AFRL-IF-RS-TR-2006-282
In-House Final Technical Report
September 2006

MODELING OF COMPLEX ADAPTIVE SYSTEMS IN AIR OPERATIONS

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

STINFO FINAL REPORT

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the Air Force Research Laboratory Rome Research Site Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-IF-RS-TR-2006-282 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/

JULIE BRICHACEK
Chief, Information Systems Research Branch

/s/

JAMES W. CUSACK
Chief, Information Systems Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
SEP 2006	Final	Nov 01 – Dec 05

4. TITLE AND SUBTITLE

MODELING OF COMPLEX ADAPTIVE SYSTEMS IN AIR OPERATIONS

5a. CONTRACT NUMBER
In-House

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
N/A

6. AUTHOR(S)

Timothy E. Busch and Dawn A. Trevisani

5d. PROJECT NUMBER
459S

5e. TASK NUMBER
IH

5f. WORK UNIT NUMBER
11

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

AFRL/IFSB
525 Brooks Rd
Rome NY 13441-4505

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/IFSB
525 Brooks Rd
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2006-282

12. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PA# 06-614

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This report introduces the concept of using simulation for both plan tracking and state estimation and prediction. Model predictive control theory provides the basis for this investigation. Given some set of objectives the military commander must devise a sequence of actions that transform the current state to the desired one. The desire to do this in faster than real-time so that many courses of action can be considered motivates us to investigate modeling techniques that explicitly produce such courses of action. The class of problem can be modeled as a Markov decision process (MDP) whose principal solution is stochastic dynamic programming. The report presents a historical context for the application of control theory to the command and control problem space and introduces a mechanism for dealing with the resulting computational complexity.

15. SUBJECT TERMS
Command and Control, POMDP, Markov Decision Processes, Decision making under uncertainty, Model Predictive Control

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Timothy Busch
U	U	U	UL	43	19b. TELEPONE NUMBER (*Include area code*)

Table of Contents

List of Figures

1.0 Summary:

We begin with a brief history of the "science" of Command and Control (C2) with the objective of highlighting the contributions of the controls community to this area. Much of the recent work in the area of C2 has been from the Artificial Intelligence (AI) and Military Operations Research (MORS) and some from these communities have denigrated the ability of control theory to have anything meaningful to say. Furthermore, the focus of C2 research has been on the command and control system itself independent of the battlespace.

The objective of this work was to develop computational models of air operations that explicitly support course of action development in near real time and then to apply them in a control theoretic approach for dynamic management of the battlespace. The intention of the work was to extend a Markov Decision Process (MDP) framework that was developed by Alphatech Inc., Boston University, DRAPER labs, and MIT that was sponsored under the Defense Advanced Research Projects Agency (DARPA) Joint Force Air Component Commander (JFACC) program to incorporate partial observability. The Partially Observed Markov Decision Process (POMDP) provides a rich framework for the investigation of decision making in Command and Control systems. DARPA formulated a model of the Joint Air Operation (JAO) environment and developed a control theoretic approach for the formulation of mission packages and assignment of packages to targets. The JAO environment is an uncertain dynamical system where decisions are made over time and the transition from state to state is stochastic and dependent on those decisions. This class of problems can be formulated as a Markov decision process whose well-known solution is provided by Stochastic Dynamic Programming (SDP). It is also well known that this approach suffers from the "curse of

dimensionality" and is intractable for realistically sized JAO problems. Significant work has been done in "approximate dynamic programming" that provides near optimal solutions to MDP problems. The current formulation however assumes perfect state information with a probabilistic transition to a new state for a given action. Unfortunately, in real combat systems the states are not necessarily known but are often inferred from observations. (This is certainly true of the adversary's states). Partially Observed Markov Decision Processes model sequential decision-making when the outcomes are uncertain and the state of the system cannot be completely observed and are suited to the JAO environment. Extending the JAO MDP to include partial observability, will undoubtedly lead to further computational complexity. The initial thought for dealing with this complexity in addition to the approximate dynamic programming techniques already investigated is through a model predictive control approach using a simulation rollout strategy. This approach has been shown to have success in a job shop scheduling application[22]. Initially, we will start by addressing issues such as real time target-weapon pairing, air-space deconfliction, or ISR rescheduling that might result from the pursuit of a time critical target.

2.0 Background:

It is appropriate to introduce a bit of history regarding the application of control theory to the problems to be addressed. Control theoretic approaches have been criticized recently as being non-productive and intractable and that applications of other approaches are better suited to the problem space. Still, the establishment of a "science" of command and control owes a lot to the control theoretic community.

There exists a substantial body of research in the science of command and control developed over the past 30 years. In 1976 the Air Force Office of Scientific Research held a

meeting with the objective of providing the managers of the Service Offices of Research with the knowledge that would allow them to formulate research programs relevant to military problems in decision information. Included in this objective was the presentation of scientific disciplines that had not previously been applied to such military problems[1].

A watershed event in research on C2 was a series of workshops, beginning in 1978, jointly sponsored by the Office of Naval Research and the Electronics Systems Laboratory (now known as the Laboratory for Information and Decision Systems (LIDS)) at MIT. The two names most prominently associated with these workshops are Michael Athans and Alex Levis, though many researchers participated over the 10 or so years during which the workshops were held. Both Athans and Levis are control theorists, and this perspective did much to shape the emphasis and direction of the work. In 1979, the National Defense University sponsored a workshop on the "Quantitative Assessment of Utility of Command and Control Systems" with the goal of developing a shared conceptual framework, language, and a coordinated program research and testing. [2] Levis stated that this workshop: "established the starting place and time" on the quest for a C3 theory.[3]

Robert Herman was commissioned in 1981 to survey the field of C3. Motivated by the results of this survey, the Joint Director of Laboratories established a technology panel to: develop a coherent theory of C3; to develop a process for joint planning; and to form affinity groups. These groups would be in the areas of decision aids, radios and links, distributed processing, data fusion, networks (with a sub-group on network and simulation support), and the basic research group.[4] This report and the affinity groups were instrumental in defining the research that took place in the ensuing 30 years.

The various meetings and workshops were responding to an accelerating need for coordination and control of C3 in an increasingly complex military environment. Control theory is a multidisciplinary science associated with dynamic systems and, while not explicitly a part of the C3 affinity groups, the controls community has contributed to the science of C3.

Preliminary to the understanding of dynamic systems is the formulation of models that capture the system's essential behavior. The model we formulate then will underpin the research that is subsequently conducted and the results obtained. To say it another way, the answers we get will depend on the questions we ask. If we look at the papers published from 1978 through 1993, we will see a point of view that focuses primarily on the C2 organization as the object of modeling and study. Additionally, it is clear from the literature that the command and control problem is rarely formulated in a control theoretic context. The work of Dockery and Woodcock highlights this through a series of papers that they refer to as the model series.[5,6,7] Their goal has been to embed a theory of C2 into a theory of combat. It was their perception that the existing theories of combat were inadequate. "For this reason we have been forced to develop an adequate theoretical understanding of some of the common processes and principles of combat *before considering the role of C2* in these processes."(Italics added).[8] They form a taxonomy of models within a Catastrophe Theory framework that include static, time dependent, time and space dependent, structural, and perceptual models. These models are suitable for use in controller design but again the problem formulation is different than one that would be familiar to the controls community. Hopple[9] also suggests a taxonomy of methods and models that are available to the decision systems engineer. There is no mention of control methods as a part of the taxonomy unless they were included in the other category. Andriole and Halpin[10], and the C2

community in general, have emphasized the information management and artificial intelligence branches of that taxonomy tree.

