
8	2	F-01	18	W27
9	3	F-01	18	V42
10	1	B-60	10	V12
11	2	B-60	10	W27
12	3	B-60	10	V42
13	1	B-60	10	V12
14	3	B-60	10	V42
15	2	B-60	10	W27
16	1	F-01	24	V12
17	2	F-01	24	W27
18	3	F-01	24	V42
19	1	W-11	15	V12
20	2	W-11	15	W27
21	1	W-11	15	V12
22	3	W-11	15	V42
23	2	W-11	15	W27
24	3	W-11	15	V42
25	1	K-12	3	V12
26	2	K-12	3	W27
27	2	K-12	3	W27
28	3	K-12	3	V42

RED FORCE

SQD	OWNER	TYPE	NUM	LOCATION
1	1	F-02	24	P12
2	2	F-02	24	O27
3	3	F-02	24	P42
4	1	F-30	20	P12
5	2	F-30	20	O27
6	3	F-30	20	P42
7	1	F-01	18	P12
8	2	F-01	18	O27
9	3	F-01	18	P42
10	1	B-60	10	P12
11	2	B-60	10	O27
12	3	B-60	10	P42
13	1	B-60	10	P12
14	3	B-60	10	P42
15	2	B-60	10	O27
16	1	F-01	24	P12
17	2	F-01	24	O27
18	3	F-01	24	P42
19	1	W-11	15	P12
20	2	W-11	15	O27
21	1	W-11	15	P12
22	3	W-11	15	P42
23	2	W-11	15	O27
24	3	W-11	15	P42
25	1	K-12	3	P12
26	2	K-12	3	O27
27	2	K-12	3	O27
28	3	K-12	3	P42

STATUS OF SAMS

BLUE FORCE

ID	TYPE	LOC	NUM LAUNCHERS	NUM SPARES
1	S-01	U7	15	50
2	S-01	CC19	15	50
3	S-01	II9	15	50
4	S-01	JJ48	15	50
5	S-01	JJ6	15	50
6	S-01	EE19	15	50
7	S-01	II43	15	50
8	S-01	FF8	15	50
9	S-01	HH50	15	50
10	S-01	AA3	15	50
11	S-02	EE25	24	50

ID	TYPE	LOC	NUM LAUNCHERS	NUM SPARES
12	S-02	V42	24	50
13	S-02	W27	24	50
14	S-02	V12	24	50
15	S-02	Y39	24	50
16	S-02	Z18	24	50
17	S-02	BB46	24	50
18	S-02	DD14	24	50
19	S-02	DD36	24	50
20	S-02	AA9	24	50

RED FORCE

ID	TYPE	LOC	NUM LAUNCHERS	NUM SPARES
1	S-01	Q7	15	50
2	S-01	I19	15	50
3	S-01	C9	15	50
4	S-01	B48	15	50
5	S-01	B6	15	50
6	S-01	G19	15	50
7	S-01	C43	15	50
8	S-01	F8	15	50
9	S-01	D50	15	50
10	S-01	K3	15	50
11	S-02	G25	24	50
12	S-02	P42	24	50
13	S-02	O27	24	50
14	S-02	P12	24	50
15	S-02	M39	24	50
16	S-02	L18	24	50
17	S-02	J46	24	50
18	S-02	H14	24	50
19	S-02	H36	24	50
20	S-02	K9	24	50

LAND-BASED TARGET STATUS

BLUE TARGETS

ID	TYPE	SAMs	LOC	DAMAGE	HARDNESS
0	Command Center	11	EE25	0	500
1	Airfield	14	V12	0	1000
2	Airfield	13	W27	0	1000
3	Airfield	12	V42	0	1000
4	Radar	18	DD14	0	300

ID	TYPE	SAMs	LOC	DAMAGE	HARDNESS
5	Radar	19	DD36	0	300
6	City	5	JJ6	0	300
7	City	4	JJ48	0	300
8	POL Site	20	AA9	0	200
9	POL Site	0	Z28	0	200
10	POL Site	17	BB46	0	200
11	Ammo Depot	16	Z18	0	600
12	Ammo Depot	15	Y39	0	600
13	Rail Center	8	FF8	0	200
14	Rail Center	0	BB40	0	200
15	Rail Center	7	II43	0	200
16	Dam	1	U7	0	200
17	Dam	0	X32	0	200
18	Dam	3	II9	0	200
19	Dam	2	CC19	0	200
20	Staging Area	0	EE43	0	200
21	Staging Area	0	CC9	0	200
22	Staging Area	0	Y23	0	200
23	Power Plant	0	JJ24	0	300
24	Power Plant	0	GG11	0	300
25	Industrial Area	9	HH50	0	400
26	Industrial Area	0	GG31	0	400
27	Industrial Area	10	AA3	0	400
28	Bridge	0	V8	0	100
29	Bridge	0	Y33	0	100
30	Bridge	6	EE19	0	100

RED TARGETS

ID	TYPE	SAMs	LOC	DAMAGE	HARDNESS
0	Command Center	11	G25	0	500
1	Airfield	14	P12	0	1000
2	Airfield	13	O27	0	1000
3	Airfield	12	P42	0	1000
4	Radar	18	H14	0	300
5	Radar	19	H36	0	300
6	City	5	B6	0	300
7	City	4	B48	0	300
8	POL Site	20	K9	0	200
9	POL Site	0	L28	0	200
10	POL Site	17	J46	0	200
11	Ammo Depot	16	L18	0	600
12	Ammo Depot	15	M39	0	600
13	Rail Center	8	F8	0	200
14	Rail Center	0	J40	0	200
15	Rail Center	7	C43	0	200
16	Dam	1	Q7	0	200
17	Dam	0	N32	0	200

ID	TYPE	SAMs	LOC	DAMAGE	HARDNESS
18	Dam	3	C9	0	200
19	Dam	2	I19	0	200
20	Staging Area	0	G43	0	200
21	Staging Area	0	I9	0	200
22	Staging Area	0	M23	0	200
23	Power Plant	0	B24	0	300
24	Power Plant	0	E11	0	300
25	Industrial Area	9	D50	0	400
26	Industrial Area	0	E31	0	400
27	Industrial Area	10	K3	0	400
28	Bridge	0	P8	0	100
29	Bridge	0	M33	0	100
30	Bridge	6	G19	0	100

Scenario #3

AIRCRAFT SQUADRON RESOURCES

BLUE FORCE

SQD	OWNER	TYPE	NUM	LOCATION
1	1	F-02	24	V12
2	2	F-02	24	W27
3	3	F-02	24	V42
4	1	F-30	20	V12
5	2	F-30	20	W27
6	3	F-30	20	V42
7	1	F-01	18	V12
8	2	F-01	18	W27
9	3	F-01	18	V42
10	1	B-60	10	V12
11	2	B-60	10	W27
12	3	B-60	10	V42
13	1	B-60	10	V12
14	3	B-60	10	V42
15	2	B-60	10	W27
16	1	F-01	24	V12
17	2	F-01	24	W27
18	3	F-01	24	V42
19	1	W-11	15	V12
20	2	W-11	15	W27
21	1	W-11	15	V12
22	3	W-11	15	V42
23	2	W-11	15	W27
24	3	W-11	15	V42
25	1	K-12	3	V12

SQD	OWNER	TYPE	NUM	LOCATION
26	2	K-12	3	W27
27	2	K-12	3	W27
28	3	K-12	3	V42

RED FORCE

SQD	OWNER	TYPE	NUM	LOCATION
1	1	F-02	24	P12
2	2	F-02	24	O27
3	3	F-02	24	P42
4	1	F-30	20	P12
5	2	F-30	20	O27
6	3	F-30	20	P42
7	1	F-01	18	P12
8	2	F-01	18	O27
9	3	F-01	18	P42
10	1	B-60	10	P12
11	2	B-60	10	O27
12	3	B-60	10	P42
13	1	B-60	10	P12
14	3	B-60	10	P42
15	2	B-60	10	O27
16	1	F-01	24	P12
17	2	F-01	24	O27
18	3	F-01	24	P42
19	1	W-11	15	P12
20	2	W-11	15	O27
21	1	W-11	15	P12
22	3	W-11	15	P42
23	2	W-11	15	O27
24	3	W-11	15	P42
25	1	K-12	3	P12
26	2	K-12	3	O27
27	2	K-12	3	O27
28	3	K-12	3	P42

STATUS OF SAMS

BLUE FORCE

ID	TYPE	LOC	NUM LAUNCHERS	NUM SPARES
1	S-01	U7	15	50
2	S-01	CC19	15	50
3	S-01	II9	15	50
4	S-01	JJ48	15	50
5	S-01	JJ6	15	50

