Game Theoretic Approach to Threat Prediction and Situation Awareness

Genshe Chen, Dan Shen, Chiman Kwan
Intelligent Automation, Inc.
Rockville, MD, U.S.A.
gchen@i-a-i.com
dshen@i-a-i.com
ckwan@i-a-i.com

Jose B. Cruz, Jr.
Department of Electrical and Computer Engineering,
The Ohio State University
Columbus, OH, U.S.A.
cruz.22@osu.edu

Martin Kruger
Office of Naval Research
875 North Randolph Street,
Suite 1425
Arlington, VA 22203
Martin_Kruger@onr.navy.mil

Abstract - The strategy of data fusion has been applied in threat prediction and situation awareness and the terminology has been standardized by the Joint Directors of Laboratories (JDL) in the form of a so-called JDL Data Fusion Model, which currently called DFIG model. Higher levels of the DFIG model call for prediction of future development and awareness of the development of a situation. It is known that Bayesian Network is an insightful approach to determine optimal strategies against asymmetric adversarial opponent. However, it lacks the essential adversarial decision processes perspective. In this paper, a highly innovative data-fusion framework for asymmetric-threat detection and prediction based on advanced knowledge infrastructure and stochastic (Markov) game theory is proposed. In particular, asymmetric and adaptive threats are detected and grouped by intelligent agent and Hierarchical Entity Aggregation in Level 2 and their intents are predicted by a decentralized Markov (stochastic) game model with deception in Level 3. We have verified that our proposed algorithms are scalable, stable, and perform satisfactorily according to the situation awareness performance metric.

Keywords: Data fusion, Game Theory, Situation Assessment, Threat Assessment, Situation awareness, Asymmetric Threats.

1 Introduction

Data fusion process has been largely applied to symmetric military warfare in which long-term strategic target development processes have developed the signatures or deductive model-based templates describing the component targets of the fielded adversary forces [1-2]. Asymmetric adversaries, such as Camouflages, Concealment, Deceptions, and “unilateral destruction”, on the other hand, are quite unpredictable in their behavior, tactics, weapons, and the choice of targets. Information and patterns of behavior that could provide advanced warning of hostile intent are often hidden in a vast background of harmless civilian activity. Automated processing techniques are needed to augment tactical intelligence analysis capabilities by automatically identifying the militarily-relevant features of all available data of different modalities (e.g., signals intelligence, human intelligence, imagery intelligence, etc.) and recognizing patterns that are out of the ordinary and/or indicate probable hostile intent [3].

As asymmetric warfare becomes more prevalent and introduces new security challenges, there is a critical need of strategies for providing actionable information to military decision makers so that the adversaries’ most likely future courses of actions (COAs) can be predicted. By successfully assessing possible future threats from the adversaries, the decision makers can make more effective targeting decisions, plan friendly COAs, mitigate the impact of unexpected adversary actions, and direct sensing systems to better observe adversary behaviors. Information fusion has been evolved as one efficient method to provide such type of information by combining diverse data from multiple sources. Many studies have dealt with the information sources directly, which is the first level of fusion and is better understood. However, to combat the present and future asymmetric threats to national and international security, information fusion developments must progress beyond current Level 1 paradigms.