A great deal of emphasis has been given to modeling the decision process within the C2 system. Wohl provides words to this persistent emphasis:

> "The current lack of understanding as concluded by OSD colloquiums simply mirrors the very same lack of understanding of the underlying human decision processes and how to improve them. Attempts to apply control theory, state-variable theory, or fuzzy set theory, while they may be analytically interesting and even productive in terms of setting performance bounds on the command and control process, cannot help us to understand the creative act of a commander who rejects a staff-recommended course of action, reviews the situation, gathers the staff together, and says, "All right, here's what we're going to do…" Without such an understanding, it will be difficult to make headway in developing tactical decision aids or in establishing and verifying associated communication needs. Thus a theory of command and control must start with a theory of decision making for command and control."[11]

Models of the decision maker, [12, 13] decision-making process ,[14] models of career progression,[15] and with models of the C2 structure [16,17] all point to the C2 system as the primary object of research. It is only within the last few years that one can see a broadening of the problems addressed by the controls community (as represented in the literature) that might include military command and control.

Models and some understanding of the dynamics of the plant quite naturally lead to the invocation of controllers that are used to improve the systems performance. This is exemplified by the Headquarters Effectiveness Assessment Tool (HEAT), which was developed to provide an objective method for the measurement of military headquarters. The theoretical approach combines military and systems theory to identify the properties of effective planning. In particular, a control theoretic analogy is used in deriving the HEAT decision-making paradigm. Within the analogy they cast effective planning as an optimal control problem using a closed

loop model. Their use of state observation and estimation in conjunction with controllers and actuators is more closely associated with a closed loop control system.

"We do not presume that command and control is simply an optimal control problem. We do assert that a rational C2 process will make the same provisions for coping with uncertainty found in optimal estimators. A rational C2 process will also make the same provisions for anticipating contingencies found in optimal control strategies." [18]

The HEAT is a control theoretic metaphor that was used by trained observers in exercises and experiments between 1983 and 1990. The results of these applications were a set of specific guidelines for the design of effective headquarters.

We contend that, with only a few exceptions, the primary research emphasis towards a theory of command and control has been on how to *structure* the C2 organization to accomplish its mission. Additionally, in those cases where the entire battlespace is considered, the problem formulation is different. Here, the set of questions of interest include: How best to decompose and distribute the decision-making tasks? What are the key interfaces, and the data flows across them? What types of delay and errors do human operators introduce, and how can these be modeled and minimized? How can a C2 organization be structured so as to be robust and survivable? The point is that the object of study here again is the C2 organization and its components.

By contrast, the point of view taken here is, first and foremost, the battle space itself. What are the right models to use for the state and dynamics of a military engagement? What are the control signals (commands) available to modify this system's behavior? What impact does noise and latency in the observation process (ISR) have on solution quality? How can one best model the adversary, who is also issuing control signals into the same plant? In other words, as

opposed to focusing on the C2 organization as an object of study in its own right, the approach taken here is "bottoms up," starting first with the objects to be controlled (the uncertain battle space, including an active intelligent adversary), and then deriving an appropriate control theoretic solution based on the mathematical structure of the modeling approach that has been adopted.

Both of these sets of questions – the C2 organization, on the one hand, and control theoretic modeling of battle space state and dynamics, on the other hand – are useful and important; both will undoubtedly play a role in future developments. The point here is that the heavy emphasis on treating the battlespace itself *as a plant to be controlled*, in the classical control theoretic sense of this term, is new, and is what distinguishes this work from much of what as gone before. It is *not* the case that this work is a rehash of questions that were answered 30 years ago; and to see why that is so, we claim, requires conscious attention to this shift of emphasis.[19]

3.0 Introduction:

The planning and execution of military operations is a complex process. The current process of Air Tasking Order (ATO) planning takes 72 hours from inception through completion with multiple cycles occurring simultaneously. This means that means that targets are being selected some 48 hours prior to their execution. The execution phase commences with the dissemination of the plan, preparation of resources, and implementation of the orders. In a perfect world, during the planning phase we would have considered all the pertinent information and formulated all necessary contingency plans. In such a world, once we provide the scripts to the actors there is nothing further required of the execution sub-system. Unfortunately, there is uncertainty in the information we use to formulate the plan, random events take place in the battle environment and there is an active intelligent adversary who has his own objectives. We

can think of the plan as a trajectory through the battle space that transitions from one set of states to a more desired set. Execution then is concerned with reducing the effects of perturbations to the trajectory and achieving the planned objectives. When considered this way, each decision necessary to address deviations has to insure that it won't result in larger perturbations or even catastrophic failure. Timing and synchronization must be maintained and anticipation of adversarial response must be considered.

The Combat Operations section within the Air Operations Center has the responsibility of orchestrating the current execution of the Air Tasking Order to achieve the commander's objectives. Current decision aids and processes don't deal well with real-time changes to the ATO. Consider the example of reassigning a weapon to a new target, in real-time, as one might do for a time critical target. Given the rules of engagement and the requirements for avoiding collateral damage, an appropriate weapon needs to be selected for the target. This target-weapon paring occurs over the resources that are available to strike the target in a timely manner. These resources might include strikers, SEAD, refuelers, etc. Once the assets are chosen a detailed route plan is needed to enable the crew to find and engage the target while insuring that there are no conflicts in the airspace caused by the change. Appropriate information collection resources need to be dispatched to provide the bomb damage assessment.

The complexity and tempo of future operations coupled with the desire to move fewer forces forward will require increasing amounts of automation. Air tasking, air coordination, and collection management individually constitute complicated systems and together they form a complex system. Real world experiments in warfare are difficult to carryout, which leads us to the use of simulations as a means of exploring its nature and some hint as to why reduction to its component parts is not enough to explain or control it.

Once developed, the simulations can then be used in a predictive mode to generate possible courses of action. Model predictive control is an optimal control method that uses a model of the dynamical system to predict the effect of a command (or sequence of commands) to the actual system. The optimal control problem is solved at each step and the optimal command sequence is implemented until another data sample is taken. The updated information is used solve a new optimal control problem and the process is repeated. The benefit of this "rolling horizon" is that we need not have perfect model of the system. It only has to be accurate over the selected horizon.

Here we introduce the concept of using simulation for both plan tracking and prediction. Deviations from the plan can then be detected through a comparison of the two and new courses of action can be developed to achieve the objective.

4.0 Model Predictive Control

We have chosen to confine our study to the "execution" phase of an operation. This assumes that such issues as determining the Commanders Intent and how it has been translated into specific objectives has already been accomplished. Resources have been apportioned and risk analysis for the overall operation has been performed. All that remains for the operator at this point is to keep the plan on track. Consider the case where feedback from the battlespace gives us an indication of an unanticipated threat. What action should we take? What are the ramifications of that action? To answer these questions we need to be able to predict the future. Our crystal ball then is derived from models of individual components which are allowed to interact in a simulation.

Models can be used to predict the consequences of various decisions that can be made at each step of command center processes associated with disaster relief, military operations,

network operations, etc. The sequence of these decisions constitutes a course of action (COA). In Model Predictive Control theory, the first decision in the COA would be implemented and the decision problem would be solved again based on the new state of the world. Model Predictive Control is attractive because the theory is well established and has been shown to be effective in the control of complex systems whose dynamics are not fully understood.

4.1 Partially Observed Markov Decision Process (POMDP)

Let's begin with a description of a Markov Decision Process (MDP)[20] that might form the computational mechanism for developing courses of action which are called policies in the literature. Formally, a MDP can be described as a tuple < S,A,T,R> where:

S is a finite set of states of the world. In this case, they might be such things as aircraft position, velocity, health, weapon status, target status etc.

A is a finite set of actions.

T: S x A→ Π(S) is the state transition function, giving for each world state and action a probability distribution over world states. We can write this as T(s,a,s') for the probability of reaching state s' given that we are in state s and take action a.

R: S x A → $\mathbb{R}$ is the reward function, giving the expected immediate reward gained by taking each action in each state. We can write this as R(s,a) for the expected reward for taking action a in state s.

We define a non-stationary policy as a sequence of rules used to select a particular action from a given state at a particular time. The policy π_k is used to choose the action on the k-th to last step as a function of the current state s_k.

