JOURNAL OF AEROSPACE COMPUTING, INFORMATION, AND COMMUNICATION
Vol. 2, February 2005

Mixed Initiative Command and Control of Autonomous Air Vehicles

Milton B. Adams* and Janet A. Lepanto†
Charles Stark Draper Laboratory, Cambridge, Massachusetts 02139

and
Mark L. Hanson‡
Charles River Analytics, Cambridge, Massachusetts 02138

Future military air operations will include multiple classes of unmanned air vehicles with both sensing and strike capabilities. The increased complexity of the associated command and control required for planning and execution of multi-target missions for these unmanned platforms was addressed by DARPA's Mixed-Initiative Control of Automa-teams (MICA) Program. Under the MICA Program, the Draper Team developed and implemented a hierarchical architecture for mixed initiative command and control of multiple teams of autonomous air vehicles prosecuting large numbers of targets simultaneously. Our approach achieved dynamic replanning of the teams' missions in response to a continuously evolving target set and knowledge of the enemy air defense threat environment.

In current operations, planning and execution for the operational functions of ISR and strike are performed locally with limited cross-function interaction, such that each function has a limited knowledge of the others' plans and execution status. The result is implicit coordination that is neither optimized nor robust, particularly when unforeseen changes in one function's plans or execution status results in inconsistencies and possible constraint violations in the other functions. Our prototype system provides an integrated controller architecture that explicitly coordinates these operational functions. Our architecture supports the creation of integrated missions across functions, the temporal decomposition of missions that span functions, and iterative planning with dynamic constraints and objectives. Our implementation also supports local (decentralized) execution monitoring and replanning and distributed information awareness.

In our prototype Controller implementation, ISR and strike objectives and solutions are explicitly coupled. For target observation planning, strike decisions influence the decisions as to which targets need to be (further) observed, and ISR results determine which strike targets have been found. ISR results influence the approach to - and weapon selection for - targets, while weaponeering considerations determine the nature of the required observations since different weapon choices impose different target location accuracies. ISR is also tasked to gain knowledge of the air defense threat laydown to support low-hazard strike route determination. Coupled ISR and strike activities are planned dynamically in a unified fashion that significantly shortens the kill chain timeline.

Received 20 August 2004; revision received 16 December 2004; accepted for publication 17 December 2004.

*Division Leader; Control, Information and Decision Systems Division; 555 Technology Square, Mail Station 3F. AIAA Senior Member.

†Program Manager; Tactical Systems Directorate; 555 Technology Square, Mail Station 55.

‡Senior Scientist; 625 Mt. Auburn Street. AIAA Member.

An Information Model provides situation awareness through two classes of information states. Grid-based information provides probability distribution information about the number and types of undetected objects (including targets and hazards) in specific geographic regions. Track-based information provides probabilistic information about individual objects.

Our Human-System-Interaction (HSI) design allows an operator to interact with the controller hierarchy in accomplishing mission objectives. The system has been designed to support two types of HSI operators: Strategic level operators and Tactical level operators, where Strategic and Tactical are associated with the levels of our hierarchical controller. Strategic operators are assumed to be more experienced personnel. They provide high-level commander's intent and are responsible for making global decisions that affect all vehicles and teams in the battlespace. Tactical operators are assumed to be more junior personnel whose primary mission is the supervision and management of specific UAV teams. They make local decisions relative to their vehicles/teams and objectives, such as reconfiguring teams, altering flight paths, and participating in weapons employment events.

Acronyms

AIM = Awareness Intent Matrix
AOR = Area of Responsibility
AVM = Aircraft Value Matrix
BDA = Battle Damage Assessment
C2 = Command and Control
CaP = Coverage and Precedence
CEP = Circular Error Probable
CORBA = Common Object Request Broker Architecture
CTPP = Cooperative Task/Path Planning
DARPA = Defense Advanced Research Projects Agency
DIM = Damage Intent Matrix
DoD = Department of Defense
EBO = Effects-Based Operation
ECM = Electronic Countermeasure
EO = Electro-Optical
EOB = Enemy Order of Battle
ESM = Electronic Surveillance Measures
EW = Early Warning
EZ = Engagement Zones
GMTI = Ground Moving Target Indicator
HARM = High-speed Anti-Radiation Missile
HIS = Human System Interface
IADS = Integrated Air Defense System
IP = Integer Programming
IPB = Intelligence Preparation of the Battlefield
ISR = Intelligence, Surveillance, Reconnaissance
JMS = Java Messaging Service
KIM = Kill Intent Matrix
LGB = Laser-Guided Bomb
LOS = Line of Sight
LW = Large Weapon
MICA = Mixed-Initiative Control of Automa-teams
MVC = Model View Controller
OEP = Open Experimental Platform
PMF = Probability Mass Function

POI = Points of Interest
ROE = Rules of Engagement
RWR = Radar Warning Receiver
SA = Situation Assessment
SAR = Synthetic Aperture Radar
SAM = Surface-to-Air Missile
SC = Small Combo
SEAD = Suppression of Enemy Air Defenses
SEB = Semantic-to-Engineering Bridge
SPARTY = Self Propelled Artillery
SSM = Surface-to-Surface Missile
SW = Small Weapon
TCTA = Team Composition and Task Allocation
TDT = Team Dynamics and Tactics
TEL = Transporter Erector Launcher
UAV = Unmanned Aerial Vehicle
UI = User Interface

I. Introduction

IF current trends in the proliferation of types of unmanned air vehicles (UAVs) continue, military air operations of the future are likely to include multiple classes of unmanned air vehicles with both sensing and strike capabilities. Concomitant with the increase in the number and types of UAVs will be an increase in the complexity of the associated command and control required for planning and execution of multi-target missions for these unmanned platforms. This command and control problem was the focus of DARPA's Mixed-Initiative Control of Automa-teams (MICA) Program. A hierarchical architecture for mixed initiative command and control of multiple teams of autonomous air vehicles prosecuting large numbers of targets that was developed and implemented under that program is the subject of this paper. This approach achieved dynamic replanning of the teams' missions in response to a continuously evolving target set and knowledge of the enemy air defense threat environment.

There are a host of novel components embodied in our design and implementation. In order for our planning algorithms to generate plans, they must be provided knowledge of the threat and target environment as well as the commander's intent. We represent commander's intent in a manner that captures the importance (by geographic region and time) of destroying targets of each type, of finding targets of each type and of collecting post-strike target damage information. Our Commander's intent representation also captures the level of risk per target type as a function of the value of unmanned vehicles that may be attritted in prosecuting that target. We have developed unique probabilistic models for evaluating expected accomplishment of commander's intent for candidate plans. These models include a novel approach to determining the value of information that may be collected by planned ISR activities.

We employ what we refer to as an *Information Model*[3] to capture a probabilistic description of the threat and target environment. The content of the Information Model is used by the planning algorithms both to make decisions about threat avoidance (or threat prosecution) and to decide what are the best available {sensor-weapon-target} sets to include in a candidate plan. That is, for a given target, we must decide, based on available sensing and weapon resources and the current (uncertain) knowledge of the target state (e.g., target type, target location, target damage state), whether to sense the target before a strike to reduce uncertainty, which weapon to employ as a function of the target type and uncertainty in target state at the time of strike and which sensor to employ to collect target damage information after a strike.

The planning problem that we formulate and solve is extremely complex in that it simultaneously addresses scheduling and allocating teams of unmanned vehicle assets to prosecute targets (which vehicles use which sensors against which targets over time), the routes that they take through a potentially dense air defense system, and the use of cooperative electronic countermeasures (jamming) and employment of decoys to ameliorate the risk of the air defense system. In order to solve this highly coupled, large-scale optimization problem, we employ a multi-level decomposition methodology,[2] augmented by a novel use of so-called *Composite Variables*[7] to further reduce the complexity and, ultimately, to make the solution computationally feasible.

Another contribution of our approach is the design of an extensive set of mechanisms that enable operator interaction with automation. Many of these mechanisms are built on the concept of *Semantic-to-Engineering Bridge*

(SEB).[2] SEBs are implemented as intelligent agents that enable the translation of information between the information representations employed within the automation algorithms and the information representations useful to an operator in making battlespace decisions. Another component of the Human-System-Interaction design is a concept we refer to as *Management Style*. Management Style refers to the mechanisms by which operators convey their approval or rejection of decisions made by the automation (e.g., replanning decisions). Management style *modes* range from *full automation* (accept all decisions made by the automation), to *exception* (accept a decision made by the automation unless the operator rejects it within a pre-specified time) and to *consent* (operator must explicitly approve of all decisions made by the automation). We have enhanced this concept by allowing for situation-dependent modes, where the "situation" characterizes a condition or classes of conditions that initiated the automated replanning activity or modes that are a function of the class of change that was made to the plan. For instance, an operator may want to provide for a full automation approval of all decisions made by the automation to change vehicle routes due to new threat information. In contrast, an operator is likely to demand management by consent for all strike decisions developed by automation.

In the following, a closed-loop, dynamic planning and execution system that integrates planning for fleeting targets, multifunction platforms and aggressive timelines is discussed, and representative experimental results of applying that prototype in the context of the MICA program's Open Experiment Platform (OEP) simulation environment are presented. The mixed-initiative human system interface (HSI) that supports operator participation in plan generation, plan execution, and situation awareness for teams of multiple, heterogeneous unmanned aircraft is also described.

II. Problem Description

In current operations, planning and execution for the operational functions of ISR and strike functions are performed locally with limited cross-function interaction, such that each function has a limited knowledge of the other functions' plans and execution status. The result is implicit coordination that is neither optimized nor robust, particularly when unforeseen changes in one function's plans or execution status results in inconsistencies and possible constraint violations in the other functions. Our prototype system provides an integrated controller architecture that explicitly coordinates these operational functions. Our architecture supports integrated missions across functions, temporal decomposition of missions that span functions, and iterative planning with dynamic constraints and objectives. Our implementation also supports local (decentralized) execution monitoring and replanning and distributed information awareness.

The MICA objective is to achieve closed-loop, synergistic human and autonomous decision-making at multiple levels in the command and control hierarchy. Draper's approach to MICA decision-making is to maximize the expected value of an objective function that rewards destruction of certain entities, penalizes damage to other types of entities, and rewards the collection of information. A typical plan for maximizing this objective function comprises the selection, sequencing, and scheduling of cooperative activities for teams of UAVs. The cooperative activities include jamming, use of decoys, where and when to point which sensors and in what modes, and where and when to release which weapons.

Given what is a priori known and expected on the basis of Intelligence Preparation of the Battlefield (IPB) about the current state of the battlefield (Red, Blue, White), and the available resources (e.g., UAVs, weapons, etc.) and their current state, activities for the available aircraft are selected, sequenced, and scheduled to maximize the total expected *Value minus Cost. Value* is computed as a function of commander's intent (described below), and is accrued from mission activities of finding, identifying, and destroying targets, and confirming battle damage assessment (BDA). *Cost* is computed as a function of UAV attrition, UAV detection, and cost of resources. Both *Value* and *Cost* are expectations taken over the probabilistic characterization of the battlespace state contained in an *Information Model* (described below) and the probabilistic characterization of sensor performance, weapon performance, and Red air defense system performance.

III. Situation Awareness – The Information Model

The decision-making processes for MICA levy a unique set of basic information processing requirements to achieve and maintain situation awareness. Achieving situation awareness includes four key steps. The first step is ***sensor looks***, in which ISR platforms direct their EO and SAR sensors with various modes (e.g., spot mode or Moving Target Indicator (MTI) mode) at various poses in the Theater. The sensor looks produce raw sensor data (e.g., collections of SAR returns). The second step is ***signal and image "preprocessing"*** to convert raw measurements into geo-registered "image" type data representations. The third step is ***manual or automatic interpretation of the preprocessed data***, which processes image-type by image interpreters that detect, classify, and locate Red and White entities in the scenes, and develop probabilistic summaries of the outcomes of the sensing process and the entities observed. This is assumed to be performed on a sensor-look-by-sensor-look basis. The final step is ***fusion of measurements into an "Information Model"*** to maintain *tracks* for Red and White entities that have been observed, and to fuse multiple measurements associated with the same track. It also maintains statistics on which entities have not been observed (or those that were previously observed, but are no longer being tracked with reasonable confidence) as a function of location and asset type.

It is also vital to keep track of entities that have not yet been observed directly. Independent intelligence about yet-to-be-tracked entities contained in Enemy Order of Battle (EOB) provided by IPB data must be accounted for along with the history of ISR activities that friendly entities have conducted. The questions "How many are out there that we are not currently tracking?" and "How thoroughly have we recently reconnoitered that area?" are inextricably linked. Furthermore, these data must be maintained in a manner consistent with the directly observed track data – if a track is created or dropped, information on unobserved objects should be updated accordingly.

The *Information Model*[*] is an estimator that has been designed to meet the basic information requirements of the MICA planning and execution system by computing three types of information. The first type is a ***track for each observed entity***. Each track includes probabilistic descriptions of an object's location, its type (within a finite set of types), its health or damage condition, and whether it is currently moving or stationary. The second type is ***information about the undetected entities***. This includes the numbers, types, and geographic distributions of entities that have not been directly observed. It is based on both prior information about the Red order of battle and its geographic distribution, and also on the history of observations that friendly entities have conducted. The third type is ***total entity count***, which is the sum of the undetected entity and those represented by tracks. Figure 1 illustrates the inputs and outputs of the Information Model.