In this research, we have achieved the following important results. First, we developed a highly innovative data-fusion framework for asymmetric-threat detection and prediction in an urban-warfare setting based on advanced knowledge infrastructure and Markov (stochastic) game theory. It consists of four closely coupled activities: 1) Level-one fusion automates the processing and integration of information from disparate sources to produce an integrated object state; 2) Level-two fusion automates the reasoning and groups the cooperative objects which perform common tasks. The main tasks of level-two fusion are estimation and prediction of relations among entities, and to include force structure and cross force relations, communications and perceptual influences, physical context, etc.; 3) Level-three fusion automates, infers and predicts the intentions and CoAs of asymmetric threats; 4) Level-four fusion uses these CoAs to optimally task available sensor assets to minimize cost of operations and decision response time. In particular, asymmetric and adaptive threats are detected and grouped by intelligent agent and Hierarchical Entity Aggregation in Level 2 and their intents are predicted by a decentralized Markov
(stochastic) game model with deception in Level 3. The proposed framework is modular, integrated, composable and adaptable. Second, we have implemented our graphical model based object refinement algorithm at level 1. We have implemented Hierarchical Entity Aggregation and Ontology-based Factlet Analysis Function to detect asymmetric treats at level 2. We have implemented an adversary Markov game [8] model with three players: red force (enemies), blue force (friendly forces), and white force (neutral objects) at level 3. Inherent information imperfection is considered and implemented in two methods: 1) decentralized decision making scheme; and 2) deception with bounded rationality. We have modified our game theoretic sensor management algorithm at level 4. Third, we have conducted extensive literature review to determine a reasonable set of performance metrics for the proposed system and evaluate its performance from this. Fourth, a software prototype has been developed with connectivity to the MICA (Mixed Initiative Control of Automa-Teams) Open Experimental Platform (ontology-based virtual battlespace) to integrate levels 1, 2, 3, and 4 data fusion and to demonstrate the performance of our proposed algorithms.

The paper is organized as follows. In section 2, we will summarize the technical approach, which includes overall architecture, hierarchical entity aggregation at level 2, and Markov game approach at level 3. Section 3 describes the experimental results. Section 4 concludes the paper.

2 Threat Prediction/Situation Analysis

2.1 Overall Structure

The overall architecture of our game theoretic data fusion is shown in Fig. 1. The level-one fusion builds the tracks of enemy targets from the reported data and writes the Red target track table, which contains time, location, target type beliefs and other information about each target. The tracks are based on data from the Blue UAV and AWACS sensors. Field reports from forward observers and signal intelligence contributes to event data. Level-one fusion establishes and maintains tracks for all ground vehicles, makes track-to-track associations, eliminates duplicates, and also initiates, maintains and drops tracks. The Blue track tables of tracks of friendly armament resources contain similar information. The main approach is graphical model.

The Level-two fusion performs spatial and temporal processing on tracks produced by Level-one multi-sensor, multi-target track fusion, supplemented with intelligence information from both structured data sources such as databases and unstructured data sources such as ontology-based documents. At this level, Hierarchical Entity Aggregation, Ontology and Factlet Analysis Function are used to cluster red entities into groups by position, find the group centers-of-mass, build target group tables, and determine certain of their critical events and behaviors over time, which it formats into frame structures to pass to the Level-three fusion process.

At level-three (threat refinement) fusion, we investigated and demonstrated the effectiveness of Markov game theory. An adversarial Markov game framework is proposed for threat refinement to drive existing and newly formulated models of threat behavior with factlets derived from situation refinement to support the determination of possible enemy COAs. An artificial intelligence planning concept, Hierarchical Task Network, is exploited to decompose the estimated ECOAs and the decompositions are feedback to level 2 (situation refinement) and used to identify and group the enemy entities that pose threats.

At level 4 (process refinement), the main tasks are to
perform resource allocation and to provide feedback information for Level 1, 2, and 3 to adjust the parameters. We use the method developed by IAI and OSU in a Navy funded on-going Phase II project named “Adaptive Cooperative Path and Mission Planning for Multiple Aerial Platforms”.