Given a policy, we can define value $V_{\pi,k}(s)$ as the expected sum of rewards gained from starting in state s and executing policy π_k. So for k= 1 $V_{\pi,1}(s)=R(s,\pi_1(s))$; that is on the last step the value is just the expected reward for the action specified by the final element of the policy. In general then:

$V_{\pi,k}(s)=R(s,\pi_k(s))+\gamma\sum_{s\in S} T(s,\pi_k(s),s')V_{\pi,k-1}(s')$

where γ is a discount factor.

Our desire then is to maximize our value by determining the optimal policy π^*. The last step is easy: it should maximize the final reward. So

$\pi^*_{k-1}(s)= \text{argmax}_a R(s,a)$.

The optimal policy for the k-th step can be defined in terms of the optimal (k-1) step value function $V_{\pi^*k-1,k-1}$ or simply V^*_{k-1}.

$\pi^*_k(s)= \text{argmax}_a [R(s,\pi_k(s))+\gamma\sum_{s\in S} T(s,a,s')V^*_{k-1}(s')]$ where $V^*_{k-1}(s')$ is derived from π^*_{k-1} and V^*_{k-2}.

The principal approach for finding a solution to this type of problem is through Stochastic Dynamic Programming. Wohletz et al[21] point out that the SDP formulation produces policies that anticipate the effects of future contingencies and have demonstrated that the algorithm in fact will produce proactive rather than reactive behaviors when applied to air operations. However, it is well known that this approach suffers from the "curse of dimensionality" and is intractable for any realistically sized problem. Given the strengths of the formulation there has been a great deal of research in approximate dynamic programming methods in recent years. One approach in particular that is of interest, is the use of simulation to estimate[22] ("rollout") future rewards. This heuristic calls for the simulation to run faster than real-time over some

number of future states examining the results for the best rewards over that horizon. We use that information to inform the decision of which action to take for the current state and then repeat the process again. Critical to the success of this approach is in knowing what state we are in currently.

Partially Observed Markov Decision Processes are a generalization of MDPs in which it is not assumed the agent knows precisely the state s of the system, in each decision stage[23]. Continuing after [23]:

A POMDP can be described as a tuple < S, A, T, R, Ω, O>, where

S, A, T, and R describe the MDP

Ω is a finite set of observations that can be experienced in the world and

O:SxA → Π(Ω) is the observation function, which gives, for each action and resulting state, a probability distribution over possible observations. We write O(s', a, o) for the probability of making observation o given that action a was selected and state s' resulted. As before, we desire to select actions that will maximize the expected future reward.

We see that there are now two forms of uncertainty associated with the POMDP. One associated with the uncertainty of the current state and the other with the uncertainty of the resulting state for a particular action selected. To use the techniques developed for the solution of MDP, we rely on our belief of the current state.

Belief states can be represented as probability distributions over the states of the world. Again referring to [23]:

"A belief state is a probability distribution over S. We let b(s) denote the probability assigned to the world state s by belief state b. The state estimator must compute a new belief

state, b', given an old belief state b, an action a, and an observation 0. The new degree of belief in some state s', b'(s'), can be obtained from Bayes theory as follows:

$$b'(s') = \Pr(s' \mid o,a,b)$$

$$= \frac{\Pr(o \mid s',a,b)\ \Pr(s' \mid a,b)}{\Pr(o \mid a,b)}$$

$$= \frac{\Pr(o \mid s',a)\ \sum_{s \in S} \Pr(s' \mid a,b,s)\ \Pr(s \mid a,b)}{\Pr(o \mid a,b)}$$

$$= \frac{O(s,a,o)\ \sum_{s \in S} T(s,a,s')\ b(s)}{\Pr(o \mid a,b)}$$

The optimal policy then is given as before only now we use the current belief rather than the current state as our starting point. It should be highlighted that while the reward function remains the same, actions may be selected to improve our belief state (and hence the reward) as well as the desired state transition.

As in the case of the MDP, the partially observed case is intractable for large state spaces. To deal with this we propose to employ POMDP only when there are significant deviations from the planned air operations. This should result in a significant reduction in the complexity of the state space.

4.2 Implementation Concept

Earlier it was pointed out that the ATO planning cycle currently takes 72 hours. We could debate the efficiency of the process but let's agree that it produces a useful product in the ATO. The ATO then contains our plan for the next 24 hours and embodies the Joint Forces Air Component Commanders intentions, our anticipation of adversarial response, weather, etc. Unfortunately, the ATO itself consists of weapons, targets, rendezvous times, etc. but does not provide a complete view of how the states will evolve. In Figure 1 we introduce a plan simulation that takes as its input the ATO and the assumptions of enemy state and response, environment, etc. that went into planning the ATO. The simulation can then present or predict the evolution of the battle (according to the plan) within some confidence intervals. The problem we are interested in is in maintaining the spirit of the plan when things go wrong.

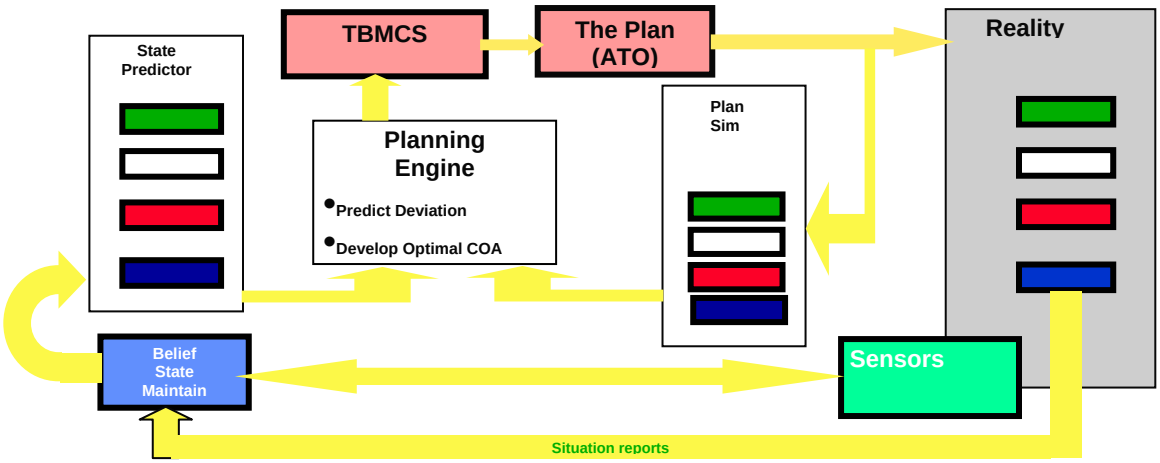

Figure 1 Proposed experimental diagram.

As much as we would like to fully instrument the battlefield we often don't have full access to the state space and even where we do reports, they are not necessarily timely or accurate. This is the problem of information fusion and falls into the realm of the intelligence analyst. There is much work already underway in belief state maintenance that might be applied directly here. Additionally, we are interested in predicting the future states. Since we don't have actual data that provides us with continuous information of what's happening on the battlefield we can use simulations as a means of providing the predicted evolution of the state space. There may be several of these running with different parameters or assumptions regarding the evolution of the operation. Certainly feedback from the battlespace can be used to update the simulations in motion to synch them up with the current belief state of the system. The results of this state predictor could then be compared with the plan and any deviations greater than the expected confidence intervals would alert an operator and initiate replaning.

Initial feasible solutions might be determined through the use of limited simulations from within the planning engine. These would then be used to seed the POMDP to determine possible courses of action to be presented to an operator for selection.