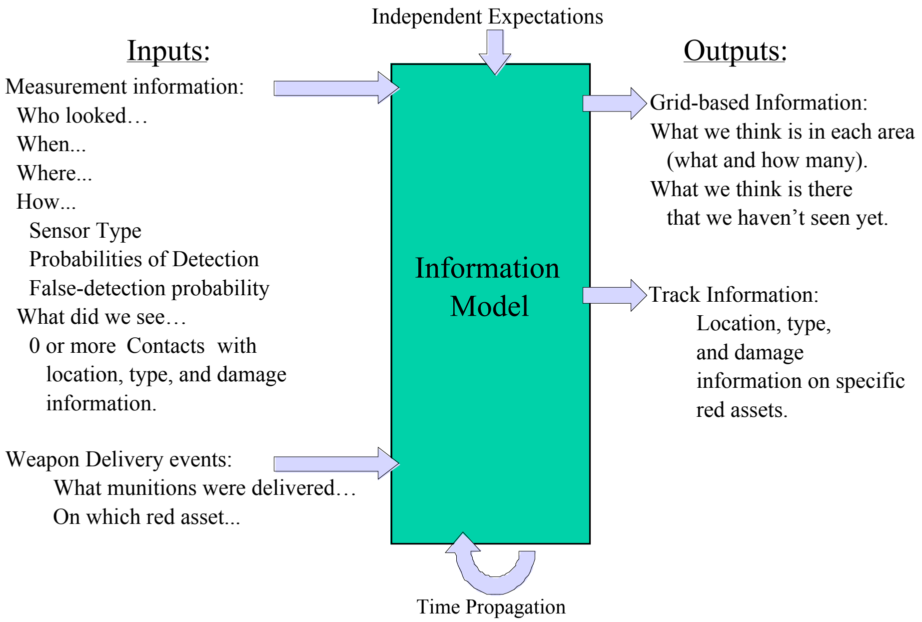


Fig. 1 Information model inputs and outputs.

To determine the value expected from striking a target, we estimate the type of the target, the degree to which the target's location is known, the entities in the vicinity that may incur secondary damage, the probability that the target will be damaged by the chosen weapons, and the probability that the aircraft delivering the weapons will be able to safely reach an area within striking range of the target. Most of these measures are based on the track

---

[*] The details of the formulation and implementation of an estimator that addresses these information requirements – traditional track processing, plus keeping track of what may remain unobserved – in an integrated manner is given in Ref. 3. Here we provide a brief overview.

information of candidate targets, but the risk of collateral damage depends on the distribution of both tracked and undetected entities in a target's vicinity.

The value expected from employing sensors in an area is a combination of the extent to which it improves the expected strike value, e.g., by improving the type or location identification of entities in the area or by finding new targets, as well as the direct value of improved information about entities in that area. This assessment depends partially on the number of entities thought to remain undetected there. The expected value is also modulated by the probability that the aircraft carrying the sensors can reach the sensing range of the area to be observed. Route planning for aircraft as they ingress, egress, or transit between activity zones is based primarily on their geographic distribution, from which we derive the probability that the aircraft can reach their destinations and the probability that they will be attritted. Furthermore, knowledge of whether a target is moving or able to move influences the ability to strike a target, the probability of recontacting a target not visited recently, and the degree of coordination required between sensing and strike assets to prosecute the target.

IV. Modeling Commander's Intent

Air operation plans are driven by commander's intent, which reflects: ***time*** (the importance of campaign phase and target time criticality), ***geography*** (the importance of region), ***target class*** (the importance of target class or grouping of targets), and ***risk*** (the importance of achieving objectives vs. loss of resources, including human). We specify commander's intent through the following four intent matrices. Each intent matrix is a function of geographic region, *r*, campaign phase, *p*, and the functional category of track or potential track, *c*, as specified by a human operator.

1) Kill Intent Matrix (KIM): Each entry of KIM, $V_K(r,p,c)$, represents the value of destroying a single track.
2) Damage Intent Matrix (DIM): Each entry of DIM, represents the value of damage assessment of a single track.
3) Awareness Intent Matrix (AIM): Each entry of AIM, $V_A(r,p,c)$, represents the value of awareness about tracks in a single grid cell beyond awareness used to support activities other than strike (e.g., routing of aircraft around threats, ground force movement).
4) Aircraft Value Matrix (AVM): Each entry of AVM represents the commander's conceived value toward the cost of losing an aircraft by type.

Activities such as finding, fixing, tracking, identifying, and assessing the damage state of a track, when used in support of strike, contribute to the overall KIM value through marginal improvements in the probability of achieving KIM value through strike rather than contributing to AIM or DIM. AIM is a specific value of awareness to support activities other than strike, such as routing of aircraft around threats or movement of ground forces. DIM is a specific value of damage assessment beyond that which is useful for determining whether additional KIM value can be obtained by striking tracks again if they have not been destroyed.

The operator can designate specific tracks as being more or less important than others in the same geographic region, campaign phase, and functional category by assigning a track-specific DIM weight, $W_D(i)$, or KIM weight, $W_K(i)$, for track *i*. For example, the operator may specify that the medium-range SAM site with Basic Encyclopedia (BE) number 113 is three times as important to strike as other medium-range SAM sites in the same geographic region, which could be represented by $W_K(113) = 3.0$. The default value of all weights is unity. Since grid cells do not move, changes to the value of specific grid cells are specified by redefining the geographic regions instead of by additional weights.

Time sensitivity of tracks and potential tracks is specified through decay constants in units of time: an *awareness decay matrix*, $\tau_A(r,p,c)$; a *damage decay matrix*, $\tau_D(r,p,c)$; and a *kill decay matrix*, $\tau_K(r,p,c)$. The operator may optionally override any of these for specific grid cells or tracks. Value is interpreted as proportional to the time rate of change in value at the time at which a particular activity occurs. Given the elapsed time from the start of the planning cycle, Δt, and the decay constant τ, value is scaled by the following time-sensitivity factor.

$$s(\Delta t,\tau;\tau_0) = \max(1,\tau_0/\tau)\exp(-\Delta t/\tau) \tag{1}$$

A nominal decay constant τ_0 is set to a large value (for example, 12 h) that is interpreted as the decay constant for time-insensitive tracks so that the time-sensitivity factor for such tracks converges to unity. An example of the time-sensitivity factor as a function of elapsed time from the start of the planning cycle is shown in the Fig. 2.

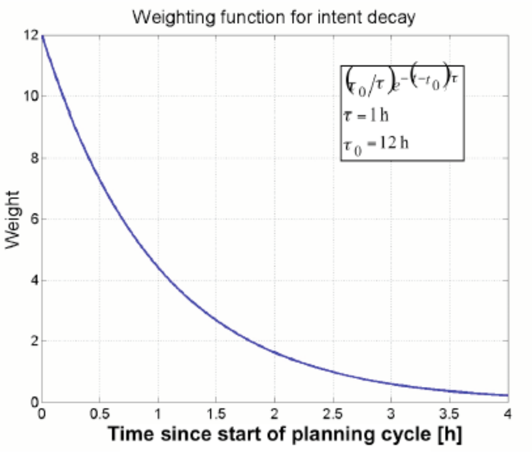


Fig. 2 Time sensitivity factor over time since start of planning cycle.

A. Commander's Intent Evaluation Models

In evaluating AIM, DIM, and KIM, we assume that geographic regions are large enough that the geographic region containing a grid cell or track is known. We define the geographic region containing grid cell *g* as *R(g)*, the geographic region containing track *i* is *R(i)*, and the campaign phase at time *t* is denoted *P(t)*. For a planning cycle that starts at a reference time t_0, the AIM value for grid cell *g* at time *t* for functional category *c* is:

$$A(g,t,c) = V_A\big(R(g),P(t),c\big)s\big(t-t_0,\tau_A\big(R(g),P(t),c\big);\tau_0\big). \tag{2}$$

The DIM value for track *i* at time *t* assuming functional category *c* is:

$$D(i,t,c) = V_D\big(R(i),P(t),c\big)W_D(i)s\big(t-t_0,\tau_D\big(R(i),P(t),c\big);\tau_0\big). \tag{3}$$

The KIM value for track *i* at time *t* assuming functional category *c* is given as follows.

$$K(i,t,c) = V_K\big(R(i),P(t),c\big)W_K(i)s\big(t-t_0,\tau_K\big(R(i),P(t),c\big);\tau_0\big). \tag{4}$$

The KIM value for a potential track in grid cell *g* at time *t* of functional category *c* is:

$$K(g,t,c) = V_K\big(R(g),P(t),c\big)s\big(t-t_0,\tau_K\big(R(i),P(t),c\big);\tau_0\big). \tag{5}$$

Some fine tune may be required in establishing appropriate scaling among AIM, DIM, and KIM, any commander's grid-cell-specific and track-specific weights, decay constants, the aircraft value matrix, and any other contributors to the objective function. For the purpose of comparing different mission options, the main goal is to ensure that all options are evaluated by a consistent objective function.

Marginal Values

In evaluating candidate/alternative plans of activities for teams and individual vehicles within teams, the planner is interested in marginal values of AIM, DIM, and KIM achieved after a specific event (look, strike, look-decide-strike are examples). This section provides a general framework for utilizing AIM, DIM, and KIM in evaluating the benefits of candidate combat operations.

The awareness value after an activity occurring at time t is given by the *decrease* in the following quantity after the activity:

$$\sum_{g \in G} \sum_{c \in C} A(g,t,c) E\{N_u(g,c)\}, \tag{6}$$

where G denotes the set of all grid cells, C denotes the set of all functional categories, E is the expectation operator, and $N_u(g, c)$ is a random variable representing the number of undetected tracks in grid cell g of type c.

The awareness function penalizes the expected number of undetected tracks of each type in each grid cell and encourages sensor looks in the cells that are most likely to contain the largest number of undetected objects. It also encourages the use of sensors that correctly identify the high-valued detected tracks because the reduction in the number of undetected tracks in that category will be greater than if the detected tracks were not identified.

Note that while the expected number of tracks after the postulated sensor look does not change because the outcome of the measurement is not known when planning the look, the expected number of *undetected* tracks after the postulated sensor look does change. This is because the number of undetected tracks represents uncertainty in the knowledge of the contents of the grid cell, and the reduction in uncertainty due to a postulated future sensor look is predictable.

The damage assessment value after an activity occurring at time t is given by the *increase* in the following quantity after the activity:

$$\sum_{i \in I} \sum_{c \in C} D(i,t,c) P_T(i,c) \max_{d \in D} P_D(i,d \mid c), \tag{7}$$

where I denotes the set of all tracks, $P_T(i, c)$ is the estimate of probability that track i is of type c, D is the set {undestroyed, destroyed}, and $P_D(i, d \mid c)$ is the estimate of probability that track i has damage state d, given that it is of type c. Although a finer representation of damage state may be used in the Information Model, values derived from DIM are only achieved by correctly determining whether or not the track has been destroyed. The probability that this determination is made correctly may depend on track type (for instance, if different threshold damage levels are required to destroy tracks of different types).

The damage assessment function rewards actions that increase the probability that the maximum-likelihood damage state is correct. This should be applied only to sensor looks. While a strike with a highly effective weapon may increase confidence that the track is in a particular damage state (namely, destroyed), the purpose of the DIM is to place explicit value on damage assessment. The type of Probability Mass Function (PMF) $P_T(i, c)$ should not be changed when evaluating this function before and after the activity; otherwise, the value may change due to different DIM weightings on different track types even if the information about the damage state does not change.

Note that while the damage state probability distribution after the postulated sensor look does not change because the outcome of the measurement is not known when planning the look, the *maximum value* over the damage state probability distribution after the postulated sensor look does change. This is because the maximum value taken over the damage state probability distribution represents uncertainty in the knowledge of the damage state, and because the reduction in uncertainty due to a postulated future sensor look is predictable.

The kill assessment value after an activity occurring at time t is given by the *increase* in the following quantity after the activity:

$$\sum_{i \in I} \sum_{c \in C} K(i,t,c) P_T(i,c) P_D(i, \text{destroyed} \mid c), \tag{8}$$

Note in this case that $P_D(i, d \mid c)$ is evaluated for a particular damage state (destroyed). This represents the actual damage state rather than our knowledge of the damage state. In other words, damage assessment does not provide the kill assessment value directly through this term because, although it improves our knowledge of the damage state, it does nothing to change the damage state itself.

The kill assessment function rewards any action that increases the probability that the track is destroyed. While this obviously applies to strike actions, it can also apply to sensor looks if, for example, the purpose is to evaluate the difference between the kill assessment value that would be achieved in an integrated look-decide-strike sequence compared with a simple strike action. The type PMF $P_T(i, c)$ should not be changed when evaluating this function before and after the activity for the same reason cited for DIM evaluation.

V. Decomposition of the MICA Problem

A. Hierarchical Decomposition

Draper developed and implemented a hierarchical solution to the large-scale, real-time optimization problem for MICA. The two principal benefits of this hierarchical approach[2,4-6] are the ability to trade off optimality and solvability, and the decomposition of the problem into tractable subproblems whose solutions can be understood, monitored, and refined by the operator through the HSI.

Draper's approach to solving the MICA problem explicitly couples ISR and strike objectives and solutions. For target observation planning, strike decisions influence which targets need to be observed, and ISR results determine which strike targets have been found. ISR results influence the approach to--and weapon selection for--targets, while weaponeering considerations determine the nature of the required observations since different weapon choices impose different target location accuracies. ISR is also tasked to gain knowledge of the air defense threat laydown, which enables us to develop strike routes that avoid these hazards. By dynamically planning coupled ISR and strike activities, Draper's approach can significantly shorten the kill chain timeline.

Our architecture uses three levels of integrated planning and execution, as illustrated in Fig. 3.