The key advantages of our proposed techniques are as follows: **First,** the proposed framework is modular, integrated, composable and adaptable. **Second,** the graph based Markov game approach is very effective when we have to confront an intelligent adversary. The nature of military conflicts is appropriately depicted by the Markov game: the determination of the strategies of one force is tightly coupled to the determination of the strategies of the opposing force. The Nash equilibrium is always guaranteed because in Markov game mixed strategies are applied. The belief of each possible ECOA is directly obtained from the Markov game result. With the consideration that the threat may manipulate the information available to friendly force (such as camouflage, jamming and decoy), we integrate the deception concept in our game approach to model the action of purposely rendering partial information to increase the payoffs of the threat. **Third,** ontology-based information representation is able to describe a set of entities (concrete and/or abstract) and the relationships between them as they evolve over time. Such relationships between these core entities can capture sufficient information about a situation, thus further supporting high-level reasoning and decision-making. **Fourth,** ECOA hypotheses and game theoretic sensor management can be predicted and evaluated in a dynamic way. This is very important especially when we face an asymmetric adversary because of the difficulty in enumerating possible ECOAs in advance for an asymmetric and adaptive adversary. **Fifth,** Hierarchical Entity Aggregation is a novel approach to Entity Aggregation which hierarchically groups aggregates based on similarity in their objectives as opposed to similarities in composition or spatial coordinates. Thus, non-obvious asymmetric aggregates can be identified. This approach also effectively reduces the hypotheses space for Level-three fusion by reducing the number of potential “threats” to consider, which consequently facilitates the application of the Markov game. **Sixth,** Sensor planner’s decision making and engagement recommendations use joint value of information (VoI) evaluation via influence diagrams, which not only makes it possible to identify synergy among multiple decisions, but also is computationally efficient to permit real-time processing for large networks. **Finally,** information representation based on graphical models is able to describe a set of entities (concrete and/or abstract) and the relationships between them as they evolve over time. Graph modeling also enables a unified approach for all levels of data fusion, allowing hypotheses to populate from various possible fusion methods. For higher-level fusion processing, data of different granulation are treated effectively through entity aggregation at lower levels.

We have conducted the implementation and analysis of several data fusion approaches at every level of JDL (DFIG) data fusion model [13]. We drive existing and newly formulated algorithms to support the determination of possible enemy COA. Asymmetric threats will be identified efficiently by Hierarchical Entity Aggregation at level-two fusion (situation refinement) and assigned special payoff functions in our Markov Game framework at level-three fusion (treat refinement) so that the intents of these irrational threats or entities will be efficiently predicted.

Due to page limitations, here we focus only on level-two and level-three data fusion and details can be found in the following subsections. A related paper summarizing our results with respect to level-one fusion algorithm will appear elsewhere.

### 2.2 Level 2 Situation Refinement

As a level-two fusion process within the JDL fusion model, Hierarchical Entity Aggregation [4-7] is designed to identify and group the entities that pose threats so that level-three fusion can be performed efficiently because of the following two major reasons. Hierarchical Entity aggregation reduces the ECOA hypothesis space for level-three fusion by reducing the number of potential “threats” to consider. In our approach, applying Markov (stochastic) game theoretic algorithm to predict ECOA becomes more feasible. The other is that Hierarchical Entity Aggregation can identify the asymmetric threats efficiently as well. Then these identified asymmetric units will be handled by our proposed Markov games.

The objectives of level-two fusion (situation refinement) include making inference about the measurements and observations that are available and establishing relationships between entities, events and the environment. Ontology-based battle-space modeling technique provides the feasibility to the representation and organization of the environmental observations in a machine-readable manner. It also facilitates inference of the potential relationships among the entities.

The Factlet Analysis Functions execute across the extent of the Virtual Battlespace, reasoning across the objects present, and within each analysis perspective, to generate both measured and inferred items of evidence, the “factlets”. These Functions are concerned with establishing the “relationships” between objects in the Virtual Battlespace. For example, the Motion Analysis Function considers the movement patterns of groups (established by the Aggregate Analysis Function) of military objects such as armored personnel carriers. The Motion Analysis Function may conclude that the current movement pattern indicates a probing behavior on the part of the adversary, rather than a full scale attack. This inference becomes a factlet. Factlets are statements or evidence about the situation in the battlespace and they form the main input to the Level 3.
2.3 Level 3 Threat Refinement

2.3.1 A Decentralized Stochastic Game Theoretical Model

As shown in Fig. 2, a decentralized Markov game is used to model the evolution of ECOAs originated from an initial prediction based on Hierarchical Entity Aggregation. Our approach has three main features: 1) Decentralized. Each cluster or team makes decisions mostly based on the local information. We put more autonomy in each group allowing for more flexibilities. 2) Markov Decision Process (MDP) can model the uncertainties in the noisy military environment. 3) Game framework [10] is an effective and ideal model to capture the nature of military conflicts.