Consider the following vignette; a surveillance asset gets an indication of an unanticipated threat and a general location. The information is fed to the belief state maintenance where a threat id and location are generated with an appropriate confidence interval (error covariance). This information is then used to update the state predictor. The impact of the threat on the ongoing mission can then be assessed. If it has no impact on any executing mission (over its entire uncertainty) then it's of no consequence to the current operation. The information is retained for future reference. If it shows an impact then a decision has to be made as to what to do. Some options might include aborting any missions through the area, targeting further

collections against the suspected threat to reduce the uncertainty, reroute missions to avoid the threat, re-task other missions to pick up various pieces of the dropped mission etc. The fact that we have been keeping track of the ongoing operation gives us an excellent starting point for carrying out the replaning. Now rather than the static plan we started with we have a much better idea of where our resources are and what's available for retasking. Additionally, we can spawn abstracted simulations that can run in much faster than real time to provide an initial solution set. This feasible set becomes the basis over which we implement our optimal control policy.

4.3 Considerations for Simulation Selection

There are several practical issues in realizing this implementation foremost of which is the selection of the simulation. We rejected consideration of the development of a specific simulation for this effort since there were several efforts already underway to provide a comprehensive simulation the most notable of which is the Joint Simulation System. Since this system was under development we need a working simulation to start with.

The Command and Control Modeling and Simulation Branch at AFRL/IF worked cooperatively with the Air Force Office of Scientific Research (AFOSR) and the Defense Modeling and Simulation Office (DMSO) to develop distributed, virtual simulation testbeds for the evaluation of C2 decision aids, awareness aids, and advance visualization techniques. The Global Awareness Virtual Test Bed (GAVTB) and Real Time Decision Support System (RTDSS) were the result. In the process of developing these systems, we learned several valuable lessons that provided guiding principles for the selection of the simulations we would use. First, the simulation needed to be Government owned. Under the RTDSS and GAVTB programs, use of commercial simulations led to difficulties when changes to the code were

required. Often our requirements didn't represent a strong enough business case for the developer to make the needed changes. Occasionally, proprietary restrictions prevented us from examining the source code to understand why particular results occurred. Second, the simulation needed to work at the entity level. Entity level models (individual soldiers, aircraft, tanks, etc.) allow great flexibility in examining C2 issues. Additionally, we were interested in representation of all military components (air, land, and sea). Finally, we wanted a simulation that had a well established user community. These criteria led to the selection of the Joint Semi-Automated Forces (JSAF) simulation.

5.0 JSAF

JSAF (Joint Semi Automated Forces) is a collection of HLA-based simulation federates used to model military entities and their individual and small unit behaviors. JSAF was first used in the DARPA Synthetic Theater of War (STOW) Advanced Concept and Technology Demonstration (ACTD) to support combat and material development experiments in a Distributed Interactive Simulation (DIS) environment. Since its initial development JSAF has grown to better than 1100 libraries executing more than a million entities using 2 supercomputing centers and has been used in numerous exercises including MILLENIUM CHALLENGE and URBAN RESOLVE. The software is owned and maintained by Joint Forces Command and is considered Government Off The Shelf (GOTS).

JSAF is an open software architecture system that is written in C and comes bundled with several utility packages to help set the simulation environment and set initial SAF entity parameters. JSAF entities behave autonomously; moving through their environment, sensing targets, firing on threats, and communicating with other objects. JSAF objects interact with each other as well as other manned simulators using a network and HLA interface.

JSAF simulates entities by enabling them to execute a realistic range of basic actions inherent to the entity type. Their resources are accurately depleted as they move through their simulated environment. JSAF entities have the ability to detect and identify targets, fire on them and react to possible collisions in their path. These capabilities are based on provided entity rules such as range, motion, activity, direction, orders, and evaluation of threat. Communication among entities occurs at both the individual and unit levels.

JSAF relies on standard military doctrine to supply default values for task parameters. The user is then allowed to modify these parameters at any time to affect the overall behavior of the entity. The user can also specify the environment within which the entity will interact. Changes in the terrain, weather, and obstacles will cause the entities to behave differently.

6.0 SIMULATION ENVIRONMENT

In designing the federation, the goal for this environment was to not only provide the necessary data/information flows tailored for investigating advanced technologies for command and control, but also to provide an environment within which we could explore the science of simulation. A flexible environment was important as well in order to allow for models or simulations within the environment to be swapped in and out easily as the application requirements change. In addition, we wanted an environment that would provide us a leave-behind to be reused for other applications and exercises and not exist as a one time environment for a particular exercise; never to be used again. Thus, the approach to constructing this environment was a modular approach in order to enable re-configurability.

The components we selected to be included in the core federation are those that are essential in any C2 environment. The baseline can then be extended as requirements for specific applications call for additional data flows to be incorporated in order to test the technologies that

will be immersed within the environment. The entire federation resides at AFRL/Rome Research Site located in Rome, NY, within the Modeling and Simulation Facility. Other hosts could easily be connected remotely to form a distributed environment, but at this point all federates are located locally on the same subnet.

6.1 System Architecture

There are a total of 16 systems that make up our environment, with multiple systems running the JSAF federate. Figure 2 displays the architecture for the environment. The components and connections shown with dotted lines represent near future enhancements. The interconnection mechanism that was selected for the federation was the High Level Architecture (HLA) with gateways to facilitate data exchange with non-HLA members. Along with HLA, the RunTime Infrastructure (RTI-S) was selected to support operations and communication of federates.

JSAF is our primary simulation federate. JSAF accepts live weather data that is funneled through the Ocean, Atmosphere and Space Environmental Services (OASES) simulation. DTSim (Dynamic Terrain Simulation) introduces dynamic terrain features as it responds to detonations, weather and other effects within the battlespace that alter the terrain. UAVSim connects to JSAF as well, offering up Unmanned Aerial Vehicles (UAVs) and UAV objects to the battlespace as well as Moving Target Indicator (MTI) detections (in the future). Clutter provides a bit of confusion or noise to the battlespace, while SNN (Simulation Network News) provides text reports of the current situation. The JSB-RD Viewer application serves as our window into the battlespace. As the mission executes, the federates communicate and interact via HLA object classes and interaction classes. Object classes are used for periodically updated persistent data such as player location. Interaction classes are used to contain information for

asynchronous events, such as weapons fire. These federates previously mentioned constitute the simulation federation or stimulator which, through the use of a gateway, will touch operational systems within the Theater Battle Management Core Systems (TBMCS) pulling situational awareness (SA) data and ATO missions, as well as track reports, target reports and mission state data. Each of these federates, along with the flow of data, are described in the following sections.

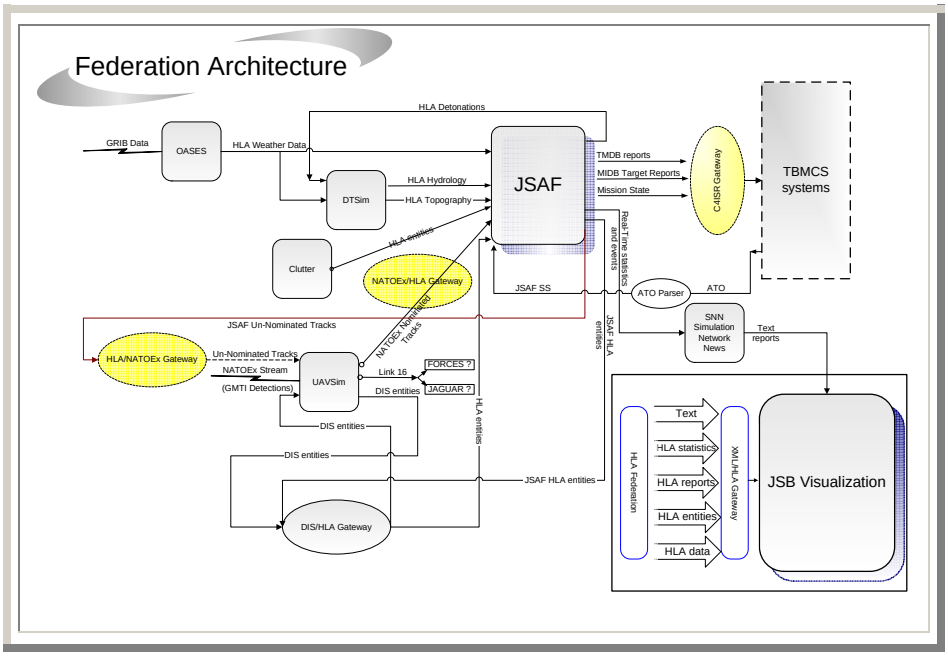

Figure 2. Federation Architecture Communication and Data Exchange

6.2 High Level Architecture (HLA)

As stated previously, the communication mechanism selected for the environment was the High Level Architecture (HLA). HLA is a general purpose software architecture used for the development and execution of distributed simulation applications to facilitate interoperability