ID	TYPE	LOC	NUM LAUNCHERS	NUM SPARES
6	S-01	EE19	15	50
7	S-01	II43	15	50
8	S-01	FF8	15	50
9	S-01	HH50	15	50
10	S-01	AA3	15	50
11	S-02	EE25	24	50
12	S-02	V42	24	50
13	S-02	W27	24	50
14	S-02	V12	24	50
15	S-02	Y39	24	50
16	S-02	Z18	24	50
17	S-02	BB46	24	50
18	S-02	DD14	24	50
19	S-02	DD36	24	50
20	S-02	AA9	24	50

RED FORCE

ID	TYPE	LOC	NUM LAUNCHERS	NUM SPARES
1	S-01	Q7	15	50
2	S-01	I19	15	50
3	S-01	C9	15	50
4	S-01	B48	15	50
5	S-01	B6	15	50
6	S-01	G19	15	50
7	S-01	C43	15	50
8	S-01	F8	15	50
9	S-01	D50	15	50
10	S-01	K3	15	50
11	S-02	G25	24	50
12	S-02	P42	24	50
13	S-02	O27	24	50
14	S-02	P12	24	50
15	S-02	M39	24	50
16	S-02	L18	24	50
17	S-02	J46	24	50
18	S-02	H14	24	50
19	S-02	H36	24	50
20	S-02	K9	24	50

LAND-BASED TARGET STATUS

BLUE TARGETS

ID	TYPE	SAMs	LOC	DAMAGE	HARDNESS
0	Command Center	11	EE25	0	500
1	Airfield	14	V12	0	1000
2	Airfield	13	W27	0	1000
3	Airfield	12	V42	0	1000
4	Radar	18	DD14	0	300
5	Radar	19	DD36	0	300
6	City	5	JJ6	0	300
7	City	4	JJ48	0	300
8	POL Site	20	AA9	0	200
9	POL Site	0	Z28	0	200
10	POL Site	17	BB46	0	200
11	Ammo Depot	16	Z18	0	600
12	Ammo Depot	15	Y39	0	600
13	Rail Center	8	FF8	0	200
14	Rail Center	0	BB40	0	200
15	Rail Center	7	II43	0	200
16	Dam	1	U7	0	200
17	Dam	0	X32	0	200
18	Dam	3	II9	0	200
19	Dam	2	CC19	0	200
20	Staging Area	0	EE43	0	200
21	Staging Area	0	CC9	0	200
22	Staging Area	0	Y23	0	200
23	Power Plant	0	JJ24	0	300
24	Power Plant	0	GG11	0	300
25	Industrial Area	9	HH50	0	400
26	Industrial Area	0	GG31	0	400
27	Industrial Area	10	AA3	0	400
28	Bridge	0	V8	0	100
29	Bridge	0	Y33	0	100
30	Bridge	6	EE19	0	100

RED TARGETS

ID	TYPE	SAMs	LOC	DAMAGE	HARDNESS
0	Command Center	11	G25	0	500
1	Airfield	14	P12	0	1000
2	Airfield	13	O27	0	1000
3	Airfield	12	P42	0	1000
4	Radar	18	H14	0	300
5	Radar	19	H36	0	300
6	City	5	B6	0	300
7	City	4	B48	0	300
8	POL Site	20	K9	0	200

ID	TYPE	SAMs	LOC	DAMAGE	HARDNESS
9	POL Site	0	L28	0	200
10	POL Site	17	J46	0	200
11	Ammo Depot	16	L18	0	600
12	Ammo Depot	15	M39	0	600
13	Rail Center	8	F8	0	200
14	Rail Center	0	J40	0	200
15	Rail Center	7	C43	0	200
16	Dam	1	Q7	0	200
17	Dam	0	N32	0	200
18	Dam	3	C9	0	200
19	Dam	2	I19	0	200
20	Staging Area	0	G43	0	200
21	Staging Area	0	I9	0	200
22	Staging Area	0	M23	0	200
23	Power Plant	0	B24	0	300
24	Power Plant	0	E11	0	300
25	Industrial Area	9	D50	0	400
26	Industrial Area	0	E31	0	400
27	Industrial Area	10	K3	0	400
28	Bridge	0	P8	0	100
29	Bridge	0	M33	0	100
30	Bridge	6	G19	0	100

Scenario #4

AIRCRAFT SQUADRON RESOURCES

BLUE FORCE

SQD	OWNER	TYPE	NUM	LOCATION
1	1	F-01	24	V6
2	2	F-01	24	W19
3	3	F-01	24	W39
4	1	F-02	20	V6
5	2	F-02	20	W19
6	1	F-03	18	V6
7	2	F-03	18	W19
8	3	F-20	24	W39
9	3	F-20	24	W39
10	3	F-20	24	W39
11	3	F-20	24	W39
12	1	B-40	12	V6
13	1	B-40	12	V6
14	2	B-40	12	W19
15	2	B-40	12	W19

SQD	OWNER	TYPE	NUM	LOCATION
16	3	B-40	12	W39
17	3	W-11	10	W39
18	3	W-11	10	W39
19	2	W-11	10	W19
20	2	W-11	10	W19
21	1	W-11	10	V6
22	1	W-11	10	V6
23	1	K-12	4	V6
24	2	K-12	4	W19
25	3	K-12	4	W39
26	3	K-12	4	W39

RED FORCE

SQD	OWNER	TYPE	NUM	LOCATION
1	1	F-01	24	P6
2	2	F-01	24	O19
3	3	F-01	24	O39
4	1	F-02	20	P6
5	2	F-02	20	O19
6	1	F-03	18	P6
7	2	F-03	18	O19
8	3	F-20	24	O39
9	3	F-20	24	O39
10	3	F-20	24	O39
11	3	F-20	24	O39
12	1	B-40	12	P6
13	1	B-40	12	P6
14	2	B-40	12	O19
15	2	B-40	12	O19
16	3	B-40	12	O39
17	3	W-11	10	O39
18	3	W-11	10	O39
19	2	W-11	10	O19
20	2	W-11	10	O19
21	1	W-11	10	P6
22	1	W-11	10	P6
23	1	K-12	4	P6
24	2	K-12	4	O19
25	3	K-12	4	O39
26	3	K-12	4	O39

STATUS OF SAMS

BLUE FORCE

ID	TYPE	LOC	NUM LAUNCHERS	NUM SPARES
1	S-02	W39	20	40
2	S-02	W19	20	40
3	S-02	V6	20	40
4	S-02	EE31	20	40
5	S-02	BB32	20	40
6	S-02	AA11	20	40
7	S-02	AA43	20	40
8	S-02	AA17	20	40
9	S-02	Z26	20	40
10	S-02	BB22	20	40
11	S-02	BB14	20	40
12	S-02	EE7	20	40
13	S-02	HH22	20	40
14	S-02	Y45	20	40

RED FORCE

ID	TYPE	LOC	NUM LAUNCHERS	NUM SPARES
1	S-02	O39	20	40
2	S-02	O19	20	40
3	S-02	P6	20	40
4	S-02	G31	20	40
5	S-02	J32	20	40
6	S-02	K11	20	40
7	S-02	K43	20	40
8	S-02	K17	20	40
9	S-02	L26	20	40
10	S-02	J22	20	40
11	S-02	J14	20	40
12	S-02	G7	20	40
13	S-02	D22	20	40
14	S-02	M45	20	40

LAND-BASED TARGET STATUS

BLUE TARGETS

ID	TYPE	SAMs	LOC	DAMAGE	HARDNESS
0	Command Center	4	EE31	0	500
1	Airfield	3	V6	0	1000

ID	TYPE	SAMs	LOC	DAMAGE	HARDNESS
2	Airfield	2	W19	0	1000
3	Airfield	1	W39	0	1000
4	Radar	6	AA11	0	300
5	Radar	5	BB32	0	300
6	City	0	II7	0	300
7	City	0	GG25	0	300
8	City	0	II43	0	300
9	POL Site	8	AA17	0	200
10	POL Site	7	AA43	0	200
11	POL Site	0	DD38	0	200
12	POL Site	0	HH14	0	200
13	Ammo Depot	11	BB14	0	600
14	Ammo Depot	9	Z26	0	600
15	Ammo Depot	0	FF46	0	600
16	Rail Center	0	DD20	0	200
17	Rail Center	0	FF36	0	200
18	Dam	0	T12	0	200
19	Dam	0	W27	0	200
20	Dam	0	HH30	0	200
21	Staging Area	14	Y45	0	200
22	Staging Area	12	EE7	0	200
23	Staging Area	10	BB22	0	200
24	Power Plant	13	HH22	0	300
25	Power Plant	0	HH34	0	300
26	Industrial Area	0	HH48	0	400
27	Industrial Area	0	EE13	0	400
28	Industrial Area	0	AA3	0	400
29	Bridge	0	U13	0	100
30	Bridge	0	X24	0	100
31	Bridge	0	X44	0	100
32	Bridge	0	GG17	0	100
33	Bridge	0	CC47	0	100