A Markov (stochastic) game [8] is given by (i) a finite set of players \( N \), (ii) a finite set of states \( S \); (iii) for every player \( i \in N \), a finite set of available actions \( D^i \) (we denote the overall action space \( D = \times_{i \in N} D^i \ )); (iv) a transition rule \( q : S \times D \rightarrow \Delta(S) \), (where \( \Delta(S) \) is the space of all probability distributions over \( S \) ); and (v) a payoff function \( r : S \times D \rightarrow \mathbb{R} \). For our threat prediction problem, we obtain the following discrete time Markov game:

Players (Decision Makers) --- Although, in our distributed (decentralized) Markov game model, each group (cluster, team) makes decisions, there are three main players: enemy, friendly force, and white objects. All clusters of enemy (friendly force, or white objects) can be considered as a single player since they have a common objective.

State Space --- All the possible COAs for enemy and friendly force consist of the state space. An element \( s \in S \) is thus a sample of enemy and friendly force COAs composed of a set of triplets (Resource, Action Verb, and Objective). As an example, an enemy COA might be: the red team 1 (Resource) attacks (Action Verb) the blue team 2 (Objective). Similarly, for the friendly force COAs, Resource is a friendly asset and Objective is an adversary entity. \( s = (s^e, s^f, s^w) \) and \( S = S^e \times S^f \times S^w \), where \( s^e \in S^e \) is the COAs of Blue (friendly) force and \( s^f = (s^e, s^f, s^w) \), \( r^e \in R^e \), \( a^e \in A^e \), \( o^e \in O^e \) \( R^e, A^e, O^e \) are the set of the resource, action, and objective of blue force, respectively. On the other hand, \( s^w \in S^w \) is the COAs of Red (enemy) force and \( s^w = (s^e, s^f, s^w) \), \( r^w \in R^w \), \( a^w \in A^w \), \( o^w \in O^w \) \( R^w, A^w, O^w \) are the set of the resource, action, and objective of blue force, respectively.

Action Space --- At every time step, each blue group chooses a list of targets with associated actions and confidences (probability distribution over the list of targets, i.e., the sum of the confidences should be equal to 1) based on its local battle field information, such as the unit type and positions of possible targets, from level-two data fusion. Let \( D^b \) denote the action space of the \( i^b \) blue team. Each element \( d^b_i \) of \( D^b \) is defined as

\[
d^b_i = \{ (a^b_i, t^b_i, p^b_i) | a^b_i \in A^b, t^b_i \in O^b, 0 < p^b_i \leq 1, \sum p^b_i = 1 \} \quad (4)
\]

where \( p^b_i \) is the probability of the action-target couple \( (a^b_i, t^b_i) \), which defined as the action \( a^b_i \) to target \( t^b_i \).

Therefore, the action space of blue side \( A^b = \times_{i \in N} D^b \). As an example, for the blue small weapon UAV 1 in blue team 1, its action might be \( d^b_1 = \{(\text{attack}, \text{red fighter } 1, 0.3), (\text{fly to}, \text{red fighter } 2, 0.5), (\text{avoid}, \text{red fighter } 3, 0.2)\} \).

Similarly, each red cluster (obtained from the level-two data fusion) determines a probability distribution over all possible action-target combinations. Let \( D^r \) denote the action space of the \( i^r \) red cluster. Each element \( d^r_i \) of \( D^r \) is defined as

\[
d^r_i = \{ (a^r_i, t^r_i, p^r_i) | a^r_i \in A^r, t^r_i \in O^r, 0 < p^r_i \leq 1, \sum p^r_i = 1 \} \quad (5)
\]

where \( p^r_i \) is the probability of action \( a^r_i \) to target \( t^r_i \).

Therefore, the action space of red force \( A^r = \times_{i \in N} D^r \). A possible action for red platform 1 (red fighter 1) is \( d^r_1 = \{(\text{attack}, \text{small weapon UAV } 1, 0.6), (\text{move to}, \text{blue soldier } 2, 0.2), (\text{avoid}, \text{blue soldier } 1, 0.2)\} \). Remark: Action and Action Verb are different concepts. Action is a set of triplets with associated probabilities while Action Verb is just a component of triplet composed of Resource, Action Verb and Objective. All Actions are included in \( A^r \) for player 1 (Blue force) and \( A^f \) for player 2 (Red force). All Action Verbs are enumerated in \( A^f \) for player 1 (Blue force) and \( A^r \) for player 2 (Red force).