and simulation reuse. The HLA was developed under the leadership of the Defense Modeling and Simulation Office (DMSO), but has since transitioned from a GOTS product to what is now a commercial product. HLA object models are used to describe an individual federation member (federate), creating an HLA simulation object model (SOM); or to describe a named set of multiple interacting federates (federation), creating a federation object model (FOM). FOMs and SOMs are characterized in terms of their objects, attributes, interactions, and parameters so as to provide a common frame of reference for describing object models in the HLA community. In either case, the primary objective is to facilitate interoperability among simulations and reuse of simulation components. The FOM we selected for our environment is a modified version of the MC02 FOM (The FOM used in Millennium Challenge 02). Each of the federates was modified to accept this FOM.

6.3 Runtime Infrastructure (RTI)

HLA is an architecture, not software. Therefore, the runtime infrastructure (RTI) software is required to support operations of a federation execution. The RTI software provides a set of services used by federates to coordinate their operations and communicate all simulation information between federates during a runtime execution. These services include the management of the federation, objects, time, declarations, ownership and data distribution. The version of RTI selected for this environment was RTI-S. This version of RTI does not provide for Time or Ownership Management services. It does, however, focus heavily on Data Distribution Management services, low latency transfers, and high throughput as well as all other service areas (Federation, Object, and Declaration). Time management is a feature that was not necessary for our environment. In that each federate executes in real-time, timing is handled by wall-clock time or system time. Because of this, startup of the environment does not require any

special ordering; federates can be initiated in any order. We incorporated the Network Time Protocol (NTP) to synchronize the clocks on each of the systems within the environment. This particular version of RTI requires that an instance of the RTI run on each federate within the simulation.

6.4 Gateways

6.4.1 DIS/HLA Gateway

Since we have Distributed Interactive Simulation (DIS) simulations included in the federation, a DIS/HLA gateway was implemented so that our DIS federates could speak to the rest of the federation. The DIS/HLA gateway is itself a federate that provides an interface to DIS simulation components and converts DIS formatted messages to HLA formatted messages and back. The DIS/HLA gateway was provided as part of the JSAF distribution and operates as a federate within the environment. The DIS/HLA gateway runs on a JSAF machine (described in section 5.3.1); a PC Athalon 3200+, 2.19 GHz system with 3 GB of memory running Linux RH 9.0 on a 100Mb connection.

6.4.2 NATOEX/XML Gateway

NATOEX streams are Moving Target Indicator (MTI) detections produced by the UAVSim simulation. These streams can not be consumed directly by JSAF. Therefore, a gateway is required to convert the NATOEX message streams into HLA formatted messages that can then be read by the other federates within the environment. This gateway is a very near future enhancement.

6.4.3 XML/HLA Gateway

One of our goals is to have an environment that is flexible enough to accept any type of application as a component within the federation; therefore, one cannot assume that all federates within the environment will be HLA compliant. Not only is HLA traffic difficult to parse and manipulate, there may exist components within the federation that will be neither HLA nor DIS based; such as our Visualization application. Given these shortcomings, we turned to another body of research that addresses the same issues of interchangeability; that being XML (eXtensive Markup Language). XML is quite easy to parse and is better understood by non-HLA and non-DIS applications. Therefore, an XML/HLA gateway, written in JAVA, was developed to convert HLA messages to XML formatted messages. The XML/HLA gateway acts as a federate within the simulation. It is a gateway that sits between the HLA federation and the non-HLA and non-DIS federates. In that the gateway is itself a federate, it subscribes to all HLA objects and interactions that are being published within the environment. It then publishes data/information to the models/simulations that have subscribed to the particular data. Currently, this is a one-way traffic flow from the HLA federation through the gateway to the non-HLA simulations/applications. Ultimately, the focus will be for the gateway to perform in a bi-directional mode so that as missions are re-tasked via the visualization application, the specific re-task commands will be sent via XML back to the gateway, where they will be converted back into HLA messages that can then be accepted by the other federates. Note: It is not yet clear how JSAF will accept re-route messages as input back into the simulation. This is left to the developers/integrators of our environment to ascertain which approach is best.

6.4.4 C4ISR Gateway

One of the motives driving this environment is to construct a simulation environment that is operationally focused toward C2 processes and decision support. The intent is to develop the simulation environment within the context of operational systems. In the past, technologies that were developed were tested within a simulation environment. These environments would be constructed toward the specific purpose of testing the technology. Only those systems (built for simulation purposes) that were absolutely necessary to test the technology were included. This would result in what we would consider an incomplete environment for C2 processes. Once the capability (or technology) was proven, the transition stage to integrate the new technology into an operational context was (and is) cost intensive. The ability to test enabling technologies within an operationally focused simulation environment dramatically reduces the cost of transitioning the technology. Therefore, our goal is to build a simulation environment that can pull/push data from/to operational systems.

This particular gateway will allow the federates within the simulation environment to communicate messages back and forth to/from real world C4ISR systems such as the Theater Battle Management Core Systems (TBMCS) by interfacing messages stemming from within the environment to a format understood by these systems. This interface/gateway has yet to be constructed. Approaches as to how this interface will be designed are currently being discussed. Consideration is being given to interfacing JSAF and/or XML messages that will translate back to HLA and be received by the other components within the environment.

6.5 Distributed Simulation Environment

6.5.1 Joint Semi-Automated Forces (JSAF)

The JSAF environment is an HLA compliant, semi-automated computer generated forces simulation. JSAF is used to simulate all the entities within the battlespace, with the exception of the Unmanned Aerial Vehicles (UAVs) which are controlled by UAVSim (described in a later section). JSAF generates entity level platforms, interactions, and behaviors. All interactions within the battlespace are resolved through JSAF at the entity level. Individual entities are task organized into units for a given mission. From within the simulation GUI, JSAF objects can then be controlled as units or individual entities.

The GUI is the main source of control over the entities in JSAF. Through the GUI, the operator/user defines the missions for each entity as well as weapon load, fuel level, etc.; all controlled as individual entities or units. This tasking operation is performed for both the blue side and the red side. Once all the tasking is complete, the scenario is saved as a file which can then be used repeatedly. Screenshots of JSAF with various control panels and displays are shown in Figure 3 and Figure 4 below. JSAF is a non-deterministic simulation, so the same scenario file can be loaded into JSAF resulting in different outcomes of the mission each time. Currently, we have incorporated an Air Tasking Order (ATO) parser that reads in a file extracted from the Air Operations Data Base (AODB) within TBMCS, parses the file and produces a spreadsheet that JSAF can read. This is a manual process that we would like to see automated in the future. The drawback of this is that the ATO does not contain “all” the data required to build the mission, so other data is required to augment the ATO in order to exercise a full mission. Along with the ATO parser, an effort is underway to develop a scenario generation tool that can

be incorporated into the simulation to streamline the process of creating scenarios. This tool would generate the entire scenario which JSAF would then read in and execute.