RED TARGETS

ID	TYPE	SAMs	LOC	DAMAGE	HARDNESS
0	Command Center	4	G31	0	500
1	Airfield	3	P6	0	1000
2	Airfield	2	O19	0	1000
3	Airfield	1	O39	0	1000
4	Radar	6	K11	0	300
5	Radar	5	J32	0	300
6	City	0	C7	0	300
7	City	0	E25	0	300
8	City	0	C43	0	300
9	POL Site	8	K17	0	200
10	POL Site	7	K43	0	200
11	POL Site	0	H38	0	200

ID	TYPE	SAMs	LOC	DAMAGE	HARDNESS
12	POL Site	0	D14	0	200
13	Ammo Depot	11	J14	0	600
14	Ammo Depot	9	L26	0	600
15	Ammo Depot	0	F46	0	600
16	Rail Center	0	H20	0	200
17	Rail Center	0	F36	0	200
18	Dam	0	R12	0	200
19	Dam	0	O27	0	200
20	Dam	0	D30	0	200
21	Staging Area	14	M45	0	200
22	Staging Area	12	G7	0	200
23	Staging Area	10	J22	0	200
24	Power Plant	13	D22	0	300
25	Power Plant	0	D34	0	300
26	Industrial Area	0	D48	0	400
27	Industrial Area	0	G13	0	400
28	Industrial Area	0	K3	0	400
29	Bridge	0	Q13	0	100
30	Bridge	0	N24	0	100
31	Bridge	0	N44	0	100
32	Bridge	0	E17	0	100
33	Bridge	0	I47	0	100

Scenario #5

AIRCRAFT SQUADRON RESOURCES

BLUE FORCE

SQD	OWNER	TYPE	NUM	LOCATION
1	1	F-01	24	V6
2	2	F-01	24	W19
3	3	F-01	24	W39
4	1	F-02	20	V6
5	2	F-02	20	W19
6	1	F-03	18	V6
7	2	F-03	18	W19
8	3	F-20	24	W39
9	3	F-20	24	W39
10	3	F-20	24	W39
11	3	F-20	24	W39
12	1	B-40	12	V6
13	1	B-40	12	V6
14	2	B-40	12	W19

SQD	OWNER	TYPE	NUM	LOCATION
15	2	B-40	12	W19
16	3	B-40	12	W39
17	3	W-11	10	W39
18	3	W-11	10	W39
19	2	W-11	10	W19
20	2	W-11	10	W19
21	1	W-11	10	V6
22	1	W-11	10	V6
23	1	K-12	4	V6
24	2	K-12	4	W19
25	3	K-12	4	W39
26	3	K-12	4	W39

RED FORCE

SQD	OWNER	TYPE	NUM	LOCATION
1	1	F-01	24	P6
2	2	F-01	24	O19
3	3	F-01	24	O39
4	1	F-02	20	P6
5	2	F-02	20	O19
6	1	F-03	18	P6
7	2	F-03	18	O19
8	3	F-20	24	O39
9	3	F-20	24	O39
10	3	F-20	24	O39
11	3	F-20	24	O39
12	1	B-40	12	P6
13	1	B-40	12	P6
14	2	B-40	12	O19
15	2	B-40	12	O19
16	3	B-40	12	O39
17	3	W-11	10	O39
18	3	W-11	10	O39
19	2	W-11	10	O19
20	2	W-11	10	O19
21	1	W-11	10	P6
22	1	W-11	10	P6
23	1	K-12	4	P6
24	2	K-12	4	O19
25	3	K-12	4	O39
26	3	K-12	4	O39

STATUS OF SAMS

BLUE FORCE

ID	TYPE	LOC	NUM LAUNCHERS	NUM SPARES
1	S-02	W39	20	40
2	S-02	W19	20	40
3	S-02	V6	20	40
4	S-02	EE31	20	40
5	S-02	BB32	20	40
6	S-02	AA11	20	40
7	S-02	AA43	20	40
8	S-02	AA17	20	40
9	S-02	Z26	20	40
10	S-02	BB22	20	40
11	S-02	BB14	20	40
12	S-02	EE7	20	40
13	S-02	HH22	20	40
14	S-02	Y45	20	40

RED FORCE

ID	TYPE	LOC	NUM LAUNCHERS	NUM SPARES
1	S-02	O39	20	40
2	S-02	O19	20	40
3	S-02	P6	20	40
4	S-02	G31	20	40
5	S-02	J32	20	40
6	S-02	K11	20	40
7	S-02	K43	20	40
8	S-02	K17	20	40
9	S-02	L26	20	40
10	S-02	J22	20	40
11	S-02	J14	20	40
12	S-02	G7	20	40
13	S-02	D22	20	40
14	S-02	M45	20	40

LAND-BASED TARGET STATUS

BLUE TARGETS

ID	TYPE	SAMs	LOC	DAMAGE	HARDNESS
0	Command Center	4	EE31	0	500
1	Airfield	3	V6	0	1000
2	Airfield	2	W19	0	1000
3	Airfield	1	W39	0	1000
4	Radar	6	AA11	0	300
5	Radar	5	BB32	0	300
6	City	0	II7	0	300
7	City	0	GG25	0	300
8	City	0	II43	0	300
9	POL Site	8	AA17	0	200
10	POL Site	7	AA43	0	200
11	POL Site	0	DD38	0	200
12	POL Site	0	HH14	0	200
13	Ammo Depot	11	BB14	0	600
14	Ammo Depot	9	Z26	0	600
15	Ammo Depot	0	FF46	0	600
16	Rail Center	0	DD20	0	200
17	Rail Center	0	FF36	0	200
18	Dam	0	T12	0	200
19	Dam	0	W27	0	200
20	Dam	0	HH30	0	200
21	Staging Area	14	Y45	0	200
22	Staging Area	12	EE7	0	200
23	Staging Area	10	BB22	0	200
24	Power Plant	13	HH22	0	300
25	Power Plant	0	HH34	0	300
26	Industrial Area	0	HH48	0	400
27	Industrial Area	0	EE13	0	400
28	Industrial Area	0	AA3	0	400
29	Bridge	0	U13	0	100
30	Bridge	0	X24	0	100
31	Bridge	0	X44	0	100
32	Bridge	0	GG17	0	100
33	Bridge	0	CC47	0	100

RED TARGETS

ID	TYPE	SAMs	LOC	DAMAGE	HARDNESS
0	Command Center	4	G31	0	500
1	Airfield	3	P6	0	1000
2	Airfield	2	O19	0	1000
3	Airfield	1	O39	0	1000
4	Radar	6	K11	0	300
5	Radar	5	J32	0	300

ID	TYPE	SAMs	LOC	DAMAGE	HARDNESS
6	City	0	C7	0	300
7	City	0	E25	0	300
8	City	0	C43	0	300
9	POL Site	8	K17	0	200
10	POL Site	7	K43	0	200
11	POL Site	0	H38	0	200
12	POL Site	0	D14	0	200
13	Ammo Depot	11	J14	0	600
14	Ammo Depot	9	L26	0	600
15	Ammo Depot	0	F46	0	600
16	Rail Center	0	H20	0	200
17	Rail Center	0	F36	0	200
18	Dam	0	R12	0	200
19	Dam	0	O27	0	200
20	Dam	0	D30	0	200
21	Staging Area	14	M45	0	200
22	Staging Area	12	G7	0	200
23	Staging Area	10	J22	0	200
24	Power Plant	13	D22	0	300
25	Power Plant	0	D34	0	300
26	Industrial Area	0	D48	0	400
27	Industrial Area	0	G13	0	400
28	Industrial Area	0	K3	0	400
29	Bridge	0	Q13	0	100
30	Bridge	0	N24	0	100
31	Bridge	0	N44	0	100
32	Bridge	0	E17	0	100
33	Bridge	0	I47	0	100

Scenario #6

AIRCRAFT SQUADRON RESOURCES

BLUE FORCE

SQD	OWNER	TYPE	NUM	LOCATION
1	1	F-01	24	V10
2	1	F-01	24	V10
3	1	F-01	24	V10
4	1	F-02	24	V10
5	2	F-02	24	X28
6	3	F-02	24	U47
7	2	F-03	24	X28
8	3	F-03	24	U47
9	2	F-03	24	X28