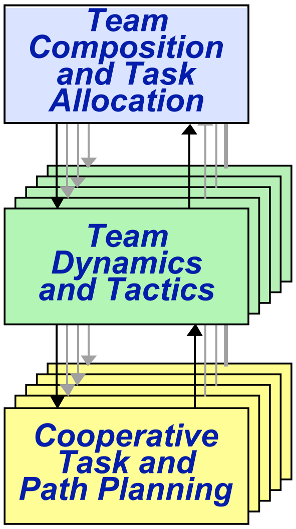


Fig. 3 MICA hierarchical controller.

The Team Composition and Task Allocation (TCTA) level partitions objectives and resources into team-sized sets, each of which defines an "option" or candidate resource-target pair for which a detailed plan is created by the lower levels of the hierarchy. The TCTA selects among team options and schedules them to maximize mission accomplishments while minimizing cost. The TCTA also defines geographic areas of responsibility (AORs) for each team.

For each candidate option generated by TCTA, the Team Dynamics and Tactics (TDT) level orders targets in the target set to maximize the value/importance achieved while minimizing cost (attrition, weapons, fuel, time), and concurrently determines optimal actions for each target. The TDT also manages uncertainty in the presence of a partially known air defense threat.

The Cooperative Task/Path Planning (CTPP) level adds detail to the TDT plan for each option by coordinating and planning for the details of the activities for each vehicle in the team to accomplish the objectives for each target (e.g., details of sensor pointing, weapon selection, and weapon release time and location, timing and pointing of jammers for Integrated Air Defense System (IADS) suppression, determining decoy trajectories and time of launch). The CTPP also determines routes for each vehicle in the team to minimize route costs (attrition, time).

B. Planning Across the Three Levels of the MICA Hierarchy

Figure 4 describes the complete process of creating a new plan for all teams across the battlespace, beginning and ending at the TCTA level. In general, this process can begin at any level as described in the next subsection.

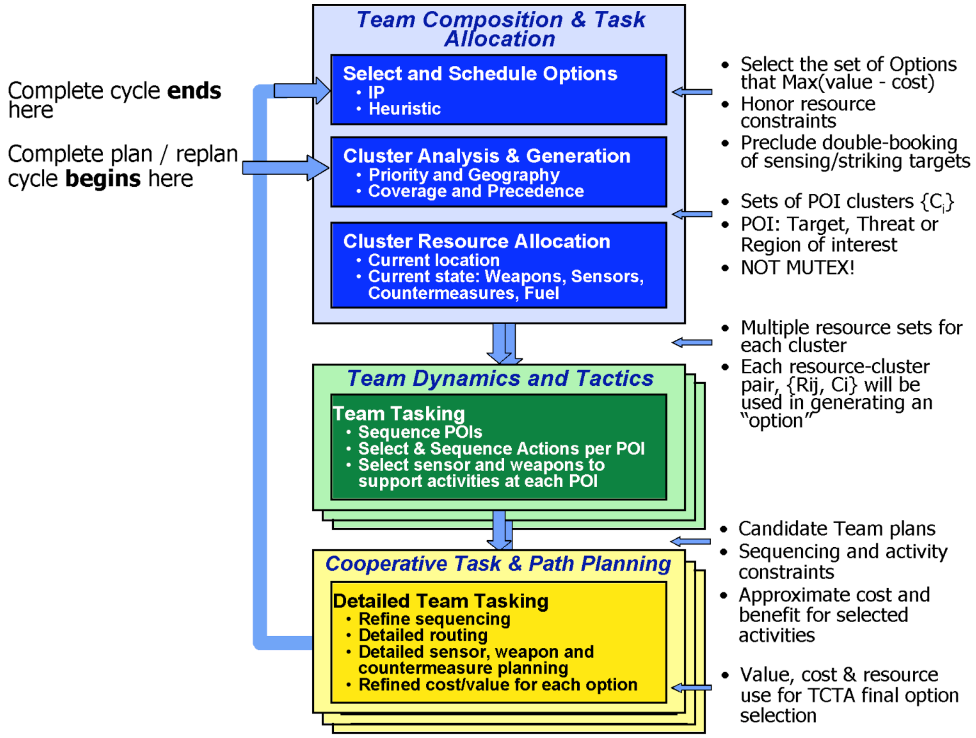


Fig. 4 Decomposition of large-scale optimization into manageable smaller problems: Principal inputs to the process are: 1) commander's intent, 2) current best estimate of battlespace state 3) available aircraft resources.

C. TCTA

Draper developed two types of innovative algorithms to achieve planning at the TCTA level: a composite-variable approach[7-9] to solve the team assignment problem, and clustering and resource allocation algorithms to define {resource – objective} pairs, each of which represents a candidate team "option" that is an input to the TDT level where details of the plan are developed. Each pair is a cluster of points of interest (POIs)[*] (objectives) and an associated team of vehicles (resources). The TCTA also imparts any ordering and grouping constraints on POIs within a cluster.

1) Composite-Variable Solution

Optimization problems of the class addressed in the MICA program are NP-complete.[7] Attempts to solve such problems optimally can result in *extremely* long execution times. This is unacceptable for real-time, closed-loop problems that require timely solutions to accommodate changes in the battlespace. An approach that is suboptimal but proved to be effective in problems of this class has been developed to address the computational complexity and timeliness issues.[7] This *Composite-Variable* approach solves the problem by mapping it into an equivalent problem for which each decision variable represents a collection of a subset of the decision variables from the original problem. For our MICA problem, the decision variables for the unabridged problem are: for every time interval, where should each Unmanned Aerial Vehicle (UAV) be and what activity should it be performing? The composite variable that we have chosen is the complete mission for a team of UAVs. We will refer to these composites, which can be viewed as "plan fragments" of the overall MICA plan for all vehicles and all teams, as "options."

The advantage of the composite-variable formulation is that it is easier to solve than the original problem. The challenge is in defining the right mapping from the original variables to composite variables and in selecting which composite variables to generate. In our case, there are exponentially many possible missions for each possible team of vehicles when considering every possible set of targets for those teams to pursue. The clustering and resource allocation approach described below has been developed to select what are expected to be a good set of team options to consider in the final solution. A Composite-Variable Master solver (Integer Program) selects the set of composites/options that maximizes the value of the effects achieved within the resource constraints.

---

[*] POI types include: Points (e.g., a track or other entity of interest) or Regions (closed convex polygon).

Clustering and Coverage Constraints and Associated Resource Allocation

Overlapping threat Engagement Zones (EZ) can provide protection ("cover") for nearby entities (threats and other targets) in complex ways. In order to be able to prosecute a protected target (e.g., a Transporter Erector Launcher, or TEL) safely, the threats (Long SAMs/Medium SAMs) that protect/cover that target must first be engaged and suppressed. We have developed approaches to analyze these coverage relationships. *In particular, target clustering algorithms* consider together the threats and targets that interact through the covering effect. *Resource allocation algorithms* use the composition of the target cluster to guide the choice of aircraft team composition (including number of aircraft, jamming, weapons, decoys, etc.). *Sequencing algorithms* use the coverage analysis to determine the possible precedence trees that would result in minimum cost approach to primary targets. These algorithms also identify those cooperative activities that must be performed sequentially and those that can be performed in parallel.

Operator Interaction with TCTA Level

Draper developed a generalization of the threat-based Coverage and Precedence (CaP) constraints that are imposed on lower levels of planning to support operator-defined ordering and precedence constraints. These CaP constraints include immediate ordering, soft ordering, and grouping.

Immediate ordering constraint of goals (e.g., targets) A and B means that if Target A is in the plan, then Target B must *immediately* precede A. *Soft ordering constraint of goals* A and B requires that if target A is in the plan and Target B is in the plan, then B must be precede A. *Grouping constraints for goals* A, B, and C mean that to achieve value for targets A, B and C, all three must be in the plan. (Note that immediate and soft ordering constraints may also apply). Although A, B, and C individually have some (small) value, additional value accrues if all three are in the plan.

D. TDT

The {resource, objective} option pairs created by the TCTA level are provided as inputs to the TDT level. The resources are a set of aircraft (a team) and the objective is the set of POIs assigned to the team. Each option pair has an associated set of CaP constraints developed by the TCTA clustering algorithms and/or provided as input by the operator. Constrained by the CaP, the TDT solves a multivehicle traveling salesman problem that orders the POIs and selects actions for each POI to maximize their value per the commander's intent. That is, for each POI, the TDT selects the optimal combination of sensing actions and weaponeering to achieve maximum value. In doing so, there may be additional supporting activities identified to reduce the risk imposed by the IADS. These activities include jamming and decoy deployment countermeasures. The ordering of POIs is constrained by the CaP constraints to ensure that the IADs threat is addressed with as low a risk as possible. The actions at each POI are selected by considering all combinations of sensing and shooting actions (e.g., look, look-shoot, look-shoot-look, shoot only and the choice of stand off or shorter range weapons and sensors) with a locally optimized choice of sensor and/or weapon for each action or combination of actions. The local optimization is constrained by the available types of weapons and attempts to choose a sensing action that is consistent with the choice of weapon and that reduces uncertainty in target type to ensure that Rules of Engagement (ROEs) regarding collateral (noncombatant) damage and fratricide are obeyed.

During the execution of a selected TDT plan, a complete TDT command and control (C2) Node (see Fig. 5) is created for the team to perform the additional functions of Monitoring, Diagnosis and Plan Execution. Monitoring and Diagnosis ensure that a replanning event is triggered when a new objective (POI) is found in the AOR defined for that team. (Note that the new POI is not necessarily discovered by the team to which it has been assigned; e.g., it may be a SAM site discovered via multilateration by another team.). The team's AOR is defined dynamically by the Execution function at the TCTA level which partitions the battlespace into regions that are near the POIs and trajectories of each team. Before replanning to add a newly found POI, the TDT node will re-analyze the CaP to ensure that a new SAM threat is addressed by the team in the appropriate order. A replan is also triggered if there is attrition of a team member.

E. CTPP

The CTPP adds detail to each team's plan that is created at the TDT level for prosecuting ordered POIs and POI activities. The CTPP chooses sensors by considering both sensor range (when standoff is important to reduce risk) and sensor-target pairing. The CTPP also schedules, modes, and points selected sensors for area searches. The CTPP provides jammer scheduling, moding, and pointing, and coordinates trajectories of team members to support standoff jamming by one team member to protect another team member entering the EZ of a SAM site. The CTPP is also responsible for minimum risk routing and planning for countermeasure decoy deployments, including the individual waypoint plans for each decoy deployed. Further, the CTPP planner can, within the constraints imposed by the CaP, search for improved ordering of POIs in a team's plan in order to reduce total team mission time and/or risk.

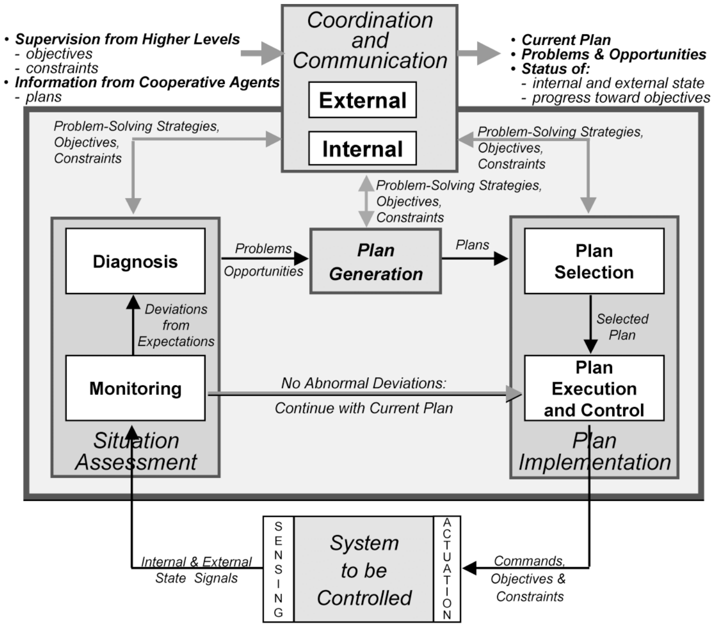


Fig. 5 Closed-loop, event-based planning and control occurs at each of the levels: TCTA, TDT, CTPP.

The routing algorithms employ a time-to-plan-sensitive variant of the A* algorithm, referred to as T*. This variant allows us to create suboptimal routes while using less computational time than a complete (optimal) A* generated plan. The variant is parameterized to control the tradeoff between computational time and solution suboptimality. Another parameter in the routing algorithm allows us to trade off total route time and risk. That is, if mission timeliness is critical, the algorithm will generate higher risk routes that take less time to execute than the optimal (or suboptimal T*). Finally, the risk minimization takes into account aspect-dependent stealth and both known (detected) and expected (but as yet undetected) IADS threats.

During a full replanning cycle (that is, during execution of a complete replan from TCTA through CTPP), CTPP sends the details of the plan for each team option, including its value and cost, to the TCTA. The TCTA uses integer programming to select the set of team options that maximizes overall campaign value while ensuring that resources (platforms) are not double booked at any time and that POIs are not assigned to more than one team. CTPP then replans the details of the team's mission.

VI. Event-Driven Replanning and Control

As described in Fig. 5 each node of the hierarchy includes (in addition to Plan Generation) Plan Monitoring, Diagnosis, and Plan Execution. Based on the sensed, changing battlespace state or changes in commander's intent, replanning can occur at any node and at any level.