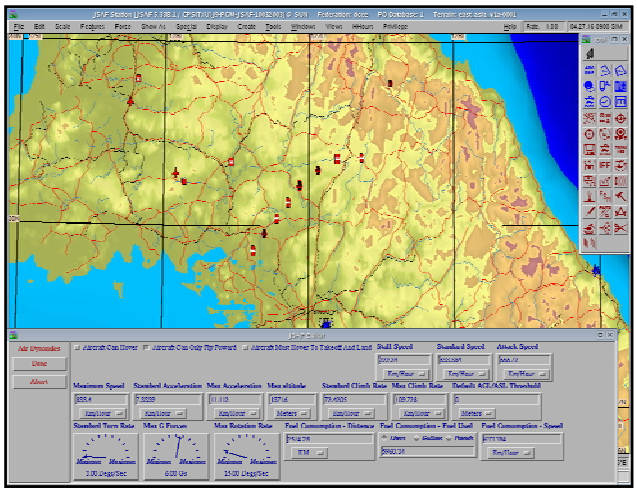

Figure 3 Snapshot of JSAF GUI with JSAF editor displaying controls over a specific aircraft.

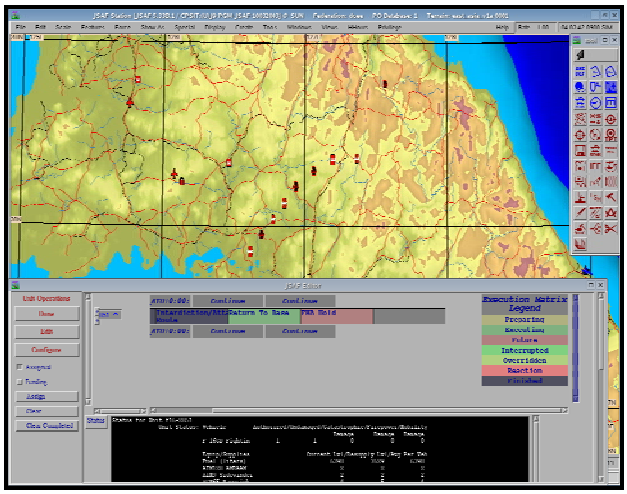

Figure 4 Snapshot of JSAF GUI with status window and mission control editor.

JSAF controls entities within the battlespace using task frame functionality for routine tasks. Air representation can also be controlled via TacAir Soar. SOAR is the artificial intelligence (AI) component for air. It is a real-time expert system that provides automated pilot

behaviors. Task frame control is the primary method for controlling the aircraft within our simulation. We plan to incorporate SOAR entities in order to accept live nine-line messages sent via Air Force standard message formats that can be transmitted by a variety of means, including voice, in order to accomplish dynamic re-tasking of aircraft.

JSAF models objects at the entity level which is a fairly high resolution. Yet it can also support a reduced resolution level for those entities that do not have high fidelity requirements; thereby allowing the operator control over the number of entities contained within the simulation. JSAF has the ability to run over 30,000 entities and up to 100,000. Our environment contains nine systems that operate JSAF. Since JSAF can run in multiple modes, most of our systems are running JSAF as back-ends (simulation only, no GUI) and just a few are running in pocket mode (front end GUI and backend together).

Command and control behaviors and architectures are realistically simulated, as are sensors, logistics, weapons effects, and entities' reactions to various combat stimuli. The synthetic environment is a representation of real world terrain, oceans, and weather conditions that affect the behaviors and capabilities of the synthetic forces. Such interactions include line of sight, time of day, weather conditions, clouds, currents, etc.

JSAF runs on PC's under the Linux operating system and is also supported on Sun, SGI, and IBM hardware and is easily ported to most versions of Unix. Our JSAF federates operate across nine systems. Six of these systems are P4's, 3.06 GHz with 3GB of memory running Linux Redhat 9.0 along a 100Mb connection. The other three systems are PC Athalons, 2.19GHz with 3GB of memory running Linux Redhat 9.0 along a 100Mb connection.

6.5.2 OASES

Ocean, Atmosphere and Space Environmental Services (OASES) is a simulated (or synthetic) natural environment (SNE) HLA federate which takes recorded or live uniform or gridded weather and distributes it to the simulations in the JSAF federation. It contains a suite of applications for creating and managing a three-dimensional, time-varying, digital representation of the natural environment. Simulated environments created by OASES are based on authoritative, validated numerical models; typically the same models that are used by METeorological/OCeanographic (METOC) personnel in support of real-world military operations[24].

An illustration of the OASES system architecture, including its primary subsystems and the principle data flow between them, is shown in Figure 5. Only some of these subsystems are exercised within our environment. A live weather feed is provided to us by the Navy for a specific region of interest which is uploaded to the GRIB repository. From within our environment, the data from this repository is ftp'd to our environment, is consumed by the ingestor subsystem and sent to the OASES database. The data within the database is then edited to be read by the Visualizer (illustrated in Figure 6), and the transformer converts the data into HLA formatted messages and publishes these messages at the appropriate time. OASES publishes to DTSim to determine the effects that the weather has on the terrain and then publishes this information to JSAF. Because weather affects more than just terrain (e.g. sensors, radar, etc.), JSAF subscribes to the weather messages from OASES as well in order to simulate the effects that weather has on entities within the mission space.

OASES operates as a federate and runs on a P4M laptop PC, 2.19GHz system with 1GB of memory running Linux Redhat 7.3 (required version) across a 100Mb connection.

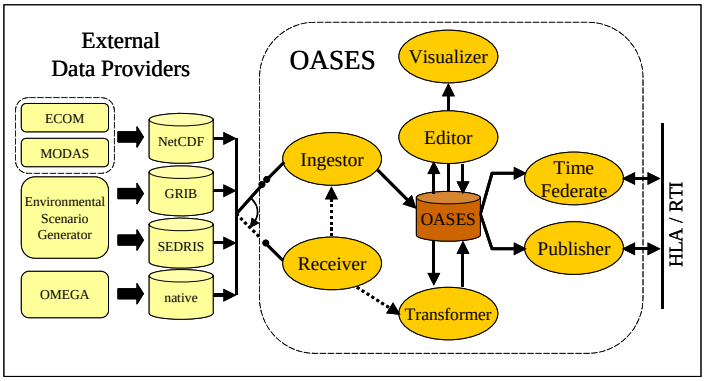

Figure 5. Schematic Architecture of OASES data flow

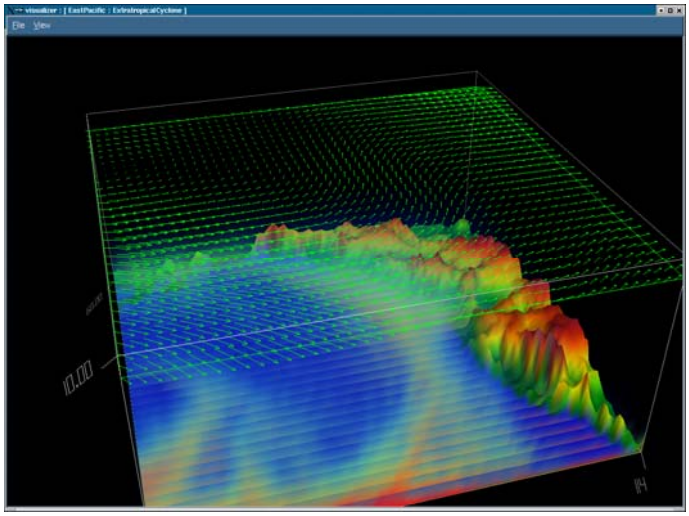

Figure 6 OASES Visualization

6.5.3 UAVSim

Unmanned Aerial Vehicles (UAVs) are utilized within the battlespace for information gathering within hostile areas. Ground Moving Target Indicator (GMTI) is a radar that is used for detecting moving targets. The GMTI radar on the UAV generates detections for each of the

targets that it sees and it outputs this information in NATOEX format. The detections from the GMTI radar are referred to as MTI data. This MTI data can be used for evaluating tracking algorithms[25].

UAVSim is a DIS simulation; therefore, the DIS/HLA gateway is needed to communicate with JSAF. Via the gateway, JSAF entities and their locations are converted to DIS format where they are consumed by UAVSim. UAVSim then displays the JSAF entities within its environment. In turn, UAVSim publishes its objects as DIS objects. The objects are converted by the gateway back to HLA and transmitted to JSAF. JSAF consumes these objects and their locations, and interacts with them as the simulation executes.