SQD	OWNER	TYPE	NUM	LOCATION
10	3	F-10	20	U47
11	3	F-10	20	U47
12	2	F-10	20	X28
13	2	F-30	20	X28
14	1	F-30	20	V10
15	3	B-50	15	U47
16	3	B-50	15	U47
17	2	B-50	15	X28
18	2	B-60	10	X28
19	2	B-60	10	X28
20	2	B-60	10	X28
21	1	B-40	12	V10
22	1	B-40	12	V10
23	1	B-40	12	V10
24	1	W-11	9	V10
25	2	W-11	9	X28
26	3	W-11	9	U47
27	3	W-11	9	U47
28	3	K-12	3	U47
29	2	K-12	3	X28
30	1	K-12	3	V10

RED FORCE

SQD	OWNER	TYPE	NUM	LOCATION
1	1	F-01	24	P10
2	1	F-01	24	P10
3	1	F-01	24	P10
4	1	F-02	24	P10
5	2	F-02	24	N28
6	3	F-02	24	Q47
7	2	F-03	24	N28
8	3	F-03	24	Q47
9	2	F-03	24	N28
10	3	F-10	20	Q47
11	3	F-10	20	Q47
12	2	F-10	20	N28
13	2	F-30	20	N28
14	1	F-30	20	P10
15	3	B-50	15	Q47
16	3	B-50	15	Q47
17	2	B-50	15	N28
18	2	B-60	10	N28
19	2	B-60	10	N28
20	2	B-60	10	N28
21	1	B-40	12	P10
22	1	B-40	12	P10
23	1	B-40	12	P10

SQD	OWNER	TYPE	NUM	LOCATION
24	1	W-11	9	P10
25	2	W-11	9	N28
26	3	W-11	9	Q47
27	3	W-11	9	Q47
28	3	K-12	3	Q47
29	2	K-12	3	N28
30	1	K-12	3	P10

STATUS OF SAMS

BLUE FORCE

ID	TYPE	LOC	NUM LAUNCHERS	NUM SPARES
1	S-01	W43	24	35
2	S-01	V18	24	35
3	S-01	FF6	24	35
4	S-01	FF48	24	35
5	S-01	T4	24	35
6	S-02	GG17	20	40
7	S-02	V10	20	40
8	S-02	X28	20	40
9	S-02	U47	20	40
10	S-02	Z32	20	40
11	S-02	AA13	20	40
12	S-02	Z8	20	40
13	S-02	Y41	20	40
14	S-02	BB38	20	40
15	S-02	CC21	20	40

RED FORCE

ID	TYPE	LOC	NUM LAUNCHERS	NUM SPARES
1	S-01	O43	24	35
2	S-01	P18	24	35
3	S-01	F6	24	35
4	S-01	F48	24	35
5	S-01	R4	24	35
6	S-02	E17	20	40
7	S-02	P10	20	40
8	S-02	N28	20	40
9	S-02	Q47	20	40
10	S-02	L32	20	40
11	S-02	K13	20	40
12	S-02	L8	20	40
13	S-02	M41	20	40

ID	TYPE	LOC	NUM LAUNCHERS	NUM SPARES
14	S-02	J38	20	40
15	S-02	I21	20	40

LAND-BASED TARGET STATUS

BLUE TARGETS

ID	TYPE	SAMs	LOC	DAMAGE	HARDNESS
0	Command Center	6	GG17	0	500
1	Airfield	7	V10	0	1000
2	Airfield	8	X28	0	1000
3	Airfield	9	U47	0	1000
4	Radar	11	AA13	0	300
5	Radar	10	Z32	0	300
6	Radar	0	AA47	0	300
7	City	0	II5	0	300
8	City	0	GG29	0	300
9	City	0	JJ50	0	300
10	PQL Site	12	Z8	0	200
11	POL Site	15	CC21	0	200
12	POL Site	14	BB38	0	200
13	POL Site	0	HH40	0	200
14	Ammo Depot	0	CC9	0	600
15	Ammo Depot	0	CC27	0	600
16	Ammo Depot	13	Y41	0	600
17	Ammo Depot	0	JJ24	0	600
18	Rail Center	0	II11	0	200
19	Rail Center	0	DD32	0	200
20	Rail Center	0	EE47	0	200
21	Dam	0	V32	0	200
22	Dam	2	V18	0	200
23	Dam	0	T50	0	200
24	Dam	0	JJ30	0	200
25	Staging Area	0	AA43	0	200
26	Staging Area	0	FF12	0	200
27	Staging Area	0	Z22	0	200
28	Power Plant	3	FF6	0	300
29	Power Plant	0	HH34	0	300
30	Industrial Area	0	JJ42	0	400
31	Industrial Area	0	II17	0	400
32	Bridge	0	V28	0	100
33	Bridge	5	T4	0	100
34	Bridge	4	FF48	0	100
35	Bridge	1	W43	0	100

RED TARGETS

ID	TYPE	SAMs	LOC	DAMAGE	HARDNESS
0	Command Center	6	E17	0	500
1	Airfield	7	P10	0	1000
2	Airfield	8	N28	0	1000
3	Airfield	9	Q47	0	1000
4	Radar	11	K13	0	300
5	Radar	10	L32	0	300
6	Radar	0	K47	0	300
7	City	0	C5	0	300
8	City	0	E29	0	300
9	City	0	B50	0	300
10	POL Site	12	L8	0	200
11	POL Site	15	I21	0	200
12	POL Site	14	J38	0	200
13	POL Site	0	D40	0	200
14	Ammo Depot	0	I9	0	600
15	Ammo Depot	0	I27	0	600
16	Ammo Depot	13	M41	0	600
17	Ammo Depot	0	B24	0	600
18	Rail Center	0	C11	0	200
19	Rail Center	0	H32	0	200
20	Rail Center	0	G47	0	200
21	Dam	0	P32	0	200
22	Dam	2	P18	0	200
23	Dam	0	R50	0	200
24	Dam	0	B30	0	200
25	Staging Area	0	K43	0	200
26	Staging Area	0	F12	0	200
27	Staging Area	0	L22	0	200
28	Power Plant	3	F6	0	300
29	Power Plant	0	D34	0	300
30	Industrial Area	0	B42	0	400
31	Industrial Area	0	C17	0	400
32	Bridge	0	P28	0	100
33	Bridge	5	R4	0	100
34	Bridge	4	F48	0	100
35	Bridge	1	O43	0	100

Scenario #7

AIRCRAFT SQUADRON RESOURCES

BLUE FORCE

SQD	OWNER	TYPE	NUM	LOCATION
1	1	F-01	24	V10
2	1	F-01	24	V10
3	1	F-01	24	V10
4	2	F-02	24	U23
5	2	F-02	24	U23
6	3	F-02	24	U43
7	2	F-03	15	U23
8	3	F-03	15	U43
9	3	F-03	15	U43
10	2	F-10	15	U23
11	2	F-10	15	U23
12	3	F-10	15	U43
13	1	B-40	15	V10
14	1	B-40	15	V10
15	2	B-40	15	U23
16	3	B-50	15	U43
17	3	B-50	15	U43
18	3	B-50	15	U43
19	3	W-11	10	U43
20	3	W-11	10	U43
21	2	W-11	10	U23
22	2	W-11	10	U23
23	1	W-11	10	V10
24	1	W-11	10	V10
25	1	K-12	4	V10
26	1	K-12	4	V10
27	2	K-12	4	U23
28	3	K-12	4	U43
29	3	K-12	4	U43

RED FORCE

SQD	OWNER	TYPE	NUM	LOCATION
1	1	F-01	24	P10
2	1	F-01	24	P10
3	1	F-01	24	P10
4	2	F-02	24	Q23
5	2	F-02	24	Q23
6	3	F-02	24	Q43
7	2	F-03	15	Q23

SQD	OWNER	TYPE	NUM	LOCATION
8	3	F-03	15	Q43
9	3	F-03	15	Q43
10	2	F-10	15	Q23
11	2	F-10	15	Q23
12	3	F-10	15	Q43
13	1	B-40	15	P10
14	1	B-40	15	P10
15	2	B-40	15	Q23
16	3	B-50	15	Q43
17	3	B-50	15	Q43
18	3	B-50	15	Q43
19	3	W-11	10	Q43
20	3	W-11	10	Q43
21	2	W-11	10	Q23
22	2	W-11	10	Q23
23	1	W-11	10	P10
24	1	W-11	10	P10
25	1	K-12	4	P10
26	1	K-12	4	P10
27	2	K-12	4	Q23
28	3	K-12	4	Q43
29	3	K-12	4	Q43