Thus, the architecture performs four functions continually at every level:

1) Planning: generate plans to accomplish objectives defined by higher level/commander.
2) Execution: send commands to next level for implementation/more detailed planning.
3) Monitoring: examine world state for evidence of deviation from expected plan state.
4) Diagnosis: determine whether differences detected by monitoring represent problems or opportunities deserving of new plans; if so, direct planner function to replan.

As described in Ref. 11, a design that incorporates all of these functions within every node allows for centralized planning and distributed plan execution and plan perturbation. Thus, when appropriate, lower level nodes (e.g., CTPP) can locally adapt their plans to the changing battlespace situation and only rely on higher-level nodes (e.g., TCTA) to redefine the problems to be solved by the lower level nodes when necessary.

VII. Implementation Architecture

Figure 6 gives an overview of the overall system implementation architecture including:

a) The automated planning and execution components in the set of Command and Control (C2) nodes comprising the *MICA Controller*.

b) The Mixed-Initiative HSI that provides mechanisms for the operator to interact with those C2 nodes, including providing commander's intent, planning guidance, and mission execution situation awareness.

c) The OEP (Open Experiment Platform) simulation environment developed by Boeing for the DARPA MICA program

d) The software components (time-stepper and MetaController) that act as an interface between the C2 nodes and HSI and the OEP.

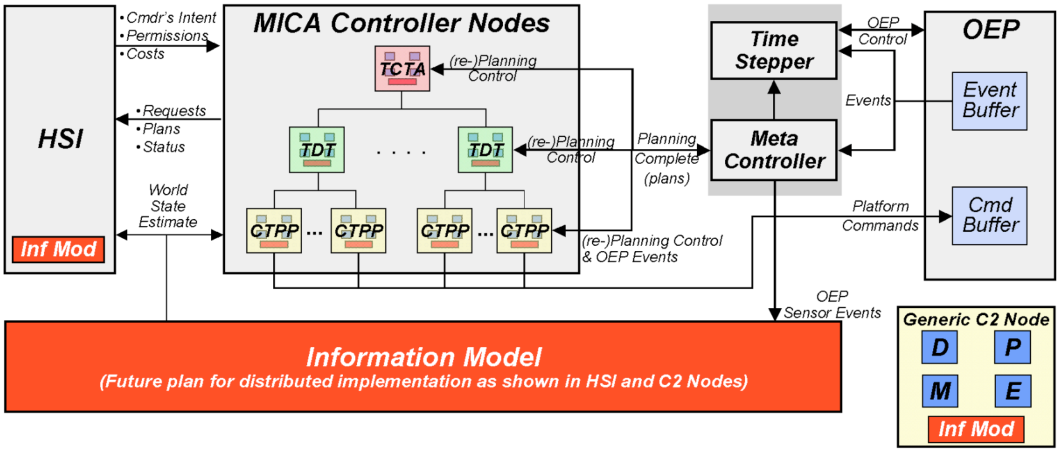


Fig. 6 MICA implementation architecture.

In the figure, *commander's intent is* provided as input from the HSI to the controller nodes and reflects on a per-target basis, as described earlier, the importance of time, geography, target class and risk. The *Information Model* provides situation awareness, as described earlier, through two classes of information states: *grid-based* probability distribution information about the number and types of undetected objects in specific geographic regions; and *track-based* probabilistic information about individual entities in the battlespace.

VIII. Operator Interaction with Automation

A. HSI and the MICA Hierarchy

An effective HSI design (including software architecture, data requirements, and interfaces) enables human interaction with a hierarchical control system to jointly solve and execute complex air operations planning and scheduling problems. This section describes the formulation and design approach and some of the features of the MICA HSI.

We distinguish two types of HSI operators: TCTA operators and TDT/CTPP operators. TCTA operators are more experienced, command-level personnel who provide high-level commander's intent to the top-level control node and make global decisions that affect all vehicles and teams in the battlespace. TDT/CTPP operators are more junior personnel than TCTA operators. Their primary mission is the supervision and management of UAV teams. They make local decisions relative to their vehicles/teams and objectives such as reconfiguring teams, altering flight paths, and participating in weapon and sensor employment events.

B. HSI Design

Figure 7 is a software-centric view of the MICA architecture depicting individual controller agents at each level of the hierarchy, i.e., TCTA, TDT, or CTPP, each of which communicates via a common "blackboard" to send and receive data. The HSI communicates with the controller via Java Messaging Service (JMS).

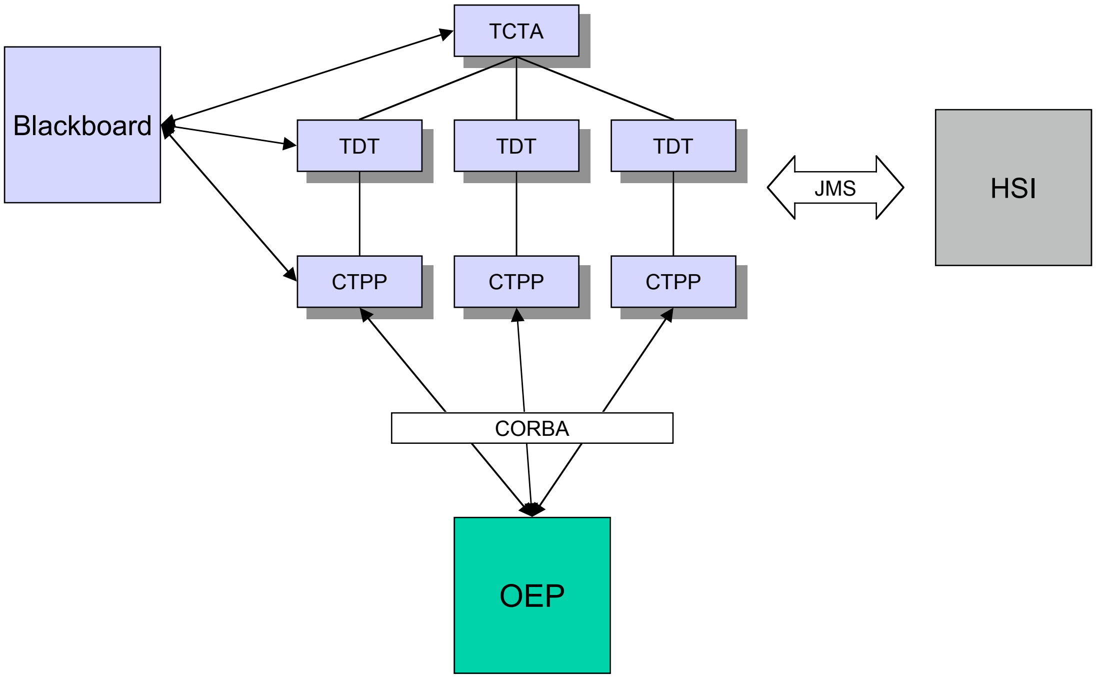


Fig. 7 Overall software integration architecture.

The system is designed to allow an HSI to operate at any level of the controller hierarchy (i.e., TCTA, TDT, or CPP) as well as to facilitate a distributed, multiprocessor implementation. JMS is the mechanism by which the controller provides the HSI with Information Model state updates and controller plans and by which the HSI sends command structures to the controller. The data models contain data appropriate for the various levels in the hierarchy (e.g., commander's intent model, a Red Information Model, Blue Information Model, plan status model, etc., for the TCTA level).

Figure 8 shows a high-level view of the HSI software architecture reflecting its Model-View-Controller (MVC) basis. The HSI is composed of two main views: 2D and 3D, each with its own controller (here we use "controller" in the MVC sense), which is integrated into the view itself. The controller is responsible for responding to operator input and, for instance, changing the display of the view or interacting with the view or data models on behalf of the operator. Additional informational views exist comprising many user interface (UI) components, including tables, trees, Gantt chart components, etc. as listed in Table 1. The data models and views communicate via the standard Java Event Model, wherein either the view or individual UI components making up the view register as listeners and respond to events thrown by data models.

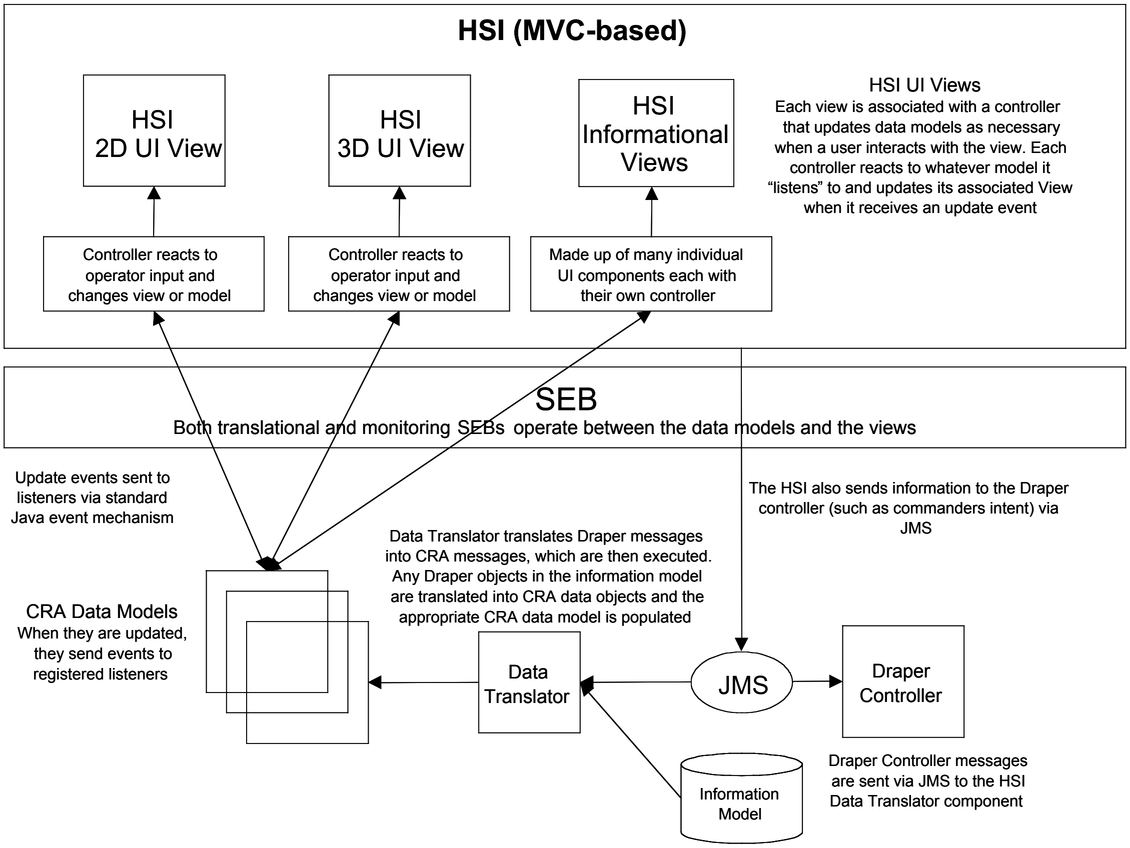


Fig. 8 HIS software architecture.

C. HSI Functional Design

The HSI allows an operator to display information for any level of the controller hierarchy; and enables the operator to achieve Situation Awareness (SA) and to intervene either proactively or when requested by the automation. The Information Model keeps track of data about Blue aircraft, Red tracks, probability mass functions for each grid cell on the map, weapon and sensor types, etc. This Information Model is updated periodically in response to messages from the MICA controller. The data translator component translates data from this Information Model into data objects. The data models are then populated with these data objects and throw events to registered components and views so that they may update themselves with new information.

Various *Semantic-to-Engineering Bridge* (SEB) algorithms operate between the data models and the views. These algorithms filter data to translate raw data into ideas and concepts more readily understandable by the operator or from ideas into raw data to send to the controller and monitor data for operator alerting. Figures 9 and 10 illustrate how the SEB concept fits in with the HSI system.

In some cases, the Semantic Engineering Bridge (SEB) responds to the events thrown by a data model. Currently, an SEB can serve one of two purposes; it either translates data or monitors data. If the SEB is a translator, it translates controller-centric data from the system data models into operator-centric SA level 2 and 3 information or vice versa. This action occurs when the operator opens a UI component that utilizes a translational SEB. Next, the UI component invokes the translational SEB, which retrieves appropriate data from the system data models and transforms the data (e.g., multiplying value by commander's intent to get weighted value), and returns a new value to the UI component for display.

As shown in Fig. 9, a *translational* SEB can also translate data in the opposite direction. An operator may use a UI component to enter data in a semantic way rather than just raw data. For instance, an objective to intent translational SEB may take a given objective (e.g., perform a SEAD mission) and convert it into the controller input data necessary to achieve that objective (e.g., commander's intent matrices). This allows the operator to express his intent in a natural way independent of the underlying controller implementation.

Table 1 Example UI components

UI Component	Purpose
TCTA Plan Summary	Tree view of the plan at the TCTA level. Gives detail about plan duration, the number of teams, and targets struck and viewed.
TCTA Team Detail	Tree view that gives details about each team being sent out, including number of aircraft in team, targets struck, and viewed by that team.
Blue Resources	Tree view that shows how many vehicles, weapons, and sensors are available to the operator.
Estimated Red Threat	Tree view that displays the identified and unidentified tracks in the COP.
TCTA Objective/ROEs	Text panels that show the objectives and ROEs for the scenario.
AIM/KIM/DIM/AVM Entry	Table entry, slider bar entry, and graph view combination that allows commander to enter his intent.
Per Track Importance Entry	Slider bar entry that allows commander to specify, on a track-by-track basis, importance multiplier against commander's intent.
TDT Plan	Tree view that displays the teams at the middle level and aspects of those teams.
Team Synchronization Matrix	Gantt chart view that shows each team and the individual vehicles in that team as well as the various activities, over time, each vehicle is performing.
Target Synchronization Matrix	Gantt chart view that shows all targets in the scenario and what actions are being taken against them, including jamming, striking, and sensing.
Team Panel	Display which gives information about team makeup, team phase, team objective and whether or not a team is performing weapons release.
Vehicle View Panel	Panel that gives information for the individual vehicles in a team.
Weapons View Panel	Panel that gives detail about the weapons load for individual vehicles of a team.