The actions of white objects are relatively simple. The main action type is the movement. Let \( D^w \) denote the action space of the \( i^w \) white unit. Each element \( d^w \) of \( D^w \) is defined as

\[
d^w = \{ (a^w, t^w, p^w) | a^w \in A^w, t^w \in O^w, 0 < p^w \leq 1, \sum p^w = 1 \} \quad (6)
\]

where \( p^w \) is the probability of action \( a^w \) to target \( t^w \).

Transition rule --- Due to the uncertainty properties of military environments, we assume that the states of the Markov game have inertia so that the decision makers have more chance in pursuit of the objective of the previous actions. We define an inertia factor vector \( \eta = (\eta_1, \eta_2, \cdots, \eta_m) \) for player \( i \), where \( m \) is the number of the teams or clusters of player \( i \), and \( 0 \leq \eta_j \leq 1, 1 \leq j \leq m \).
So, for the $j^{th}$ team of the $i^{th}$ player, there is a probability of $\eta_j$ to keep the current action-target couple and a probability of $(1-\eta_j)$ to use the new action composed of action-target couples.

There are two steps to calculate the probability distribution over the state space $S$, where $s_1, s_2, \ldots, s_m$ are states of time step $k$ and $k+1$ respectively, $a_i^j, a_i^j$ are the decisions of player 1 (blue force or friendly force) and player 2 (red force or enemy) respectively, at time step $k$.

Step 1: with the consideration of inertia factor vector $\eta'$, we combine the current state with decisions of both players to obtain fused probability distributions over all possible action-target couples for red and blue forces. To do this, we first decompose the current state into the action-target couples for each team of player (red force or blue force). Let $\Psi(s_i)$ denote the resulting action-target couple related to the $j^{th}$ team of the $i^{th}$ player. For example, if there is one trip of (blue team 1, attack, red fighter 2) in the current state $s_k$, then the action-target couple for blue team 1 (the first team of blue force) is $\Psi(s_1) = \text{(attack, red fighter 2)}$. Secondly, for each specified team, say the $j^{th}$ cluster of player 2, we combine the current state with decisions of both teams or clusters of player 2, and then we know the probability of the action-target couple $(a_j^i, t_j^i)$ in current state $s_k$ is 0 and probability of $(a_j^i, t_j^i)$ in current action is 0. So, according to the definition of inertia, the fused probability of the action-target couple $(a_j^i, t_j^i)$ is $0(1-\eta_j^i)+1(\eta_j^i) = \eta_j^i$. 4) The action-target couple $(a_j^i, t_j^i)$ occurs neither in the current state $s_k$ nor in the current action of $j^{th}$ cluster of player 2, and then we know the probability of $(a_j^i, t_j^i)$ in current state $s_k$ is 0 and probability of $(a_j^i, t_j^i)$ in current action is 0. So, according to the definition of inertia, the fused probability of the action-target couple $(a_j^i, t_j^i)$ is $0(1-\eta_j^i)+0(\eta_j^i) = 0$.

Similarly, the new probability distribution for the $j^{th}$ team of player 1 (blue force) is

$$p_j^i(1-\eta_j^i), (a_j^i, t_j^i, p_j^i) \in d_j^i \text{ and } (a_j^i, t_j^i) \in \Psi(s_i) \quad (7)$$

$$p_j^i(1-\eta_j^i)+\eta_j^i, (a_j^i, t_j^i, p_j^i) \in d_j^i \text{ and } (a_j^i, t_j^i) \in \Psi(s_i) \quad (8)$$

$$\eta_j^i, (a_j^i, t_j^i, p_j^i) \notin d_j^i \text{ and } (a_j^i, t_j^i) \in \Psi(s_i) \quad (9)$$

$$0, (a_j^i, t_j^i, p_j^i) \notin d_j^i \text{ and } (a_j^i, t_j^i) \notin \Psi(s_i) \quad (10)$$

Step 2: we determine the probability distribution over all possible outcomes of state $s_{k+1}$.