STATUS OF SAMS

BLUE FORCE

ID	TYPE	LOC	NUM LAUNCHERS	NUM SPARES
1	S-02	DD44	17	25
2	S-02	GG35	20	40
3	S-02	U43	20	40
4	S-02	U23	20	40
5	S-02	V10	20	40
6	S-02	AA15	20	40
7	S-02	X34	20	40
8	S-02	Z22	20	40
9	S-02	Y15	20	40
10	S-02	EE39	20	40
11	S-02	FF30	20	40
12	S-02	II11	20	40
13	S-02	Z8	20	40
14	S-02	CC31	20	40
15	S-02	HH28	20	40
16	S-02	HH24	20	40
17	S-02	II5	20	40
18	S-02	T18	20	40

RED FORCE

ID	TYPE	LOC	NUM LAUNCHERS	NUM SPARES
1	S-02	H44	17	25
2	S-02	E35	20	40
3	S-02	Q43	20	40
4	S-02	Q23	20	40
5	S-02	P10	20	40
6	S-02	K15	20	40
7	S-02	N34	20	40
8	S-02	L22	20	40
9	S-02	M15	20	40
10	S-02	G39	20	40
11	S-02	F30	20	40
12	S-02	C11	20	40
13	S-02	L8	20	40
14	S-02	I31	20	40
15	S-02	D28	20	40
16	S-02	D24	20	40
17	S-02	C5	20	40
18	S-02	R18	20	40

LAND-BASED TARGET STATUS

BLUE TARGETS

ID	TYPE	SAMs	LOC	DAMAGE	HARDNESS
0	Command Center	2	GG35	0	500
1	Airfield	5	V10	0	1000
2	Airfield	4	U23	0	1000
3	Airfield	3	U43	0	1000
4	Radar	6	AA15	0	300
5	Radar	0	Z30	0	300
6	Radar	1	DD44	0	300
7	City	0	JJ50	0	300
8	City	0	JJ4	0	300
9	City	0	II21	0	300
10	POL Site	13	Z8	0	200
11	POL Site	8	Z22	0	200
12	POL Site	0	Z38	0	200
13	POL Site	15	HH28	0	200
14	Ammo Depot	9	Y15	0	600
15	Ammo Depot	7	X34	0	600
16	Ammo Depot	10	EE39	0	600
17	Ammo Depot	12	II11	0	600
18	Rail Center	0	EE11	0	200

ID	TYPE	SAMs	LOC	DAMAGE	HARDNESS
19	Rail Center	11	FF30	0	200
20	Dam	18	T18	0	200
21	Dam	0	CC23	0	200
22	Dam	0	Z48	0	200
23	Dam	0	DD6	0	200
24	Staging Area	0	BB10	0	200
25	Staging Area	0	HH42	0	200
26	Staging Area	0	AA43	0	200
27	Staging Area	17	II5	0	200
28	Power Plant	16	HH24	0	300
29	Power Plant	0	GG49	0	300
30	Industrial Area	0	FF16	0	400
31	Industrial Area	14	CC31	0	400
32	Industrial Area	0	II35	0	400
33	Bridge	0	T16	0	100
34	Bridge	0	W39	0	100
35	Bridge	0	DD14	0	100
36	Bridge	0	CC49	0	100

RED TARGETS

ID	TYPE	SAMs	LOC	DAMAGE	HARDNESS
0	Command Center	2	E35	0	500
1	Airfield	5	P10	0	1000
2	Airfield	4	Q23	0	1000
3	Airfield	3	Q43	0	1000
4	Radar	6	K15	0	300
5	Radar	0	L30	0	300
6	Radar	1	H44	0	300
7	City	0	B50	0	300
8	City	0	B4	0	300
9	City	0	C21	0	300
10	POL Site	13	L8	0	200
11	POL Site	8	L22	0	200
12	POL Site	0	L38	0	200
13	POL Site	15	D28	0	200
14	Ammo Depot	9	M15	0	600
15	Ammo Depot	7	N34	0	600
16	Ammo Depot	10	G39	0	600
17	Ammo Depot	12	C11	0	600
18	Rail Center	0	G11	0	200
19	Rail Center	11	F30	0	200
20	Dam	18	R18	0	200
21	Dam	0	I23	0	200
22	Dam	0	L48	0	200
23	Dam	0	H6	0	200
24	Staging Area	0	J10	0	200
25	Staging Area	0	D42	0	200

ID	TYPE	SAMs	LOC	DAMAGE	HARDNESS
26	Staging Area	0	K43	0	200
27	Staging Area	17	C5	0	200
28	Power Plant	16	D24	0	300
29	Power Plant	0	E49	0	300
30	Industrial Area	0	F16	0	400
31	Industrial Area	14	I31	0	400
32	Industrial Area	0	C35	0	400
33	Bridge	0	R16	0	100
34	Bridge	0	O39	0	100
35	Bridge	0	H14	0	100
36	Bridge	0	I49	0	100

Scenario #8

AIRCRAFT SQUADRON RESOURCES

BLUE FORCE

SQD	OWNER	TYPE	NUM	LOCATION
1	1	F-01	24	V10
2	1	F-01	24	V10
3	1	F-01	24	V10
4	2	F-02	24	U23
5	2	F-02	24	U23
6	3	F-02	24	U43
7	2	F-03	15	U23
8	3	F-03	15	U43
9	3	F-03	15	U43
10	2	F-10	15	U23
11	2	F-10	15	U23
12	3	F-10	15	U43
13	1	B-40	15	V10
14	1	B-40	15	V10
15	2	B-40	15	U23
16	3	B-50	15	U43
17	3	B-50	15	U43
18	3	B-50	15	U43
19	3	W-11	10	U43
20	3	W-11	10	U43
21	2	W-11	10	U23
22	2	W-11	10	U23
23	1	W-11	10	V10
24	1	W-11	10	V10
25	1	K-12	4	V10
26	1	K-12	4	V10

SQD	OWNER	TYPE	NUM	LOCATION
27	2	K-12	4	U23
28	3	K-12	4	U43
29	3	K-12	4	U43

RED FORCE

SQD	OWNER	TYPE	NUM	LOCATION
1	1	F-01	24	P10
2	1	F-01	24	P10
3	1	F-01	24	P10
4	2	F-02	24	Q23
5	2	F-02	24	Q23
6	3	F-02	24	Q43
7	2	F-03	15	Q23
8	3	F-03	15	Q43
9	3	F-03	15	Q43
10	2	F-10	15	Q23
11	2	F-10	15	Q23
12	3	F-10	15	Q43
13	1	B-40	15	P10
14	1	B-40	15	P10
15	2	B-40	15	Q23
16	3	B-50	15	Q43
17	3	B-50	15	Q43
18	3	B-50	15	Q43
19	3	W-11	10	Q43
20	3	W-11	10	Q43
21	2	W-11	10	Q23
22	2	W-11	10	Q23
23	1	W-11	10	P10
24	1	W-11	10	P10
25	1	K-12	4	P10
26	1	K-12	4	P10
27	2	K-12	4	Q23
28	3	K-12	4	Q43
29	3	K-12	4	Q43

STATUS OF SAMS

BLUE FORCE

ID	TYPE	LOC	NUM LAUNCHERS	NUM SPARES
1	S-02	DD44	17	25
2	S-02	GG35	20	40
3	S-02	U43	20	40
4	S-02	U23	20	40

ID	TYPE	LOC	NUM LAUNCHERS	NUM SPARES
5	S-02	V10	20	40
6	S-02	AA15	20	40
7	S-02	X34	20	40
8	S-02	Z22	20	40
9	S-02	Y15	20	40
10	S-02	EE39	20	40
11	S-02	FF30	20	40
12	S-02	II11	20	40
13	S-02	Z8	20	40
14	S-02	CC31	20	40
15	S-02	HH28	20	40
16	S-02	HH24	20	40
17	S-02	II5	20	40
18	S-02	T18	20	40

RED FORCE

ID	TYPE	LOC	NUM LAUNCHERS	NUM SPARES
1	S-02	H44	17	25
2	S-02	E35	20	40
3	S-02	Q43	20	40
4	S-02	Q23	20	40
5	S-02	P10	20	40
6	S-02	K15	20	40
7	S-02	N34	20	40
8	S-02	L22	20	40
9	S-02	M15	20	40
10	S-02	G39	20	40
11	S-02	F30	20	40
12	S-02	C11	20	40
13	S-02	L8	20	40
14	S-02	I31	20	40
15	S-02	D28	20	40
16	S-02	D24	20	40
17	S-02	C5	20	40
18	S-02	R18	20	40