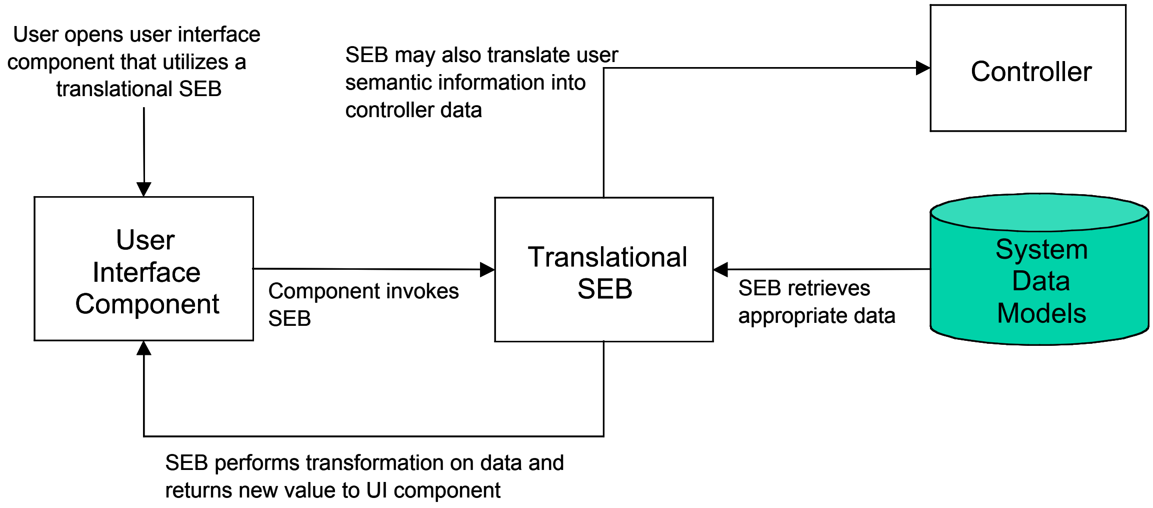


Fig. 9 How a translational SEB fits into the HSI system.

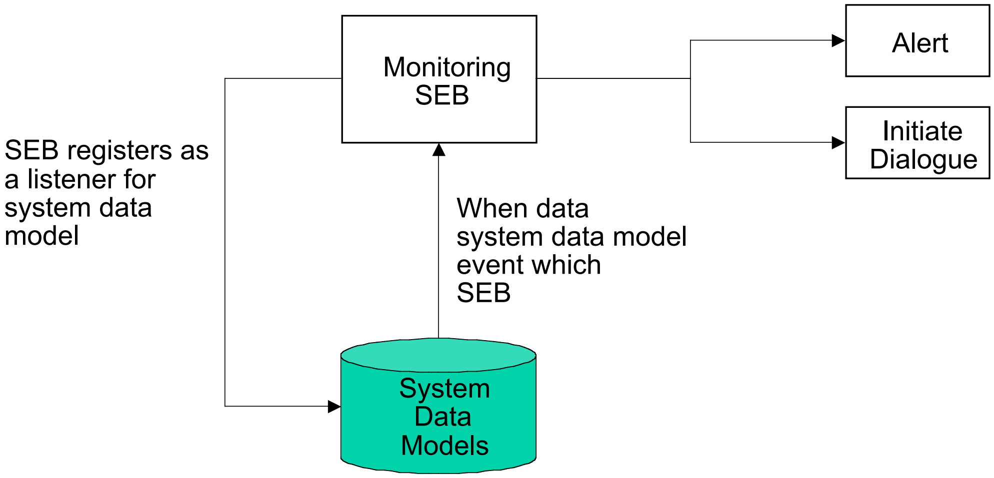


Fig. 10 How a monitoring SEB fits into the HSI system.

Figure 10 shows the architecture for a *monitoring* SEB. This algorithm constantly analyzes the data presented in the system data models and uses techniques such as Bayesian Networks, fuzzy logic, and expert systems to detect tactically relevant events. Depending on the impact of these events, the SEB may alert the operator to the change or initiate a dialogue with the operator to determine a new course of action. This initiation of dialogue can come in many flavors from a simple message window to a more complicated presentation of "advice" from the SEB as to how to approach the situation.

D. The HSI as an Intelligent Interface

This section describes in detail several intelligent agents and SEBs that were created for the MICA HSI.

1) The Semantic Engineering Bridge (SEB)

An SEB can be described as a general relationship between a semantic concept (e.g., importance of a target or target class as a function of geographic region) and low-level data. To date, we have implemented this relationship to process PMF grid cell data in the form of

$$G \equiv F(\text{cell}, \text{PMF}, \underline{\Theta}) \tag{9}$$

where G is the semantic concept of interest, F is a mathematical description of G, and $\underline{\Theta}$ is a set of weights or parameters that describe G. For example, if G is meant to represent the importance of region_{ij}, setting $F = \Sigma\, \text{PMF}_{ij}\, \underline{\Theta}_j$, where $\underline{\Theta}_j$ is either the AIM or KIM matrix, generates the importance of the region relative to the commander's intent.

E. TCTA SEB

Several SEBs have been developed and implemented: two SEBs to help a TCTA operator obtain situation awareness of the battlespace and construct a plan; and two SEBs to help a TDT/CTPP operator understand and monitor each team's high-level objective.

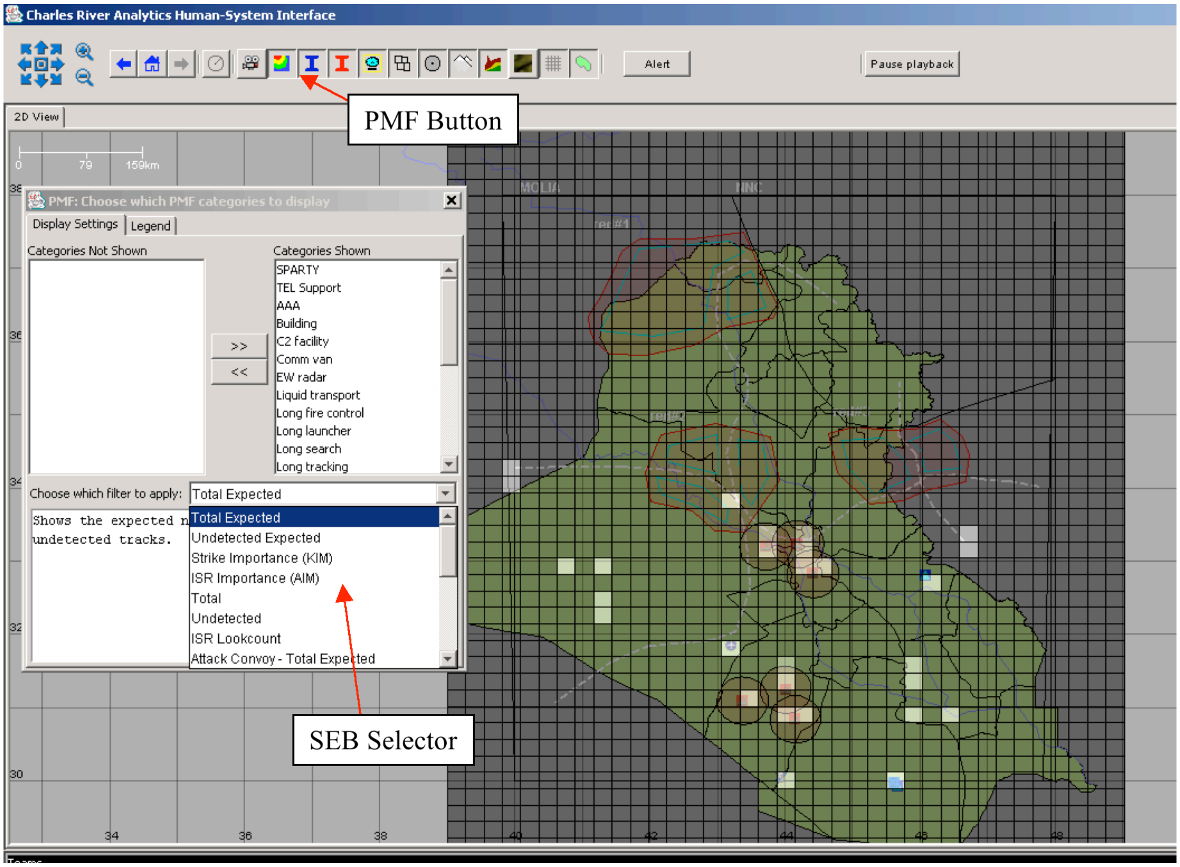
For the TCTA operator, we identified information requirements related to geographic awareness of Red activity, such as where are likely places that Red forces may attack Blue forces and launch SSMs. Using the Red IPB, we developed a belief network topology that models intuitive Courses of Actions (COAs). The intent was not to provide exact models that would be applicable in all domains, but rather to demonstrate and investigate how such models could aid the TCTA operator as well as provide a path for integration with other Department of Defense (DoD) efforts (e.g., Effects-Based Operations). A graphical depiction of the relationships contained in and implied by the Red IPB was developed in an easy-to-understand network format. For example they show the relationship inherent in the fact that attack convoys tend to have certain vehicle types and Red forces tend to attack Blue forces using attack convoys that travel on roads and use civilian (White) objects for deception. Similarly, retreat convoys contain certain vehicle types and also tend to travel on roads with White objects moving a direction that exits the battlespace. Also, Red forces tend to launch SSMs using TELs, which are near TEL support facilities and roads.

In an effort to generalize the applicability of this model, we used binary states of *tactically significant* and *tactically insignificant* for the other nodes. For simplicity, we initially defined the tactical significance (TS) to be

$$TS_i = X_i/N_i \tag{10}$$

where X is the number of tracks of type I and N is a normalizing factor. This modeling choice provides the TCTA operator the flexibility to change or define tactical significance for a given scenario.

We embedded this SEB in the HSI. For each X_i, we used a value equal to the expected total number in the PMF values in the Information Model. This SEB can be viewed as a PMF filter that combines tactical knowledge, i.e., it combines the IPB concerning Red behavior with real-time sensor data. As an example of the application of this SEB, Fig. 11 shows the locations where attack convoys are most likely. Higher probability locations are shown in lighter colors. For this scenario, no vehicles have taken off and gathered sensor data, so the highest probability is only 7%.


Charles River Analytics Human-System Interface

PMF: Choose which PMF categories to display

Categories Not Shown

Categories Shown
- SPARTY
- TEL Support
- AAA
- Building
- C2 facility
- Comm van
- EW radar
- Liquid transport
- Long fire control
- Long launcher
- Long search
- Long tracking

Choose which filter to apply: Total Expected

Shows the expected n
undetected tracks.

- Total Expected
- Undetected Expected
- Strike Importance (KIM)
- ISR Importance (AIM)
- Total
- Undetected
- ISR Lookcount
- Attack Convoy - Total Expected

Fig. 11 Implementation of Red COA SEB.

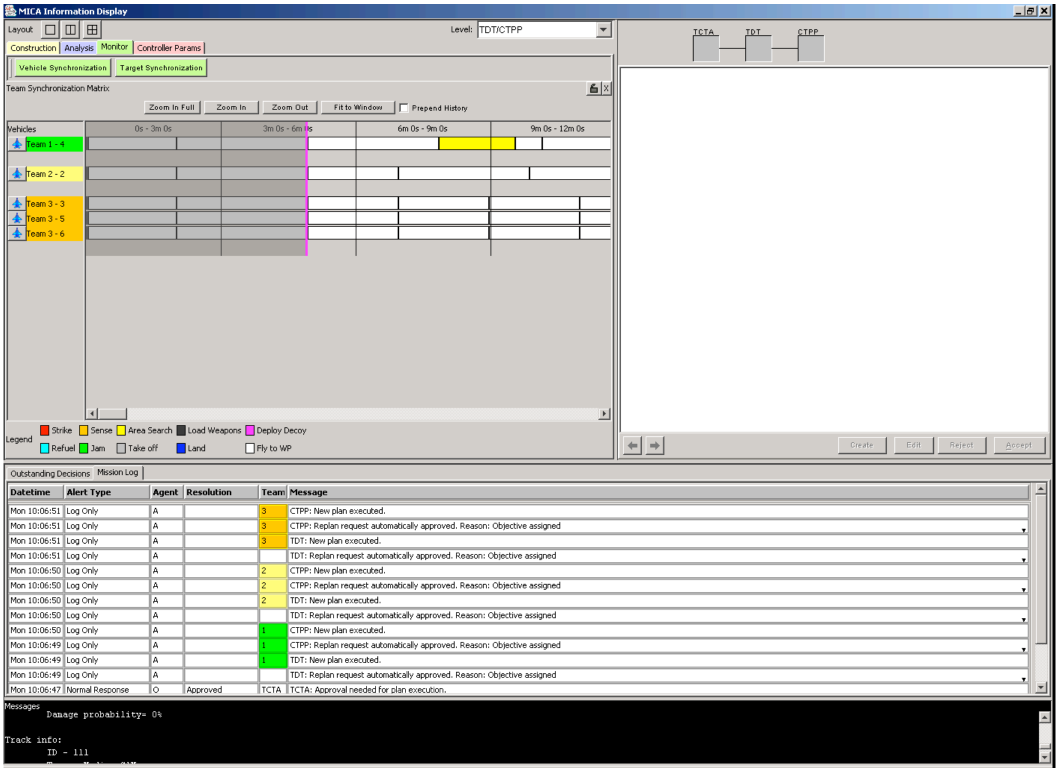
In addition to using belief networks to filter PMF data, we also developed algorithms that combine commander's intent with PMF data. These algorithms help the operator understand regional importance with respect to established criterion. Table 2 lists the implemented SEBs. First, threat concentration importance is defined as the summation of the KIM times the total expected number PMF to indicate that regions are more important if they likely contain more high-value targets that the commander wants destroyed. Second, threat uncertainty is defined as the summation of the AIM times the expected number of undetected PMF indicating regional importance based on the likelihood of finding more high-priority unknown targets.