UAVSim owns the UAV entities within the simulation. The UAV entities (Predator and Global Hawk) and ground truth tracks are consumed by the simulation via a text file. As the simulation executes, the UAVSim flight model flies the aircraft within the battlespace in a racetrack pattern. As the GMTI sensor detects moving targets on the ground, it forms the nominated tracks and outputs the data in NATOEX format. In addition to NATOEX, UAVSim outputs track information via Link 16 messages which can then be consumed by other federates that accept this particular message format.

With the development of the NATOEX/HLA converter, un-nominated derived tracks output by JSAF can be consumed by UAVSim. UAVSim can then read in these tracks, detect them and, again via the gateway, publish nominated tracks back to JSAF. This would eliminate the need for a track file to be input to UAVSim and the simulation could respond dynamically to the un-nominated derived tracks offered by JSAF. This would imply that JSAF would be running in two different modes; ground truth mode and perception mode.

UAVSim is a DIS federate that runs on a P4M laptop PC, 2.19GHz system with 1GB of memory running Linux Redhat 7.3 (required) on a 100Mb connection.

6.5.4 DTSim (Dynamic Terrain Simulation)

DTSim is an HLA federate that provides a simulation of dynamic terrain such as roads, tank ditches, berms, fighting positions, craters, multi-state buildings and bridges. This federate is responsible for modeling global changes and secondary effects within the battlespace. Such changes and events include detonations and their effects on buildings and on the surface of the terrain. The DTSim models changes that occur to both existing fixed objects, and on newly added objects constructed during wartime. Included with DTSim is a component called DTScribe which is responsible for storing and transmitting changes in the terrain. As detonation events occur within JSAF, these interactions are published to DTSim. Along with detonations, weather data from OASES is published to DTSim in real-time. Both weather and events that affect the terrain are consumed by DTSim, which processes the data; whereupon DTScribe publishes the results to JSAF in terms of hydrological and topographical messages[26]. Figure 7 displays a snapshot of the DTSim display and configuration panel.

DTSim was provided as part of the JSAF distribution and operates as an HLA federate within the environment. The DTSim federate runs on PCs running Linux and a variety of Silicon Graphics, Inc. (SGI) workstations. The most common platform is a PC with a minimum of 512 Megabytes of RAM, and 700+ Mz processor, and enough disk space to hold the required CTDB and dynamic terrain data base (DTDB) databases[26]. We have DTSim running on a JSAF machine; a PC Athalon 3200+, 2.19 GHz system with 3 GB of memory running Linux RH 9.0 on a 100Mb connection.

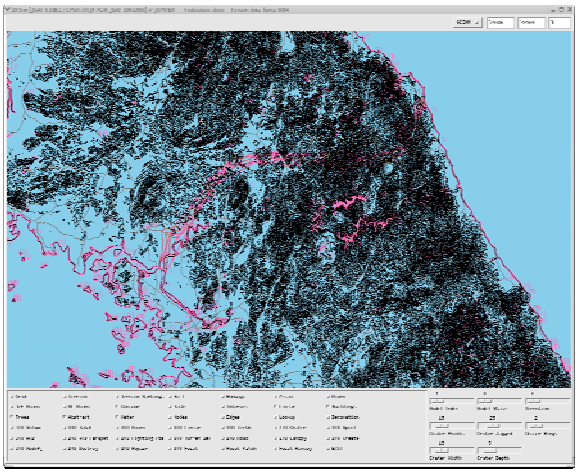

Figure 7. DTSim display with feature options.

6.5.5 Clutter

The clutter federate augments the battlespace with mobile entities such as buses, civilian vehicles, trucks, etc., that travel along roads in order to introduce a bit of confusion or noise to the battlespace. Clutter was provided as part of the JSAF distribution and operates as a federate within the environment. It runs on a JSAF machine; a P4 PC, 3.06GHz system with 3 GB of memory running Linux RH 9.0 on a 100Mb connection.

6.5.6 Simulation Network News (SNN)

SNN is a flexible event monitoring federate which summarizes events that occur within the environment such as weapons fire and damage assessment. It summarizes these events, prints them out, logs them, and maintains the totals in real time. SNN was provided as part of the JSAF distribution and operates as a federate within the environment. It runs on a JSAF machine; a PC Athalon 3200+, 2.19 GHz system with 3 GB of memory running Linux RH 9.0 on a 100Mb connection.

6.5.7 Theater Battle Management Core Systems (TBMCS)

TBMCS is the Combat Air Force (CAF) information system as well as a decision support system. It is the primary C2 system for air battle management. It supports combined and joint air operations for the Joint Forces Commander (JFC). It allows warfighters - pilots, navigators, weapon control officers, planners, intelligence officers - to access information and see a common picture of air operations. The TBMCS warfighting system integrates a suite of C2 applications, and a full range of air mission functions, sensor data and intelligence gathering, and automates many elements that comprise the planning and execution phases for theater air operations. TBMCS functionality includes intelligence processing; air campaign planning, execution and monitoring; aircraft scheduling; unit-level maintenance operations; unit and force-level logistics planning; and weather monitoring and analysis. For our purposes, we are interested in communicating with a mere handful of the systems contained within TBMCS.

The components of TBMCS that we are interested in communicating with are the Air Operations Data Base (AODB) for the ATO, the MIDB for intelligence data (including target information), the Situation Awareness and Assessment (SAA) module (and possibly the Common Operating Picture (COP)) for current situational data and the Track Management Data Base (TMDB) to push or pull track data.

We have an unclassified and classified local version of TBMCS at our facility. Initially, we plan to work with the unclassified version and migrate toward the classified version as exercise requirements deem it necessary to do so. On the unclassified side, the C4ISR gateway will act as the router between the JSB-RD simulation environment and the operational system. On the classified side, we will need to incorporate data classification/declassification tools to support the transfer of information between the two classification levels of data.

6.5.8 JSB-RD Viewer Application

Although the visualization component of the environment has yet to be appointed an official name, its purpose is to provide the user with situational awareness (SA) and control over what is displayed within the visualization space. As shown in Figure 8, the architecture is distributed amongst multiple CPU's (information servers) and monitors (information displays). This design includes 3 servers with 7 display units; however, with this arrangement, the user has the flexibility to configure the visualization environment to have as many or as few displays as he prefers. An added benefit to this configuration design is that it lends itself well to enabling distributed visualization. CPU's and monitors are inexpensive; leaving this particular design a low-cost alternative to a larger screen projection unit that displays many different windows. The visualization application is written in Java and uses the JView visualization toolkit developed by AFRL/IFSB.

The individual servers subscribe to the data published by the XML gateway parse the XML messages and display the information on the individual monitors. A sample configuration for the displays is shown in Figure 9. Each display unit has the ability to present the data in a different format/view/resolution, etc. The content/format of the information for each window can be tailored by the user. As the data is received by the server, the server distributes the information to the appropriate display. Each display also has the ability to be mapped to other displays, such that as the information or area of interest (AOI) changes in one display, the other displays that are mapped to that one display will also change (indicated by the yellow arrows).

Because the displays are interactive, the user has the ability to select alternate AOI's, access drill-down information, task/re-task aircraft from the map displays, etc. In this case, the servers act as publishers by publishing either requests or commands/tasks to the XML gateway

which, in turn, publishes the messages to the other federates within the environment. Again, the purpose of the visualization environment is to provide the user with SA and control over what is displayed within the mission space.

The specific implementation of the data mapping/distribution is currently being discussed. Although it is possible to implement this approach to visualization, it is left to the developers/integrators of our environment to ascertain which approach is best.