LAND-BASED TARGET STATUS

BLUE TARGETS

ID	TYPE	SAMs	LOC	DAMAGE	HARDNESS
0	Command Center	2	GG35	0	500
1	Airfield	5	V10	0	1000
2	Airfield	4	U23	0	1000
3	Airfield	3	U43	0	1000
4	Radar	6	AA15	0	300

ID	TYPE	SAMs	LOC	DAMAGE	HARDNESS
5	Radar	0	Z30	0	300
6	Radar	1	DD44	0	300
7	City	0	JJ50	0	300
8	City	0	JJ4	0	300
9	City	0	II21	0	300
10	POL Site	13	Z8	0	200
11	POL Site	8	Z22	0	200
12	POL Site	0	Z38	0	200
13	POL Site	15	HH28	0	200
14	Ammo Depot	9	Y15	0	600
15	Ammo Depot	7	X34	0	600
16	Ammo Depot	10	EE39	0	600
17	Ammo Depot	12	II11	0	600
18	Rail Center	0	EE11	0	200
19	Rail Center	11	FF30	0	200
20	Dam	18	T18	0	200
21	Dam	0	CC23	0	200
22	Dam	0	Z48	0	200
23	Dam	0	DD6	0	200
24	Staging Area	0	BB10	0	200
25	Staging Area	0	HH42	0	200
26	Staging Area	0	AA43	0	200
27	Staging Area	17	II5	0	200
28	Power Plant	16	HH24	0	300
29	Power Plant	0	GG49	0	300
30	Industrial Area	0	FF16	0	400
31	Industrial Area	14	CC31	0	400
32	Industrial Area	0	II35	0	400
33	Bridge	0	T16	0	100
34	Bridge	0	W39	0	100
35	Bridge	0	DD14	0	100
36	Bridge	0	CC49	0	100

RED TARGETS

ID	TYPE	SAMs	LOC	DAMAGE	HARDNESS
0	Command Center	2	E35	0	500
1	Airfield	5	P10	0	1000
2	Airfield	4	Q23	0	1000
3	Airfield	3	Q43	0	1000
4	Radar	6	K15	0	300
5	Radar	0	L30	0	300
6	Radar	1	H44	0	300
7	City	0	B50	0	300
8	City	0	B4	0	300
9	City	0	C21	0	300
10	POL Site	13	L8	0	200
11	POL Site	8	L22	0	200

ID	TYPE	SAMs	LOC	DAMAGE	HARDNESS
12	POL Site	0	L38	0	200
13	POL Site	15	D28	0	200
14	Ammo Depot	9	M15	0	600
15	Ammo Depot	7	N34	0	600
16	Ammo Depot	10	G39	0	600
17	Ammo Depot	12	C11	0	600
18	Rail Center	0	G11	0	200
19	Rail Center	11	F30	0	200
20	Dam	18	R18	0	200
21	Dam	0	I23	0	200
22	Dam	0	L48	0	200
23	Dam	0	H6	0	200
24	Staging Area	0	J10	0	200
25	Staging Area	0	D42	0	200
26	Staging Area	0	K43	0	200
27	Staging Area	17	C5	0	200
28	Power Plant	16	D24	0	300
29	Power Plant	0	E49	0	300
30	Industrial Area	0	F16	0	400
31	Industrial Area	14	I31	0	400
32	Industrial Area	0	C35	0	400
33	Bridge	0	R16	0	100
34	Bridge	0	O39	0	100
35	Bridge	0	H14	0	100
36	Bridge	0	I49	0	100

Scenario #9

AIRCRAFT SQUADRON RESOURCES

BLUE FORCE

SQD	OWNER	TYPE	NUM	LOCATION
1	1	F-01	24	U7
2	1	F-01	24	U7
3	2	F-01	24	V28
4	2	F-01	24	V28
5	3	F-01	24	U41
6	3	F-01	24	U41
7	1	F-03	20	U7
8	1	F-03	20	U7
9	2	F-03	20	V28
10	2	F-03	20	V28
11	3	F-03	20	U41
12	3	F-03	20	U41

SQD	OWNER	TYPE	NUM	LOCATION
13	1	F-20	20	U7
14	1	F-20	20	U7
15	2	F-20	20	V28
16	2	F-20	20	V28
17	3	F-20	20	U41
18	3	F-20	20	U41
19	1	B-50	16	U7
20	1	B-50	16	U7
21	2	B-50	16	V28
22	2	B-50	16	V28
23	3	B-50	16	U41
24	3	B-50	16	U41
25	1	W-11	12	U7
26	2	W-11	12	V28
27	3	W-11	12	U41
28	1	K-12	4	U7
29	2	K-12	4	V28
30	3	K-12	4	U41

RED FORCE

SQD	OWNER	TYPE	NUM	LOCATION
1	1	F-01	24	Q7
2	1	F-01	24	Q7
3	2	F-01	24	P28
4	2	F-01	24	P28
5	3	F-01	24	Q41
6	3	F-01	24	Q41
7	1	F-03	20	Q7
8	1	F-03	20	Q7
9	2	F-03	20	P28
10	2	F-03	20	P28
11	3	F-03	20	Q41
12	3	F-03	20	Q41
13	1	F-20	20	Q7
14	1	F-20	20	Q7
15	2	F-20	20	P28
16	2	F-20	20	P28
17	3	F-20	20	Q41
18	3	F-20	20	Q41
19	1	B-50	16	Q7
20	1	B-50	16	Q7
21	2	B-50	16	P28
22	2	B-50	16	P28
23	3	B-50	16	Q41
24	3	B-50	16	Q41
25	1	W-11	12	Q7
26	2	W-11	12	P28

SQD	OWNER	TYPE	NUM	LOCATION
27	3	W-11	12	Q41
28	1	K-12	4	Q7
29	2	K-12	4	P28
30	3	K-12	4	Q41

STATUS OF SAMS

BLUE FORCE

ID	TYPE	LOC	NUM LAUNCHERS	NUM SPARES
1	S-02	HH14	20	40
2	S-02	U7	20	40
3	S-02	V28	20	40
4	S-02	U41	20	40
5	S-02	AA37	20	40
6	S-02	AA23	20	40
7	S-02	Y17	20	40
8	S-02	AA7	20	40
9	S-02	BB12	20	40
10	S-02	DD34	20	40
11	S-02	BB18	20	40
12	S-02	EE25	20	40
13	S-02	CC5	20	40
14	S-02	EE7	20	40
15	S-02	GG11	20	40
16	S-02	HH30	20	40
17	S-02	GG23	20	40
18	S-02	HH8	20	40
19	S-02	Z30	20	40
20	S-02	II39	20	40

RED FORCE

ID	TYPE	LOC	NUM LAUNCHERS	NUM SPARES
1	S-02	D14	20	40
2	S-02	Q7	20	40
3	S-02	P28	20	40
4	S-02	Q41	20	40
5	S-02	K37	20	40
6	S-02	K23	20	40
7	S-02	M17	20	40
8	S-02	K7	20	40
9	S-02	J12	20	40
10	S-02	H34	20	40
11	S-02	J18	20	40
12	S-02	G25	20	40

ID	TYPE	LOC	NUM LAUNCHERS	NUM SPARES
13	S-02	I5	20	40
14	S-02	G7	20	40
15	S-02	E11	20	40
16	S-02	D30	20	40
17	S-02	E23	20	40
18	S-02	D8	20	40
19	S-02	L30	20	40
20	S-02	C39	20	40

LAND-BASED TARGET STATUS

BLUE TARGETS

ID	TYPE	SAMs	LOC	DAMAGE	HARDNESS
0	Command Center	1	HH14	0	500
1	Airfield	2	U7	0	1000
2	Airfield	3	V28	0	1000
3	Airfield	4	U41	0	1000
4	Radar	10	DD34	0	300
5	Radar	9	BB12	0	300
6	City	0	II49	0	300
7	City	0	JJ26	0	300
8	City	0	GG5	0	300
9	POL Site	8	AA7	0	200
10	POL Site	6	AA23	0	200
11	POL Site	0	BB42	0	200
12	POL Site	16	HH30	0	200
13	Ammo Depot	5	AA37	0	600
14	Ammo Depot	7	Y17	0	600
15	Ammo Depot	15	GG11	0	600
16	Ammo Depot	20	II39	0	600
17	Rail Center	0	Y49	0	200
18	Rail Center	0	JJ46	0	200
19	Rail Center	12	EE25	0	200
20	Rail Center	14	EE7	0	200
21	Dam	0	X10	0	200
22	Dam	0	U33	0	200
23	Dam	0	CC29	0	200
24	Dam	0	HH20	0	200
25	Staging Area	0	EE45	0	200
26	Staging Area	19	Z30	0	200
27	Staging Area	0	EE17	0	200
28	Staging Area	13	CC5	0	200
29	Power Plant	0	GG47	0	300
30	Power Plant	17	GG23	0	300