Table 2 Linear SEB filters

G	W	PMF
Threat Concentration	KIM	Total Expected Number
Threat Uncertainty	AIM	Expected Number of Undetected Tracks

F. TDT/CTPP SEB

For a TDT/CTPP operator, a team objective and a team phase SEB were implemented. These mechanisms help the operator comprehend vehicle-level flight plans and actions. They were added as a result of Human-in-the-Loop (HiTL) experiments, where the operators had difficulty determining how each team supported the commander's intent and determining what each team was doing at any given moment. At that time, the primary means for an operator to determine a team's objective was to manually inspect the team synchronization matrix (see Fig. 12). This is a labor-intensive process whereby the operator must identify the target types and location for each team and infer from those the team's objective. For small numbers of teams that have relatively few vehicles, this process is tractable, but it is not for large numbers of teams.



Datetime	Alert Type	Agent	Resolution	Team	Message
Mon 10:06:51	Log Only	A		3	CTPP: New plan executed.
Mon 10:06:51	Log Only	A		3	CTPP: Replan request automatically approved. Reason: Objective assigned
Mon 10:06:51	Log Only	A		3	TDT: New plan executed.
Mon 10:06:51	Log Only	A			TDT: Replan request automatically approved. Reason: Objective assigned
Mon 10:06:50	Log Only	A		2	CTPP: New plan executed.
Mon 10:06:50	Log Only	A		2	CTPP: Replan request automatically approved. Reason: Objective assigned
Mon 10:06:50	Log Only	A		2	TDT: New plan executed.
Mon 10:06:50	Log Only	A			TDT: Replan request automatically approved. Reason: Objective assigned
Mon 10:06:50	Log Only	A		1	CTPP: New plan executed.
Mon 10:06:49	Log Only	A		1	CTPP: Replan request automatically approved. Reason: Objective assigned
Mon 10:06:49	Log Only	A		1	TDT: New plan executed.
Mon 10:06:49	Log Only	A			TDT: Replan request automatically approved. Reason: Objective assigned
Mon 10:06:47	Normal Response	O	Approved	TCTA	TCTA: Approval needed for plan execution.

```
        Damage probability= 0%

Track info:
        ID - 111
```

Fig. 12 Vehicle synchronization matrix for determining team objective and mission phase.

To implement a team objective SEB, an objective-to-target mapping (Table 3) was constructed. This approach assumes that objectives have characteristics, one of which is a relative target value. For example, destroying SAM sites is more important than destroying supply trucks for neutralizing IADS. Currently, this approach does not include geographic information.

We undertook a similar effort to automate determining the currently executing mission phase for each team, allowing the operator to know what each team is doing at any point in time during its mission. As before, the operator could manually accomplish this by looking at the current vehicle actions for each team in the team synchronization matrix and then inferring from that the mission phase. To reduce the cognitive effort, we developed a prototype automated approach employing a rule-based system wherein the team phase is inferred based on the highest priority action of any one vehicle. For example, if a vehicle is striking a target while another is sensing a track, the rule for striking a target is used.

Figure 13 shows a screenshot of the team objective and team mission phase SEBs. In this case, Team 1's mission is SEAD and the current phase is either ingressing into or egressing from the target area. Similarly, Team 3's mission is reconnaissance. Note that this SEB could be integrated with the alerting system to notify the operator of potential targeting errors.

Table 3 Target-to-objective prioritization matrix

Target Type	Neutralize IADS	Protect Blue Operating Base	Prevent SSM Launches
IADS Command Center	High	Low	Low
EW Radar	Very High	Low	Medium
Long-Range SAM Launcher	High	Low	Medium
Long-Range SAM Fire Control Radar	High	Low	Medium
Long-Range SAM Tracking Radar	High	Low	Medium
Long-Range SAM Search Radar	High	Low	Medium
Medium-Range SAM	Slightly High	Low	Medium
Short-Range SAM	Slightly High	Low	Medium
TEL	Low	Very High	Very High
TEL Support Facilities	Low	Very High	Very High
Tank	Very Low	High	Very Low
SPARTY	Very Low	High	Very Low
Armored Personnel Carrier	Very Low	High	Very Low
Supply Trucks	Low	Medium	Low
Liquid Transport Truck	Low	Medium	Low
Communication Van	Low	High	Very Low
Mobile HQ	Low	High	Low
Anti-Aircraft Artillery (AAA)	Medium	Low	Low

2) *Mixed-Initiative Agent*

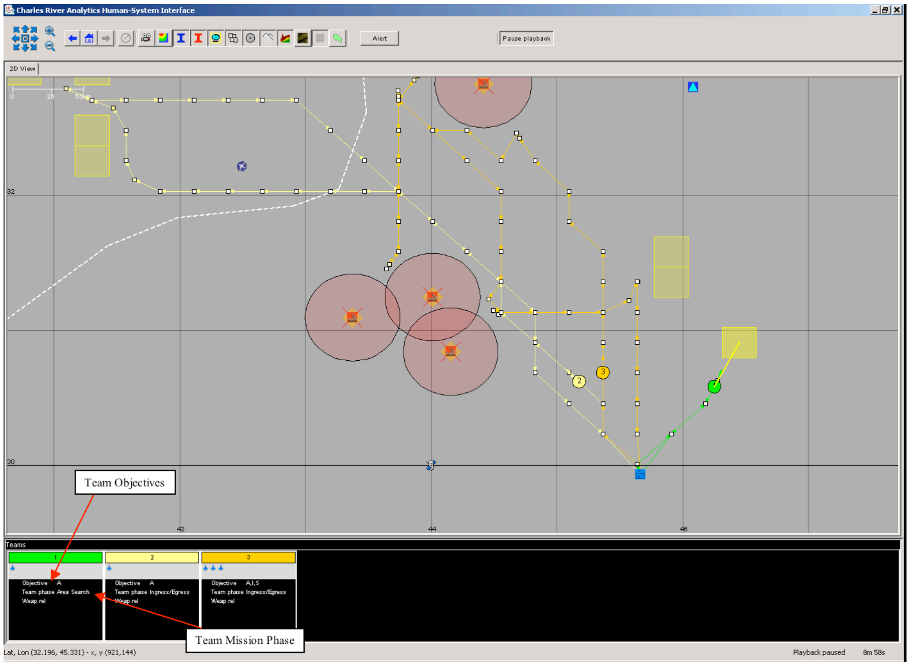
The mixed-initiative agent is an agent designed to act on behalf of the operator in response to requests generated by the controller. Currently, the operator has the ability to configure this agent to behave as a function of the nature of the incoming request. There are three supported behaviors:

a) Shared by Consent: A request must be accepted by the operator in order for the controller to proceed. In this case, the agent forwards the request to the operator and waits for a response. If the operator does not respond within a certain timeout period, the agent will indicate a rejection to the controller. This is done so that, if the operator becomes unavailable, the controller can still proceed with the current course of action.

b) Shared by Exception: A request must be rejected by the operator for the controller to not proceed. In this case, the agent forwards the request to the operator and waits for a response. If the operator does not respond within a certain timeout period, the agent will indicate acceptance of the request to the controller. This is done so that, if the operator becomes overloaded, the controller can proceed with requested actions without explicit operator acceptance.

c) Fully Automatic: A request is automatically accepted by the agent. The agent does not forward the request to the operator at all, but rather acting on behalf of the operator, accepts the controller's request.

These behaviors can be set by the operator for the various types of requests initiated by the controller (e.g., replanning due to a track moving, replanning due to a new track, etc.). Figure 14 shows an example of the operator-interface component used to modify these behaviors. Each level of the hierarchy can be set independently as well.



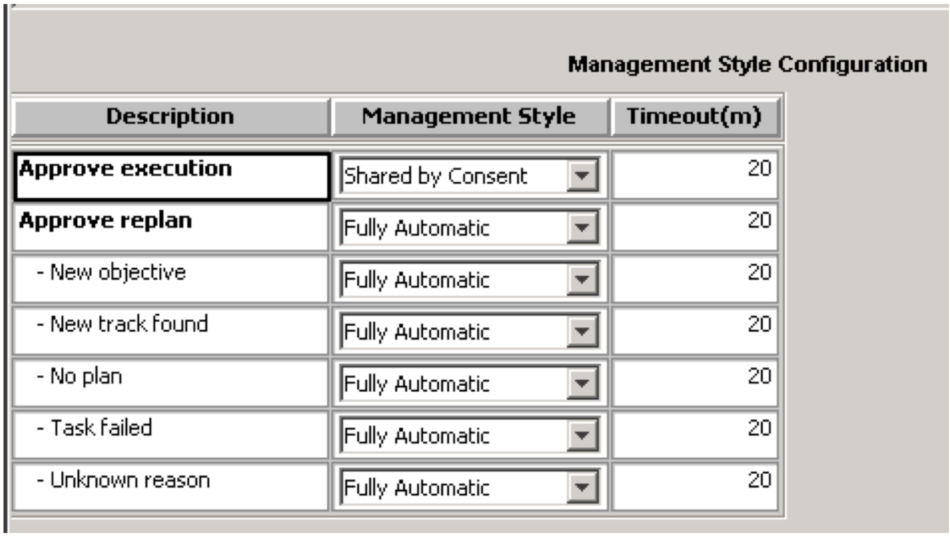
1
Objective A
Team phase Area Search
Weap rel

2
Objective A
Team phase Ingress/Egress
Weap rel

3
Objective A,I,S
Team phase Ingress/Egress
Weap rel

Lat, Lon (32.196, 45.331) - x, y (921,144)

Fig. 13 Screenshot of team objective and team phase SEB.



Management Style Configuration

Description	Management Style	Timeout(m)
Approve execution	Shared by Consent	20
Approve replan	Fully Automatic	20
- New objective	Fully Automatic	20
- New track found	Fully Automatic	20
- No plan	Fully Automatic	20
- Task failed	Fully Automatic	20
- Unknown reason	Fully Automatic	20

Fig. 14 An example of the UI component used to modify the mixed-initiative agent behavior based on controller request.

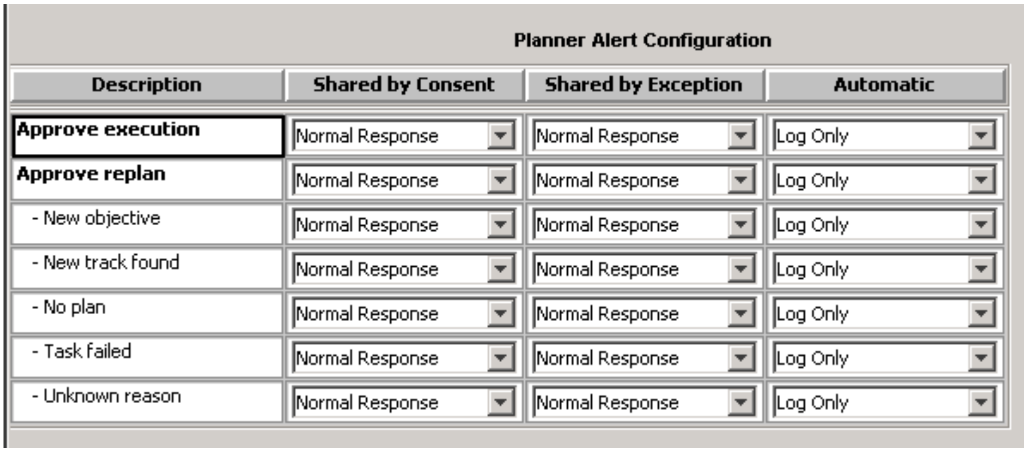
Currently, the agent is not implemented to take action based on its own interpretation of the current situation. Future work would include modifying this agent to be more reactive, and more importantly, proactive in making decisions on when to involve the human operator in the decision-making for controller requests. For example, if a track has only moved 100 km, there should be little reason to involve the operator in deciding if it is okay for the controller to replan due to this world state change. The agent should be capable of determining the level of interaction with the operator based on the request and the current world state and, potentially, acting on behalf of the operator outside of the operator defined interaction style for that situation.

3) Intelligent Alerting System Agent

One example of the intelligent agents running in the HSI is the intelligent alerting system agent. The alerting system is based on Ref. 10, which describes several response styles for alerts including:

a) Emergency Response: alert requires operator attention immediately.
b) Quick Response: alert requires that operator must be made aware of the alert, but response is not needed immediately.
c) Normal Response: Alert requires the operator be made aware of the alert, but with minimal interruption (e.g., logged to a table that the operator can acknowledge later).
d) Information Only: Alert requires no operator interruption, and alert is shown to the operator for a certain period of time and then logged to a table in the interface. Requires no operator acknowledgement.
e) Log Only: Alert requires no operator interruption and is just logged to a table in the interface.