$$q(s_{k+1} | s_k, a_1^i, a_2^i, a_j^i) = \prod_{i=1}^{m_1} \prod_{j=1}^{m_2} \prod_{k=1}^{m_3} q(a_j^i, t_j^i)$$

where $m_1$ is the number of the teams or clusters of player 1 (blue force), $m_2$ is the number of the teams or groups of player 2 (red force) and $m_3$ is the number of the units of player 3 (white force). $\{t_j^i, a_j^i, t_j^i\}$ is the set of the all possible (with positive probability) triplets for the $i^{th}$ team of player 1 (blue). Therefore $\bigcup_{i=1}^{m_1} \bigcup_{j=1}^{m_2} \bigcup_{k=1}^{m_3} \{t_j^i, a_j^i, t_j^i\}$ contains all the possible (with positive probability) triplets for the blue force. From the step 1, we know that the fused probability of each specified $(a_j^i, t_j^i)$ is $p_j^i(1-\eta_j^i)+0(\eta_j^i)$ defined in equation (4). With the assumption that all teams of blue force are independent, we obtain the overall probability of blue force, $\prod_{i=1}^{m_1} \prod_{j=1}^{m_2} \prod_{k=1}^{m_3} q(a_j^i, t_j^i)$.

Similarly, $\prod_{i=1}^{m_1} \prod_{j=1}^{m_2} \prod_{k=1}^{m_3} q(a_j^i, t_j^i)$ and $\prod_{i=1}^{m_1} \prod_{j=1}^{m_2} \prod_{k=1}^{m_3} q(a_j^i, t_j^i)$ are the overall probabilities of the enemy and white force, respectively. So the probability distribution over all possible outcomes of state $s_{k+1}$ (composed of all possible sub-states of blue force and red force) can be calculated via equation (10).
Payoff Functions --- In our proposed decentralized Markov game model, there are two levels of payoff function for each player (enemy or friendly force).

The lower level payoff functions are used by each team or cluster to determine the team actions based on the local information. For the \( j \)th team of blue force, the payoff function is defined as 

\[
f^j_\alpha (s_j^\alpha, d_j^\alpha, W_j^\alpha) = \sum_{i=1}^{n} w^\alpha(i, a_i^\alpha, t_i^\alpha, W_i^\alpha) p_i^\alpha g^\alpha(i, a_i^\alpha, t_i^\alpha, s_j^\alpha) \tag{11}
\]

where, \( w^\alpha(j, a_i^\alpha, t_i^\alpha, W_i^\alpha) \) will calculate the weigh for any specified action-target couple for \( j \)th team of blue force from the \( W_i^\alpha \), \( p_i^\alpha \) is the probability of the action-target couple \((a_i^\alpha, t_i^\alpha)\), and \( g^\alpha(j, a_i^\alpha, t_i^\alpha, s_j^\alpha) \) will determine the gain from the action-target couple \((a_i^\alpha, t_i^\alpha)\) for \( j \)th team of blue force according to the positions and features, such as platform values and defense/offense capability, of blue and red platforms. Similarly, we obtain the lower level payoff functions for the \( j \)th team of red or enemy force,

\[
f^j_\beta (s_j^\beta, d_j^\beta, W_j^\beta) = \sum_{i=1}^{n} w^\beta(i, a_i^\beta, t_i^\beta, W_i^\beta) p_i^\beta g^\beta(i, a_i^\beta, t_i^\beta, s_j^\beta) \tag{12}
\]

\[
f^j_\gamma (s_j^\gamma, d_j^\gamma, W_j^\gamma) = \sum_{i=1}^{n} w^\gamma(i, a_i^\gamma, t_i^\gamma, W_i^\gamma) p_i^\gamma g^\gamma(i, a_i^\gamma, t_i^\gamma, s_j^\gamma) \tag{13}
\]

For some asymmetric threats, such as suicide bombers, the payoff functions may only consider the loss of the blue side. For some camouflage, and concealment entities, their objectives are to hide themselves and move close to the blue units. Other deception units will do some irrational movements to hide their true goals with the cost the time.