ID	TYPE	LOC	NUM LAUNCHERS	NUM SPARES	
31	Power Plant	18	HH8	0	300
32	Industrial Area	0	JJ4	0	400
33	Industrial Area	11	BB18	0	400
34	Industrial Area	0	HH36	0	400
35	Bridge	0	V34	0	100
36	Bridge	0	T48	0	100
37	Bridge	0	EE49	0	100
38	Bridge	0	DD30	0	100
39	Bridge	0	Z10	0	100
40	Bridge	0	JJ10	0	100

RED TARGETS

ID	TYPE	SAMs	LOC	DAMAGE	HARDNESS
0	Command Center	1	D14	0	500
1	Airfield	2	Q7	0	1000
2	Airfield	3	P28	0	1000
3	Airfield	4	Q41	0	1000
4	Radar	10	H34	0	300
5	Radar	9	J12	0	300
6	City	0	C49	0	300
7	City	0	B26	0	300
8	City	0	E5	0	300
9	POL Site	8	K7	0	200
10	POL Site	6	K23	0	200
11	POL Site	0	J42	0	200
12	POL Site	16	D30	0	200
13	Ammo Depot	5	K37	0	600
14	Ammo Depot	7	M17	0	600
15	Ammo Depot	15	E11	0	600
16	Ammo Depot	20	C39	0	600
17	Rail Center	0	M49	0	200
18	Rail Center	0	B46	0	200
19	Rail Center	12	G25	0	200
20	Rail Center	14	G7	0	200
21	Dam	0	N10	0	200
22	Dam	0	Q33	0	200
23	Dam	0	I29	0	200
24	Dam	0	D20	0	200
25	Staging Area	0	G45	0	200
26	Staging Area	19	L30	0	200
27	Staging Area	0	G17	0	200
28	Staging Area	13	I5	0	200
29	Power Plant	0	E47	0	300
30	Power Plant	17	E23	0	300
31	Power Plant	18	D8	0	300
32	Industrial Area	0	B4	0	400
33	Industrial Area	11	J18	0	400

ID	TYPE	SAMs	LOC	DAMAGE	HARDNESS
34	Industrial Area	0	D36	0	400
35	Bridge	0	P34	0	100
36	Bridge	0	R48	0	100
37	Bridge	0	G49	0	100
38	Bridge	0	H30	0	100
39	Bridge	0	L10	0	100
40	Bridge	0	B10	0	100

Scenario #10

AIRCRAFT SQUADRON RESOURCES

BLUE FORCE

SQD	OWNER	TYPE	NUM	LOCATION
1	1	F-01	24	U7
2	1	F-01	24	U7
3	2	F-01	24	V28
4	2	F-01	24	V28
5	3	F-01	24	U41
6	3	F-01	24	U41
7	1	F-03	20	U7
8	1	F-03	20	U7
9	2	F-03	20	V28
10	2	F-03	20	V28
11	3	F-03	20	U41
12	3	F-03	20	U41
13	1	F-20	20	U7
14	1	F-20	20	U7
15	2	F-20	20	V28
16	2	F-20	20	V28
17	3	F-20	20	U41
18	3	F-20	20	U41
19	1	B-50	16	U7
20	1	B-50	16	U7
21	2	B-50	16	V28
22	2	B-50	16	V28
23	3	B-50	16	U41
24	3	B-50	16	U41
25	1	W-11	12	U7
26	2	W-11	12	V28
27	3	W-11	12	U41
28	1	K-12	4	U7
29	2	K-12	4	V28
30	3	K-12	4	U41

RED FORCE

SQD	OWNER	TYPE	NUM	LOCATION
1	1	F-01	24	Q7
2	1	F-01	24	Q7
3	2	F-01	24	P28
4	2	F-01	24	P28
5	3	F-01	24	Q41
6	3	F-01	24	Q41
7	1	F-03	20	Q7
8	1	F-03	20	Q7
9	2	F-03	20	P28
10	2	F-03	20	P28
11	3	F-03	20	Q41
12	3	F-03	20	Q41
13	1	F-20	20	Q7
14	1	F-20	20	Q7
15	2	F-20	20	P28
16	2	F-20	20	P28
17	3	F-20	20	Q41
18	3	F-20	20	Q41
19	1	B-50	16	Q7
20	1	B-50	16	Q7
21	2	B-50	16	P28
22	2	B-50	16	P28
23	3	B-50	16	Q41
24	3	B-50	16	Q41
25	1	W-11	12	Q7
26	2	W-11	12	P28
27	3	W-11	12	Q41
28	1	K-12	4	Q7
29	2	K-12	4	P28
30	3	K-12	4	Q41

STATUS OF SAMS

BLUE FORCE

ID	TYPE	LOC	NUM LAUNCHERS	NUM SPARES
1	S-02	HH14	20	40
2	S-02	U7	20	40
3	S-02	V28	20	40
4	S-02	U41	20	40
5	S-02	AA37	20	40
6	S-02	AA23	20	40
7	S-02	Y17	20	40
8	S-02	AA7	20	40

ID	TYPE	LOC	NUM LAUNCHERS	NUM SPARES
9	S-02	BB12	20	40
10	S-02	DD34	20	40
11	S-02	BB18	20	40
12	S-02	EE25	20	40
13	S-02	CC5	20	40
14	S-02	EE7	20	40
15	S-02	GG11	20	40
16	S-02	HH30	20	40
17	S-02	GG23	20	40
18	S-02	HH8	20	40
19	S-02	Z30	20	40
20	S-02	II39	20	40

RED FORCE

ID	TYPE	LOC	NUM LAUNCHERS	NUM SPARES
1	S-02	D14	20	40
2	S-02	Q7	20	40
3	S-02	P28	20	40
4	S-02	Q41	20	40
5	S-02	K37	20	40
6	S-02	K23	20	40
7	S-02	M17	20	40
8	S-02	K7	20	40
9	S-02	J12	20	40
10	S-02	H34	20	40
11	S-02	J18	20	40
12	S-02	G25	20	40
13	S-02	I5	20	40
14	S-02	G7	20	40
15	S-02	E11	20	40
16	S-02	D30	20	40
17	S-02	E23	20	40
18	S-02	D8	20	40
19	S-02	L30	20	40
20	S-02	C39	20	40

LAND-BASED TARGET STATUS

BLUE TARGETS

ID	TYPE	SAMs	LOC	DAMAGE	HARDNESS
0	Command Center	1	HH14	0	500
1	Airfield	2	U7	0	1000

ID	TYPE	SAMs	LOC	DAMAGE	HARDNESS
2	Airfield	3	V28	0	1000
3	Airfield	4	U41	0	1000
4	Radar	10	DD34	0	300
5	Radar	9	BB12	0	300
6	City	0	II49	0	300
7	City	0	JJ26	0	300
8	City	0	GG5	0	300
9	POL Site	8	AA7	0	200
10	POL Site	6	AA23	0	200
11	POL Site	0	BB42	0	200
12	POL Site	16	HH30	0	200
13	Ammo Depot	5	AA37	0	600
14	Ammo Depot	7	Y17	0	600
15	Ammo Depot	15	GG11	0	600
16	Ammo Depot	20	II39	0	600
17	Rail Center	0	Y49	0	200
18	Rail Center	0	JJ46	0	200
19	Rail Center	12	EE25	0	200
20	Rail Center	14	EE7	0	200
21	Dam	0	X10	0	200
22	Dam	0	U33	0	200
23	Dam	0	CC29	0	200
24	Dam	0	HH20	0	200
25	Staging Area	0	EE45	0	200
26	Staging Area	19	Z30	0	200
27	Staging Area	0	EE17	0	200
28	Staging Area	13	CC5	0	200
29	Power Plant	0	GG47	0	300
30	Power Plant	17	GG23	0	300
31	Power Plant	18	HH8	0	300
32	Industrial Area	0	JJ4	0	400
33	Industrial Area	11	BB18	0	400
34	Industrial Area	0	HH36	0	400
35	Bridge	0	V34	0	100
36	Bridge	0	T48	0	100
37	Bridge	0	EE49	0	100
38	Bridge	0	DD30	0	100
39	Bridge	0	Z10	0	100
40	Bridge	0	JJ10	0	100