The operator can set the response style for each alert type available using a simple operator-interface component (shown in Fig. 15). Based on these settings, when an alert is generated by the system, either through a controller event or a system monitor event, the alerting system agent alerts the operator accordingly. For example, if the operator has chosen Emergency Response for a weapons release, then when a weapons release alert is generated, the system will flash the operator alert button, display a dialog indicating a weapons release alert has been raised, and issue an auditory alert signal to gain the attention of the operator.



Planner Alert Configuration

Description	Shared by Consent	Shared by Exception	Automatic
Approve execution	Normal Response	Normal Response	Log Only
Approve replan	Normal Response	Normal Response	Log Only
- New objective	Normal Response	Normal Response	Log Only
- New track found	Normal Response	Normal Response	Log Only
- No plan	Normal Response	Normal Response	Log Only
- Task failed	Normal Response	Normal Response	Log Only
- Unknown reason	Normal Response	Normal Response	Log Only

Fig. 15 Example UI component for setting the response style for various alerts.

As of the current implementation, the alerting agent's interaction with the operator is dictated solely by the operator's chosen style for the raised alert. Future work for this agent would include introducing more mixed-initiative aspects into the agent such that the agent would decide how to best interact with the operator based on the raised alert, current operator workload, the degree of change in world state, and other factors. These decisions made by the agent would fall outside the realm of strict operator style choice and would present a much more intelligent alerting system.

4) Red Information Adaptive Display Agent

A second intelligent agent incorporated into the HSI is the red information adaptive display agent. Using an expert-system-based set of rules, when activated, this agent adjusts the display of the Red information layer such that the operator is made aware of only the Red tracks deemed important by the agent. This agent is built using Charles River's in-house agent architecture called SAMPLE.

Using the ability in the Red information layer to display a track as a centroid or an icon, the adaptive display agent determines which tracks are important, on a team by team basis, and displays them as icons while displaying unimportant tracks as centroids. This effectively declutters the screen and focuses the operator's attention on those tracks that are of the most concern to a particular team or set of teams. The agent follows certain rules when determining if a track should be displayed as an icon or a centroid. The rules are as follows:

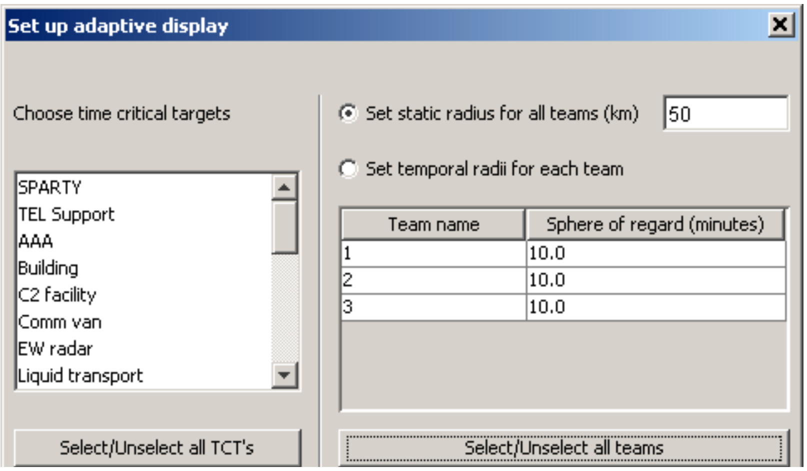
a) If the track is a time-critical target (TCT), show as an icon.

b) Otherwise, if the track is a sense/strike/jam target and it is within the specified radius around a team, show as an icon.

c) Otherwise, if the perceived track type is designated as friendly (Blue) or neutral (White), and it is within the specified radius around a team, show as an icon.

d) Otherwise, show the track as a centroid.

The agent is configurable by the operator using the interface shown in Fig. 16.



● Set static radius for all teams (km) 50

○ Set temporal radii for each team

Team name	Sphere of regard (minutes)
1	10.0
2	10.0
3	10.0

Fig. 16 UI component used to configure the adaptive display agent.

As can be seen, the agent can be configured in several ways. First, the designated TCT types can be selected on the left-hand section of the component. Second, the operator can specify a radius for all teams (in kilometers) or, on a team-by-team basis, specify a sphere of regard. This “sphere,” specified in minutes, combined with the current maximum speed of all vehicles in a team yields a dynamic radius for each team and is representative of important tracks reachable within the specified time frame. Finally, the operator can select which teams are currently being monitored using the agent.

IX. Sample Results

Here we present some representative results of our MICA controller that was run in the completely autonomous mode (no operator interaction) against a complex scenario simulated in the OEP.

A. Scenario

The scenario employed for the results presented here is a scud-hunting problem in which Red has a strong Air-Defense presence in the operational area. A significant number of long- and medium-range SAM systems “cover” the high-value targets, e.g., the TELs and TEL_support entities. Note that the entities in play in this scenario are representative of the types of assets that may be available to Blue and Red, but are not intended to model any specific, existing platforms.

1) Platforms

Table 4 summarizes the principle Blue and Red assets. The Blue assets are classes of notional future unmanned air vehicles. The large and small sensor Blue platforms are sensor-only platforms for Intelligence, Surveillance, and Reconnaissance (ISR) equipped with the following sensing capabilities:

a) An electro-optical (EO) sensor for target identification and Battle Damage Assessment (BDA).

b) An Electronic Surveillance Measures (ESM) sensor for locating and identifying enemy Radio frequency (RF) emissions.

c) A Multi-Mode Radar (MMR) that operates in one of several modes, including Synthetic Aperture Radar (SAR) strip map, spot image, and Ground Moving Target Indicator (GMTI).

The differences between the large and small platform are: increased sensor range for the large platform and decreased radar cross-section for the small platform.

Table 4 The "order of battle" for Red/Blue in the strong air-defense system scenario

Blue Order of Battle		Red Order of Battle	
large_sensor	6	TEL_support	2
large_weapon	6	TEL	4
small_sensor	8	c2_facility	2
small_weapon	8	ew_radar_site	6
small_combo	8	red_esm	6
		long_sam_launcher	56
		long_sam_search_radar	14
		long_sam_tracking_radar	14
		medium_sam_site	31

The Blue large and small weapon platforms are strike-only platforms whose sole sensor is an Electronic Surveillance Measures (ESM) sensor for detecting any Red air defense radars that may be tracking them. They are capable of carrying a variety of weapon load configurations comprising coordinate-seeking GPS guided weapons and Laser Guided weapons, with the former having longer standoff range but requiring more accurate target location information. The differences between the large and small weapon platforms are: increased carrying capacity for the large platform and reduced radar cross-section for the small platform. The small combo is a multifunction platform with weapons, ESM, and EO sensors. Note that all platforms are equipped with ESM warning sensors that are capable of detecting Surface-to-Air Missile (SAM) tracking radar as well as with Electronic Countermeasure (ECM) to support jamming.

The Red air defense comprises early warning radars (ew_radar_site), fixed ground-based ESM to detect Blue jammers, Long Range SAM systems and Medium Range SAM systems, where a SAM system includes launchers, search and tracking radars. The small and large red circles in Fig. 17 depict the effective range of the medium and long SAM systems, respectively.

Figure 17 shows the "truth" laydown of the Red order of battle. Note that the TELs in the southwest of the battlespace, which are the primary targets in this scenario, are protected by multiple long SAM systems.

Initial Intelligence Preparation of the Battlefield (IPB)

The "truth" laydown of Red is not initially known to Blue. Table 5 summarizes the poor level of initial knowledge about the specific location of the majority of Red's assets. More than 75% of Red's assets are only approximately localized as being somewhere within a 25-km x 25-km cell.

Blue's knowledge of Red's assets from IPB at time zero is illustrated in Figs. 18 and 19. Figure 18 shows the laydown of Red assets whose position is known initially by Blue to within 1 km. Figure 19 shows highlighted grid cells where poorly known Red assets are expected to be concentrated. Note that the location of none of the TELs is known initially and that some of the long SAM systems protecting the TELs are also poorly known at time zero.

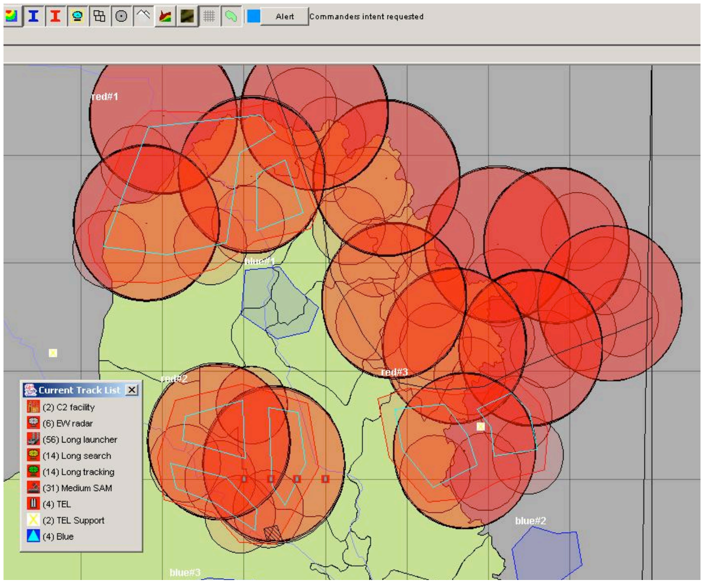


Fig. 17 The true laydown of Red's entities.

Table 5 Initial knowledge of Red order of battle

Type	% Known with location error < 1 km	Location CEP (m)	Type/Damage Certainty
TEL_support	0%		
TEL	0%		
c2_facility	100%	10	95%
ew_radar_site	100%	10	95%
red_esm	0%		
long_sam_launcher	25%	100	70%
long_sam_search_radar	25%	100	90%
long_sam_tracking_radar	25%	100	90%
medium_sam_site	20%	1000	70%

2) *Initial Missions*

Figure 20 illustrates the initial missions developed for 6 teams of UAVs. Note that although there are a few strikes planned (red "Xs" in Figure 20), most of this initial activity is ISR (indicated by yellow shaded rectangles) for either locating TELs using standoff sensing or for improving the location of the SAM systems prior to launching a strike against them. Since our routing algorithms are conservative in accounting for expected IADS threat (i.e., those with poorly known locations), improving our knowledge of their locations improves the ability of the teams to more closely approach targets with known locations.

3) *Results over 20 Hours of Campaign Time*

Figures 21, 22 and 23 summarize the results of planning and replanning over a 20 hours of campaign time. The targets located and destroyed summaries focus on long and medium SAMs and TELs as the SAMs are the principle impediment to accomplishing the mission objective of destroying the TELs. The values at time zero in the plot of Targets Located (Fig. 21), reflect the initial IPB information provided to the controller. Over the span of twenty hours, a large number of medium and long SAMs are located along with the four TELs of interest.

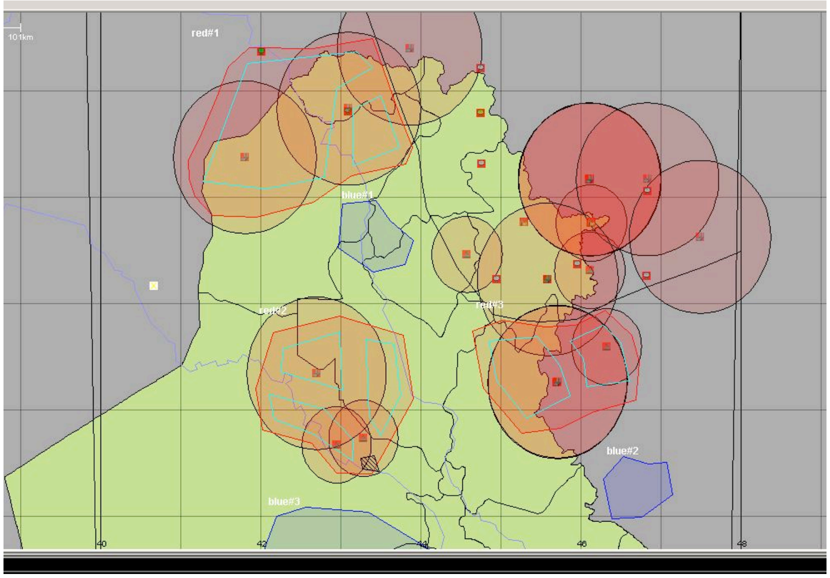
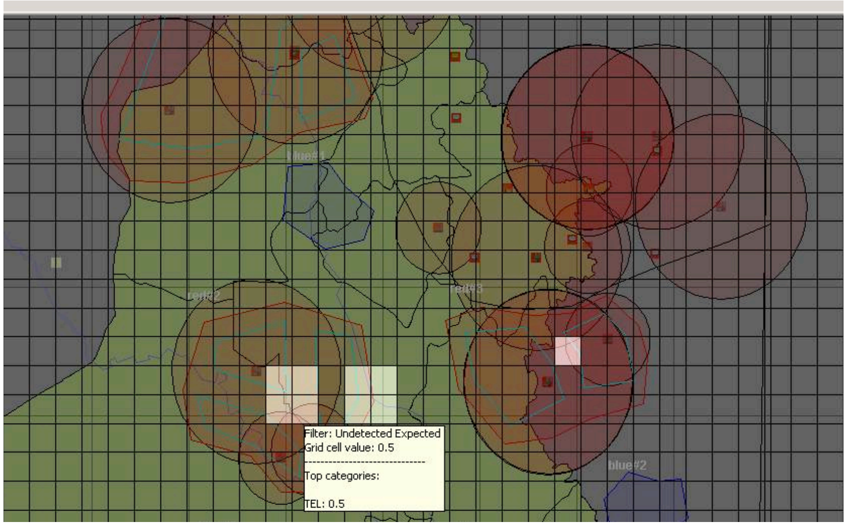


Fig. 18 Blue IPB of Red assets with known locations (uncertainty < 1 km).