The top level payoff functions are used to evaluate the overall performance of each player.

\[
J^s = \sum_k \left[ \sum_{s_k^\beta} f^s_k (s_k^\beta, d_k^\beta, W_k^\beta) \right] \tag{14}
\]

\[
J^x = \sum_k \left[ \sum_{s_k^\alpha} f^x_k (s_k^\alpha, d_k^\alpha, W_k^\alpha) \right] \tag{15}
\]

\[
J^p = \sum_k \left[ \sum_{s_k^\gamma} f^p_k (s_k^\gamma, d_k^\gamma, W_k^\gamma) \right] \tag{16}
\]

where \( k \) is the time index. In our approach, the lower level payoffs are calculated distributed and sent back to commander/ supervisor via communication networks.

The strategies --- In this project, we have tried several well known types of strategies. Here we only give a brief description about the following three of them:

**Pure Nash Strategies with finite horizon.** In game theory, the Nash equilibrium (named after John Nash \[9\]) who proposed it) is a kind of optimal collective strategy in a game involving two or more players, where no player has anything to gain by changing only his or her own strategy. If each player has chosen a strategy and no player can benefit by changing his or her strategy while the other players keep theirs unchanged, then the current set of strategy choices and the corresponding payoffs constitute a Nash equilibrium. In our approach, we use a game search tree to find the solution.

In our proposed approach, the solution to the Markov game tree is obtained via a \( K \) time-step look-ahead approach, in which we only optimize the solution in the \( K \) time-step horizon. \( K \) usually takes 2, 3, 4, or 5. The suboptimal technique is used successfully for reasoning in games such as chess, backgammon and monopoly.

**Mixed Nash Strategies.** A mixed strategy is used in game theory to describe a strategy comprised of possible actions and an associated probability, which corresponds to how frequently the action is chosen. Mixed strategy Nash equilibria are equilibria where at least one player is playing a mixed strategy. Nash proved that that every finite game has Nash equilibria but not all has a pure strategy Nash equilibrium.

**Correlated Equilibria** \[8\]. Unlike Nash equilibria, which are the concept of equilibria formulated in independent strategies, the correlated equilibria were developed from the correlated strategies in noncooperative games. The correlated equilibrium of a Markov game describes a solution for playing a dynamic game in which players are able to communicate but are self-interested. Based on the signals generated by the correlated devices and announce to the each decision maker, players choose their actions according to the received private signals. There are two types of correlation devices: autonomous and stationary devices. An autonomous correlation device is a pair \( D = ((M^n_i)_{i \in N}, d_n)_{i \in N} \), where (i) \( M^n_i \) is a finite set of signals for player \( i \) at time step \( n \), and (ii) \( d_n : M(n) \to \Delta(M_n) \), \( M_n = x_{i=1} M^n_i \) and \( M(n) = M_1 \times M_2 \times \cdots \times M_n \). A stationary correlation device is a pair \( D = (((M^n_i)_{i \in N}, d)) \), where \( d \in \Delta(M) \) and \( M = x_n M^n \). Actually, a stationary correlation device is a special case of an autonomous correlation device, where \( M^n_i \) is independent of \( n \) and \( d_n \) is a constant function that is independent of \( n \).

Given a correlation device \( D \), we define an extended game \( G(D) \). The game \( G(D) \) is played exactly as the original game, but at the beginning of each stage \( n \), a signal combination \( m_n = (m^n_i)_{i \in N} \) is drawn according to the probability function \( d_n(m_1, m_2, \cdots, m_n) \) and each player \( i \) is informed of \( m^n_i \). Then each decision maker must base his choice of actions on the received signal. Any deviator will be punished via his min-max value. The punishment only occurs if a player disobeys the recommendation of the device. Every Markov game with an autonomous correlated device admits a correlated equilibrium \[8\].
2.3.2 Hierarchical Task Network

Once the ECOA hypotheses have been generated, they must be evaluated. However, since the generated hypotheses are not directly observable, they are not suitable for correctness testing. As with any hypothesis test, observables must be identified. These observables act as indicators to refute or support ECOA hypotheses. A Hierarchical Task Network (HTN) planner [11] is employed to decompose ECOA hypotheses into observable task sequences.