RED TARGETS

ID	TYPE	SAMs	LOC	DAMAGE	HARDNESS
0	Command Center	1	D14	0	500
1	Airfield	2	Q7	0	1000
2	Airfield	3	P28	0	1000
3	Airfield	4	Q41	0	1000
4	Radar	10	H34	0	300

ID	TYPE	SAMs	LOC	DAMAGE	HARDNESS
5	Radar	9	J12	0	300
6	City	0	C49	0	300
7	City	0	B26	0	300
8	City	0	E5	0	300
9	POL Site	8	K7	0	200
10	POL Site	6	K23	0	200
11	POL Site	0	J42	0	200
12	POL Site	16	D30	0	200
13	Ammo Depot	5	K37	0	600
14	Ammo Depot	7	M17	0	600
15	Ammo Depot	15	E11	0	600
16	Ammo Depot	20	C39	0	600
17	Rail Center	0	M49	0	200
18	Rail Center	0	B46	0	200
19	Rail Center	12	G25	0	200
20	Rail Center	14	G7	0	200
21	Dam	0	N10	0	200
22	Dam	0	Q33	0	200
23	Dam	0	I29	0	200
24	Dam	0	D20	0	200
25	Staging Area	0	G45	0	200
26	Staging Area	19	L30	0	200
27	Staging Area	0	G17	0	200
28	Staging Area	13	I5	0	200
29	Power Plant	0	E47	0	300
30	Power Plant	17	E23	0	300
31	Power Plant	18	D8	0	300
32	Industrial Area	0	B4	0	400
33	Industrial Area	11	J18	0	400
34	Industrial Area	0	D36	0	400
35	Bridge	0	P34	0	100
36	Bridge	0	R48	0	100
37	Bridge	0	G49	0	100
38	Bridge	0	H30	0	100
39	Bridge	0	L10	0	100
40	Bridge	0	B10	0	100

APPENDIX E

LIST OF TASKS FOR THE BLUE FORCE

Task ID	Importance	Description
Scenario #1		
1	10	Attack Red airfield at P12
2	8	Attack Red airfield at Q27
3	10	Defend Blue airfield at V12
4	10	Defend Blue airfield at V42
5	5	Defend Blue bridge at V8
6	8	Defend Blue airfield at W27
7	9	Attack Red POL site at L28
8	10	Attack Red radar at H36
9	7	Attack Red ammo depot at M39
10	7	Defend Blue ammo depot at Z18
11	6	Attack Red ammo depot at M23

Scenario #2

Task ID	Importance	Description
12	8	Defend Blue airfield at V12
13	9	Defend Blue airfield at W27

Task ID	Importance	Description
14	7	Attack Red POL site at K9
15	10	Attack Red HQ at G25
16	9	Attack Red radar at H36
17	5	Defend Blue bridge at V8
18	8	Defend Blue radar at DD36
19	10	Attack Red radar at H14
20	7	Attack Red airfield at O27
21	8	Attack Red airfield at P12
22	6	Defend Blue dam at U7
23	10	Attack Red ammo depot at L18
24	8	Attack Red POL site at L28
25	5	Attack Red munition depot at J46
	Scenario #3	
26	9	Attack Red airfield at P12
27	10	Attack Red airfield at O27
28	8	Attack Red airfield at P42

Task ID	Importance	Description
29	7	Attack Red industrial area at K3
30	5	Attack Red ammo depot at M39
31	9	Defend Blue radar at DD14
32	9	Defend Blue radar at DD36
33	10	Defend Blue HQ at EE25
34	4	Attack Red POL site at J46
35	5	Attack Red rail center at J40
		Scenario #4
36	9	Defend Blue airfield at V6
37	7	Defend Blue POL site at AA17
38	9	Attack Red airfield at P6
39	8	Attack Red airfield at O19
40	10	Defend Blue HQ at EE31
41	10	Attack Red radar at K11
42	10	Attack Red airfield at O39
43	7	Attack Red POL site at K43

Task ID	Importance	Description
44	7	Attack Red radar at J32
45	7	Attack Red POL site at K17

Scenario #5

Task ID	Importance	Description
46	6	Defend Blue dam at T12
47	10	Defend Blue bridge at U13
48	10	Attack Red airfield at O39
49	8	Defend Blue airfield at W19
50	9	Attack Red radar at J32
51	10	Attack Red airfield at O39
52	9	Attack Red ammo depot at L26
53	10	Attack Red airfield at P6
54	7	Attack Red staging area at M45
55	5	Attack Red bridge at N44
56	10	Defend Blue aifield at V6

Scenario #6

Task ID	Importance	Description
56	5	Defend Blue airfield at V10
57	8	Defend Blue airfield at U47
58	8	Attack Red radar at K13
59	9	Attack Red radar at K47
60	9	Defend Blue radar at AA13
61	10	Defend Blue radar at AA47
62	10	Attack Red HQ at E17
63	7	Attack Red airfield at N28
64	6	Defend Blue dam at V32
65	5	Attack Red dam at P18
66	6	Attack Red dam at P32

Scenario #7

Task ID	Importance	Description
67	10	Defend Blue airfield at U43
68	9	Defend Blue airfield at U23
69	10	Attack Red radar at K15

Task ID	Importance	Description
70	9	Attack Red POL site at L38
71	8	Attack Red POL site at L8
72	6	Defend Blue POL site at Z38
73	6	Attack Red ammo depot at M15
74	7	Attack Red radar at L30
75	8	Defend Blue radar at Z30

Scenario #8

Task ID	Importance	Description
76	10	Attack Red airfield at P10
77	9	Attack Red ammo depot at N34
78	10	Defend Blue radar at AA15
79	7	Defend Blue radar at Z30
80	6	Defend Blue bridge at T16
81	9	Defend Blue airfield at U23
82	8	Defend Blue airfield at V10
83	6	Attack Red radar at L30
84	8	Attack Red radar at H44

Task ID	Importance	Description
85	7	Attack Red ammo depot at G39
		Scenario #9
86	10	Attack Red airfield at P28
87	9	Attack Red airfield at Q41
88	9	Defend Blue airfield at U41
89	8	Attack Red POL site at K7
90	4	Defend Blue dam at X10
91	8	Defend Blue staging area at Z30
92	8	Defend Blue POL site at AA7
93	8	Attack Red bridge at L10
94	7	Attack Red staging area at L30
95	9	Attack Red airfield at Q7
96	10	Defend Blue airfield at V28
		Scenario #10
97	10	Attack Red airfield at P28
98	9	Attack Red airfield at Q41

Task ID	Importance	Description
99	8	Defend Blue airfield at U41
100	7	Defend Blue airfield at V28
101	9	Attack Red ammo depot at M17
102	9	Attack Red rail center at M49
103	8	Attack Red dam at N10
104	6	Attack Red dam at Q33
105	7	Defend Blue dam at X10
106	5	Attack Red bridge at L10
107	9	Defend Blue airfield at U7

BIOGRAPHICAL SKETCH

Gregory Dean Elder was born in Munich, Germany, on December 7, 1956. He spent most of his youth in El Paso, Texas, where he attended Crosby Elementary School and Andress High School. He received a Congressional Appointment to the United States Air Force Academy in 1975. He graduated from the Academy in May 1979 with a Bachelor of Science degree in computer science and a commission as a Second Lieutenant in the United States Air Force. From 1979-1983, he was assigned to Gunter Air Force Station where he worked at the Air Force Data Systems Design Center and the Air Force Small Computer/Office Automation Service Organization. In 1983 he was re-assigned to Wright-Patterson Air Force Base where he served as a Deputy Chief in the Information Systems and Technology Center of the Aeronautical Systems Division. In 1984 he began graduate study at Wright State University in Dayton, Ohio. He received a Master of Science degree in computer science in 1986. In 1987, he served as an Assistant Professor of Aerospace Studies at the University of Nebraska-Lincoln. From 1988-1990, he served on the faculty of the United States Air Force Academy teaching and conducting research in the Military Studies Department. He entered the Graduate College of Arizona State University in August 1990 to begin study towards a degree of Doctor of Philosophy in computer science. He currently holds the rank of Major in the Air Force and has completed the following Professional Military Education courses: Squadron Officer School, Air Command and Staff College, and the Marine Corps Command and Staff Course. He is a member of the Association for Computing Machinery, the IEEE Computer Society, the American Association for Artificial Intelligence, the Air Force Association, the National Association of Rocketry, and the Upsilon Pi Epsilon Honor Society.