Filter: Undetected Expected
Grid cell value: 0.5

Top categories:

TEL: 0.5

Fig. 19 Blue IPB highlighting regions of expected TELs.

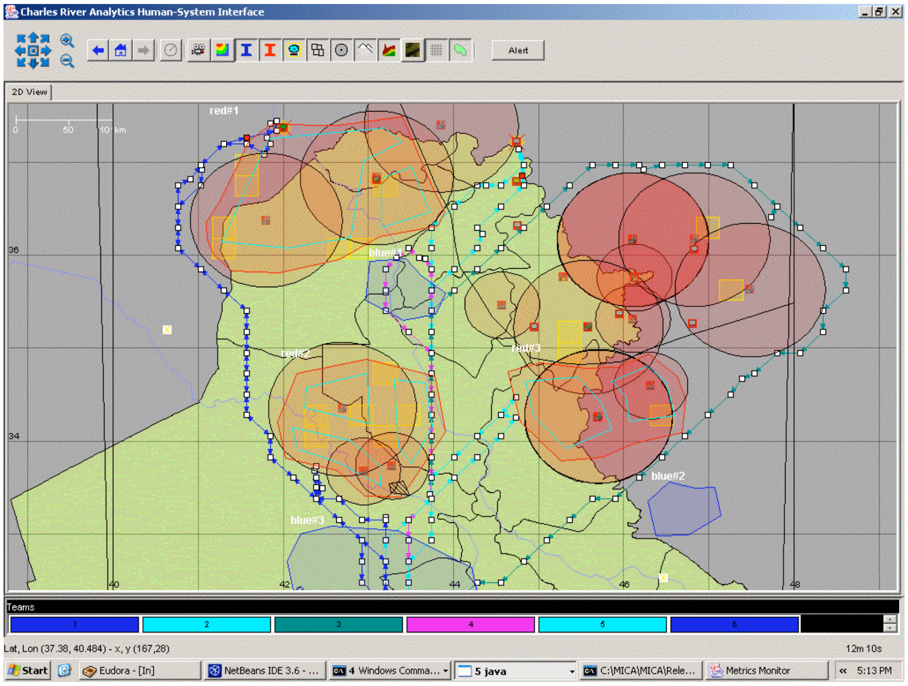


Fig. 20 Initial plan – six teams.

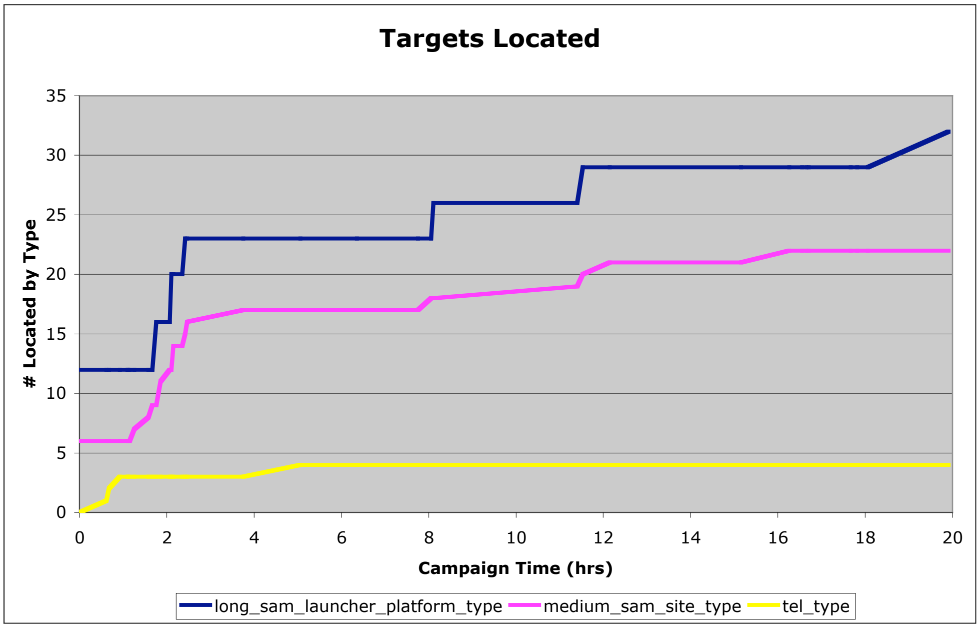


Fig. 21 Timeline of Red target discovery.

The Targets Destroyed summary in Fig. 22 highlights the fact that a variety of long and medium SAMs had to be destroyed before it was possible to safely attack the TELs. Because commander’s intent indicated additional value (though much lower than that for the TELs) in destroying the IADs, more SAMs than just those “covering” the TELs were destroyed. Note that by hour 17 of the campaign, all four TELs were located, identified and destroyed (the primary mission).

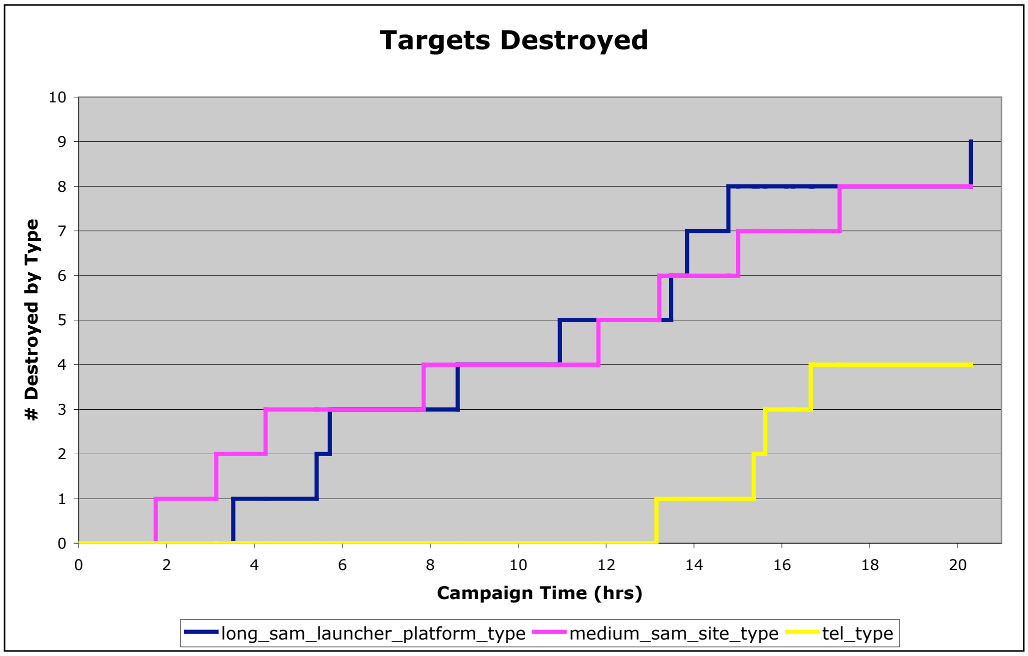


Fig. 22 Timeline of Red target destruction.

Figure 23 shows the Blue aircraft use over the 20 hours of campaign time. Note that the large sensor platform was the most heavily used throughout. This occurred because the large sensor is the only platform carrying sensors that have sufficient range to stand off outside of the Long SAM engagement range and that are capable of both accurately locating and identifying the Long SAMs. Furthermore, because the longest range of a standoff weapon is shorter than the Long SAM engagement range, standoff jamming is required to support the small weapon platform as it closes on the Long SAM to take a shot. The large sensor platform carries the most powerful ECM jammers that are required for this standoff jamming protection. Over the course of the 20 hours of campaign, only one Blue platform was lost – a small weapon platform.

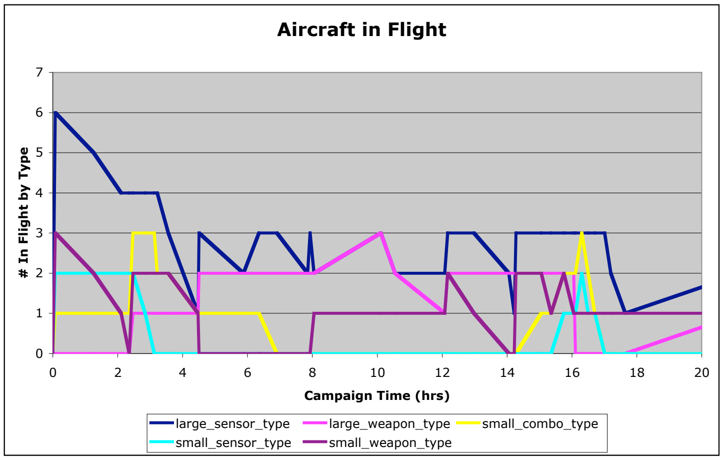


Fig. 23 Aircraft resource use over time.

X. Conclusions

We were able to successfully decompose and solve an extremely complex planning, scheduling and resource allocation problem for heterogeneous teams of unmanned air vehicles. To our knowledge, an air operations problem of this magnitude has not heretofore been addressed *as a whole*. Algorithms were developed that made the solution to this complex problem tractable. A hierarchical solution has been employed to enable centralized coordination with decentralized execution and plan perturbation. A probabilistic "Information Model" was enhanced to support the evaluation of candidate strike and ISR activities and the generation of low-risk team trajectories. A variety of novel human-system-interface mechanisms were designed and implemented to allow operators at both command and team levels to interact with plan generation, plan monitoring and plan execution. The integrated system has been tested and evaluated in the context of a challenging, independently developed simulation environment (the Boeing Open Experiment Platform).

While a great deal has been accomplished by the Draper team in the MICA program, there remains additional research and development required before such a system would be operationally useful. These include: a truly distributed implementation of mission planning, execution, and the information model, more sophisticated cooperative jamming techniques, explicit planning for cooperative ISR and strike of moving targets, use of decoys for threat discovery (our current implementation employs decoys only for team self protection in the face of pop up SAM threats), explicit planning for team redundancy and further refinement of the HSI and more semantic-to-engineering bridges (SEBs) to support broader operator participation in development of tactics to be employed within the plan generation algorithms.

Acknowledgments

The authors would like to thank the many members of the Draper MICA team who worked on the design, development and implementation of the algorithms, architecture, interfaces and software that formed the basis of this paper. They include Francis Carr, Carmen Falcone, Chris Guerra, Bill Hall, Michael Hutchins, Bill Kreamer, Lee Yang, Marc McConley and Lance Page of Draper Laboratory; and Chris Hopkins and Greg Zacharais of Charles River Analytics.

This material is based upon work supported by DARPA and AFRL/FI under Contract No. F30602-01-C-0201. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of DARPA or AFRL/IF.

This material is based upon work supported by DARPA and AFRL/FI under Contract No. F30602-01-C-0201. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of DARPA or AFRL/IF.

References

[1] Adams, M., Lepanto, J., and Hanson, M., "Mixed-Initiative Planning and Control Under Uncertainty Design Document," CSDL-R-2934, The Charles Stark Draper Laboratory, Jan. 2004.

[2] Adams, M., Hall, W., Hanson, M., Zacharias, G., McEneaney, W., and Fitzpatrick, B., "Mixed-Initiative Planning and Control Under Uncertainty," *Proceedings, First AIAA UAV Conference*, Portsmouth, VA, May 2002.

[3] Page, L., and Hall, W., "A State Estimator Supporting Command and Control of Multiple Aircraft Mission Types," *Proceedings, 2nd AIAA "Unmanned Unlimited" Systems, Technologies, and Operations-Aerospace, Land, and Sea Conference*, Sept. 2003.

[4] Adams, M., Deutsch, O., Hall, W., Hildebrant, R., Kreamer, W., McConley, M., and Vuong, H. F., "Closed-Loop Operation of Large-Scale Enterprises: Application of a Decomposition Approach," *Proceedings of Advances in Enterprise Control*, DARPA, July 2000.

[5] Tamura, H., and Yoshikawa, T. (eds.), *Large-Scale Systems Control and Decision-Making*, Marcel Dekker, New York, 1990.

[6] Deutsch, O., Adams, M., and Lepanto, J., "Closed-Loop, Hierarchical Control of Military Air Operations," Charles Stark Draper Laboratory Report CSDL-R-2913, May 2001.

[7] Armacost, A. P., "Composite Variable Formulations for Express Shipment Service Network Design," Ph.D. Dissertation, MIT Operations Research Center, June 2000.

[8] Clark, S., "Large-Scale Network Design Using Composite Variables: An Application to Air Mobility Command's 30-Day Channel Route Network," M.S. Thesis, MIT Operations Research Center, June 2001.

[9] Nielsen, C., "Large-Scale Network Design Using Composite Variables: An Application to Air Mobility Command's 30-Day Channel Route Network," M.S. Thesis, MIT Operations Research Center, June 2002.

[10] Obermayer, R. W., and Nugent, W. A., "Human-Computer Interaction (HCI) for Alert Warning and Attention Allocation Systems of the Multi-Modal Watchstation (MMWS)," *Proceedings of the 43rd Annual Meeting Human Factors and Ergonomics Society*, 1999.

[11] Ricard, M., Abramson, M., and Adams, M., "ADEPT: A Framework for Intelligent Autonomy," *Proceedings of the AUVSI*, 2000.