A construct known as the Hierarchical Task Network (HTN) provides a representation of tasks at various levels of specificity. The HTN not only mimics the variation in specificity found in military echelons, it also allows a computational construct for analyzing ECOAs. In our game theoretic approach to level-three fusion (threat refinement), the HTN is employed to provide a method for decomposing high-level ECOAs into more specific tasks. The HTN representation is the basis of most modern planning algorithms. It is based on the concept that humans plan by decomposing tasks into smaller ones until a sequence of tractable tasks are found that satisfy the objective [12]. These are tasks that the fusion processes attempt to infer or observe directly and are assumed to be tractable.

3 Experiments

In the Simulation part, we build a virtual battle-space and a typical urban scenario based on the Ontology concept, which is an explicit, formal, machine-readable semantic model that defines the classes (or concepts) and their possible inter-relations specific to some specified domain. To simulate our data fusion approach, we implemented and tested our battle-space, scenario and algorithms on the MICA Open Environment Platform (OEP) based on the Boeing C4Isim simulation, which models the collection, processing, and dissemination of battlefield information.

We used a scenario shown in Fig. 3 to demonstrate the performance of our proposed threat prediction and situation awareness algorithm. In the shown urban environment, the blue force’s missions are to capture two bridges and to do security patrol on the mains roads connecting two bridges. The blue ground force consists of 3 teams of three soldiers each with Sniper Rifles. The red force includes 3 armed fighters and 3 asymmetric adversaries hiding in and acting like the white objects (the civilians and vehicles). We assume there is an asymmetry in total forces between blue side and red side. Blue side has more soldiers than red side. Moreover, the objectives of blue side and red side are asymmetric: the objectives of red side are to kill blue forces without considering the loss of themselves and the consideration of collateral damage. The main challenge for both sides is to understand the situation from the fused sensor data and predict the intent of the opponent under the “believed” war situation.

For the scenario, in a specific simulation run (Markov game approach with correlated equilibrium), blue team 1 and blue team 3 were assigned to secure Bridge 1 and Bridge 2, respectively, almost in the whole simulation period of 30 minutes. Blue team with 3 blue soldiers was doing security patrol on the two major roads connecting two bridges and some important areas. On the other hand, red fighters and asymmetric adversaries are trying their best to kill blue forces. The first battle happened when Red Fighter 2 tried to attack Blue Team 2 with the help of an asymmetric white vehicle with deception (hiding in white vehicles). During this period, our algorithm detected and killed one asymmetric adversary vehicle with deception. Without the help of the red vehicle, red fighter 2 was killed by blue team 2. Almost at the same time, the asymmetric adversaries near Bridge 1 and Bridge 2 were attacking the blue team 1 and 2. At this stage, two civilians were detected and killed as asymmetric adversaries. Without the help from the asymmetric adversaries with deception, red fighter 1 and 3 were killed by blue team 1 and 3 at bridge 1 and 2, respectively. In this specific run, there is no loss of blue soldiers since our algorithm predicted the intents of the red side correctly and promptly.

In addition to the explained run, we performed many experiments. We compared the results using the various options, such as without game theoretic fusion (without levels 2 and 3), without asymmetric-threat prediction (with level 2 but the payoff function of game model at...
level 3 doesn’t change dynamically), game approach with mixed Nash strategy, game approach with correlated equilibria, and the game approach without collateral damage consideration in the cost function of blue side. Since the simulation is stochastic, the results consist of the mean of 10 runs for each case, which are shown in Fig. 5 (Only the damage information of the Blue side is shown). From the damage comparison results, we can see that our proposed Markov game framework with correlated equilibrium and deception consideration for threat detection and situation awareness is better than the other methods.

Fig. 5: Damage comparison of various options

4 Conclusions
Game theoretic tools have a potential for threat prediction that takes real uncertainties in Red plans and deception possibilities into consideration. In this paper, we have evaluated the feasibility of the advanced data fusion algorithm and their effectiveness through extensive simulations. We have verified that our proposed algorithms are scalable, stable, and perform satisfactorily.

Acknowledgements
This research was supported by the US Navy under contract number N00014-05-M-0205. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Navy.

References