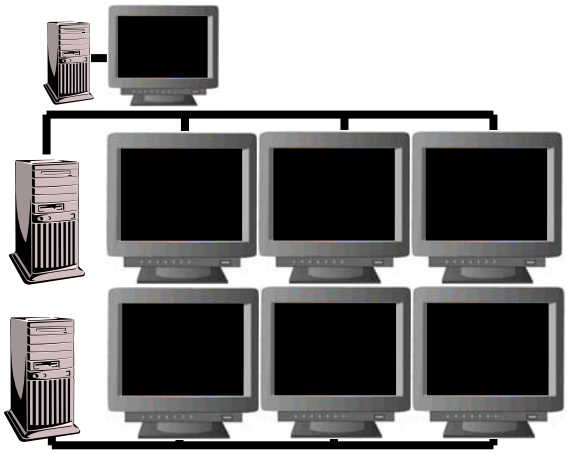

Figure 8 Hardware configuration of visualization component for situational awareness.

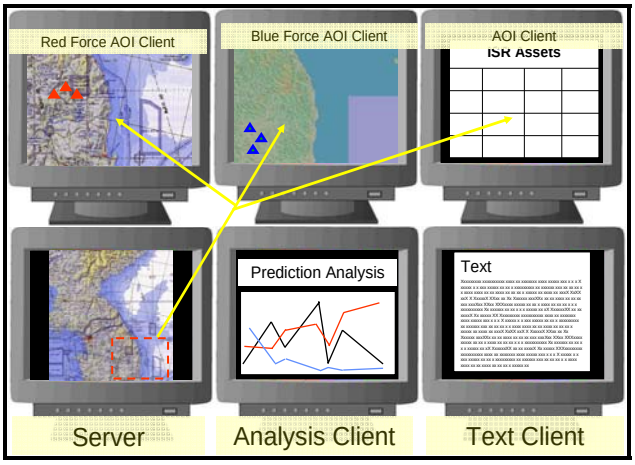

Figure 9. An example of a possible visualization configuration displaying disparate sources of information.

7.0 Conclusion

The original intent of the project was to explore and extend computational models of air operations using control theoretic approaches in complex simulation environments. All of the work reported here represents the establishment of the necessary environment. Initial staffing issues resulted in significant program delays. Our choice of the JSAF federation has had mixed results. JSAF is owned and maintained by Joint Forces Command and is extensively used to support their Joint Futures Lab. They have provided AFRL/IF with excellent support over the course of this effort. JSAF however must be considered experimental software. Each time the developers make changes in support of JFCOM experiments, some other portion of the software is broken. This has caused us to expend many hours in code repair for our own use. Despite these issues we have developed a robust capability for experimentation and in conjunction with contractual work performed by RAM laboratories Inc[27]. we will shortly be able to complete the implementation envisioned in section 4.2.

---

[1]C. Tsokos and R. Thrall; Editors, Workshop on Decision Information for Tactical Command and Control, Academic Press, NY (1979)

[2] Proceedings for Quantitative Assessment of Utility of Command and Control Systems, National Defense University, Washington D.C., Report MTR-80W00025 (1980).

[3] A. Levis and M. Athans, "The Quest for a C3 Theory: Dreams and Realities", *Science of Command and Control: Coping with Uncertainty*, S. Johnson and A. Levis; Editors, p. 4, (1988)

[4] C. Jones, "Introduction", *Toward a Science of Command, Control, and Communications*, C. Jones; Editor, p.1,(1993)

[5] A.Woodcock, J. Dockery, "Models of Combat with Embedded C2 I:Catastrophe Theory and the Lanchester Equations," *International Command and Control, Communications and Information Systems Journal*, vol.2, no.3, pp. 34-62, (1988)

[6] J. Dockery, A.Woodcock, "Models of Combat with Embedded C2 II:Catastrophe Theory and Chaotic Behavior," *International Command and Control, Communications and Information Systems Journal*, vol.2, no.4, pp. 17-51, (1988)

[7] A.Woodcock, J. Dockery, "Models of Combat with Embedded C2 III:Catastrophe Recruitment, Disaffection, and the Tactical Control of Insurgents," *International Command and Control, Communications and Information Systems Journal*, vol.3, no.1, pp. 5-38, (1989)

[8] J. Dockery and A. Woodcock, "No New Mathematics! No New C2 Theory!", *Toward a Science of Command, Control, and Communications*, C. Jones; Editor, Progress in Astronautics and Aeronautics vol. 156, A.R. Seebass, Editor-in-Chief, AIAA Washington D.C., p.64,(1993)

[9] G. Hopple, "Decision aiding dangers:The law of the hammer and other maxims," *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-16, no. 6, pp. 967-981, Nov-Dec (1986)

[10] S. Andriole and S. Halpin, "Introduction: Perspectives on Command, Control, and Information Technology," *Information technology for command and control: methods and tools for system development and evaluation*, S. Andriole and S. Halpin; Editors, IEEE NY, pp. 1-7,(1991)

[11] J. Wohl, Force Management Decision Requirements for Air Force Tactical Command and Control, IEEE Trans. Syst., Man, Cybern., vol. SMC-11, no. 9, p.618, Sept. (1981).

[12] R. Farrell, S. Bonder, L. Proegler, G. Miller, and D. Thompson, Capturing Expertise: Some Approaches to Modeling Command Decisionmaking in Combat Analysis, IEEE Trans. Syst., Man, Cybern., vol. SMC-16, no.6, pp. 766-773, Nov./Dec. (1986)
[13] J. Wohl, D. Serfaty, E. Entin, J. Deckert, and R. James, Human Cognitive Performance in Antisubmarine Warfare: Situation Assessment and Data Fusion, IEEE Trans. Sys., Man, Cyber., vol. 18 no.5, pp. 777-786, Sept./Oct. (1988).
[14] G. Klein, Naturalistic Models of C3 Decision Making, *Science of Command and Control: Coping with Uncertainty*, S. Johnson and A. Levis; Editors, pp 86-92 (1988)
[15] F. O'Brien, J. O'Neil, "Decision-Making and Currency for C2 Systems," *Proceedings of the 1999 Command and Control Research and Technology Symposium*, Naval War College, Newport RI, 29 June- 1 July (1999)
[16] A. Levis, "Colored Petri Net Model of Command and Control Nodes," *Toward a Science of Command, Control, and Communications*, C. Jones; Editor, Progress in Astronautics and Aeronautics vol. 156, A.R. Seebass, Editor-in-Chief, AIAA Washington D.C., p.181-192, (1993)
[17] E.Entin, "Optimized Command and Control Architectures for Improved Process Performance," *Proceedings of the 1999 Command and Control Research and Technology Symposium*, Naval War College, Newport RI, 29 June- 1 July (1999)
[18] R. Choisser and John Shaw, "Headquarters Effectiveness Assessment Tool," *Toward a Science of Command, Control, and Communications*, C. Jones; Editor, Progress in Astronautics and Aeronautics vol. 156, A.R. Seebass, Editor-in-Chief, AIAA Washington D.C., p.39-61, (1993)
[19] Morse S., Private communication.
[20] L.P. Kaelbling, M.L. Littman, and A.R. Cassandra, Planning and acting in partially observable stochastic domains, Artificial Intelligence 101 (1998), 99-134.
[21] J.Wohletz, D. Castanon, and M. Curry, "Closed Loop control for Joint Air Operations", American Controls Conference, 2001.
[22] D. Bertsekas, D.Castanon, "Rollout Algorithms for stochastic scheduling problems" Journal of Heuristics, Vol. 5 No. pp 89-108, Apr 1999
[23] K.Astrom, "Optimal control of Markov decision processes with incomplete state estimation." J Math Anal Applic 10 pp 174-205, 1965.
[24] *OASES User Manual*, February 2003, Northrop Grumman Information Technology Defense Enterprise Solutions)
[25] *UAV SIM USER MANUAL*, Distributed with UAVSim software
[26] *DT Sim / DT Scribe User's Guide*, Lockheed-Martin Information Systems Advanced Simulation Center, December 2001
[27] R. McGraw, "Software Infrastructure to support Dynamic Situational Awareness and Prediction (DSAP) capabilities" AFRL-IF-TR-2006-153, May 2006