An Information Fusion Game Component

JOEL BRYNIELSSON
Swedish National Defence College

STEFAN ARNBORG
Royal Institute of Technology

Higher levels of the data fusion process call for prediction and awareness of the development of a situation. Since the situations handled by command and control systems develop by actions performed by opposing agents, pure probabilistic or evidential techniques are not fully sufficient tools for prediction. Game-theoretic tools can give an improved appreciation of the real uncertainty in this prediction task, and also be a tool in the planning process. Based on a combination of graphical inference models and game theory, we propose a decision support tool architecture for command and control situation awareness enhancements.

This paper outlines a framework for command and control decision-making in multi-agent settings. Decision-makers represent beliefs over models incorporating other decision-makers and the state of the environment. When combined, the decision-makers' equilibrium strategies of the game can be inserted into a representation of the state of the environment to achieve a joint probability distribution for the whole situation in the form of a Bayesian network representation.

Manuscript received December 31, 2004; revised January 30, 2006 and June 11, 2006; released for publcation on July 30, 2006.

Refereeing of this contribution was handled by Shozo Mori.

Authors' addresses: J. Brynielsson, Swedish National Defence College, PO Box 27805, SE-115 93 Stockholm, Sweden, E-mail: (joel@kth.se); S. Arnborg, Royal Institute of Technology, SE-100 44 Stockholm, Sweden, E-mail: (stefan@nada.kth.se).

1557-6418/06/$17.00 © 2006 JAIF

1. INTRODUCTION

The military domain is one of the purest possible game arenas, and history is full of examples of how mistakes in handling uncertainty about the opponent have had large consequences. For an entertaining selection, see, e.g., [16]. Commanders on each side have resources at their disposal, and want to use them to achieve their, mostly opposing, goals. In the network centric warfare [1, 2] era, they are aided by large amounts of information about the opponent from sensors and historical data bases, and about the status of their own resources from their own information technology infrastructure. In recently proposed infostructures for command and control (C2) [12], decision support tools play a prominent role. These tools seldom include game-theoretic means. Gaming is, however, a prominent feature of military training and the regulated decision processes often assign the roles of red and blue players to staff officers in manual planning activities [52]. Gaming is thus a conceptual part of the planning process in many organizations. It must be emphasized, however, that there are significant differences between practice and theory in application of such regulations. It has, for example, been shown in studies that the Swedish defense organization practices a more naturalistic decision-making process than the recommended one [51]. A pure naturalistic planning process relies more on unobservable mental capabilities of decision-makers than on rational analyses of alternative moves and their utilities [28]. The most common way to deal with uncertainty is, however, to make an assumption—and to forget that it was made. These observations have been the starting point for introducing a less complex planning model—PUT (Planning Under Time-pressure)—in the Swedish defense organization. PUT is based on analyses of a few opponent alternatives and incremental improvement of one's own plans [51]. It thus has potential for the use of gaming tools, provided they are realized in a way that supports subjective improvement of decision situations and decision quality [3].

Data fusion aims at providing situation awareness at different levels for a commander. The JDL model [47, 56] has been proposed for structuring the fusion process into five levels where the third level consists of higher level prediction of possible future problems and possibilities. We believe that the problem of predicting the future in a C2 context comes in two variations that differ in complexity and dependencies: the problem of capturing all aspects of a complex situation, and the problem of strategic dependence in a multi-agent encounter. Considering the former problem, the influence diagram is a well-established and appropriate modeling technique for modeling everything that is not dependent on our own or the opponents' actions, for example doctrine and terrain. Efforts in this direction have been proposed; see for example [50] for a discussion about doctrine modeling using dynamic Bayesian networks [40].

Looking at the latter problem, predicting decisions is also a game-theoretic problem, which has been noted in a recent proposal for revisions to the JDL model where the authors suggest the use of game-theoretic algorithms for the estimation process in higher level data fusion [37].

In this paper, we outline a schematic model using influence diagrams to obtain parameters for a description of the situation in the form of a Bayesian game. The result from the game is a description of equilibrium strategies for participants that can be incorporated in the influence diagram to form a Bayesian network (BN) description of the situation and its development, changing decision nodes to chance nodes.

We review some applications of gaming and simulation in Section 2 and describe our use of influence diagrams in Section 3. Section 4 gives background and some historical notes regarding agent interaction and Section 5 gives a short background on game theory. Section 6 contains an outline of the game component representation. Section 7 discusses solutions and addresses the problem of obtaining these solutions in a computationally feasible manner. In Section 8 we illustrate the use of Bayesian game-theoretic reasoning for operations planning by transforming a decision situation into a Bayesian game that we solve. Section 9 addresses the problems and possibilities that the ambiguities typical for a game-theoretic solution pose. Section 10 discusses related work and Section 11 is devoted to conclusions and discussion regarding future research.

2. THE GAMING PERSPECTIVE

Tools proposed to support the gaming perspective include microworlds [10, 17, 35], which are computer tools where several operators train together; and computer-intensive sensitivity analyses of simple models [22, 39]. There are also large numbers of full and small scale simulation systems used to assess effectiveness of new types of equipment and ways to use them. These microworld and simulation systems are used for off-line analyses to define recommended strategies in conceivably relevant situations.

Systems built for real-time decision-making can take advantage of anytime algorithms with which a coarse prediction can be obtained instantly but is subject to successive refinements when additional time, resources and observations arrive. In such a system, refinements are typically based on either solution improvement or solution re-calculation. An interesting prototype system based on solution improvement is [25] where the situation picture is continuously improved as new observations arrive. The method used is particle filtering, a method where new observations strengthen, weaken or eliminate current hypotheses. A somewhat similar prototype system based on solution re-calculation is [9] where a predicted future situation picture is calculated as a one shot event. Here, solely the particle filtering prediction step is used. The actual choice between the two principles depends on several factors such as the system's intended usage, i.e., whether the decision problem is a one shot problem or a continuous task, and the nature of the problem itself, i.e., whether the present solution can actually be used as basic data for the calculation of a new solution.

Recently, it has become possible to build Bayesian networks to identify the opponent's course of action (COA) from information fusion data using the plan recognition paradigm, which was extended from a single agent context to that of a composite opponent consisting of a hierarchy of partly autonomous units [50]. The conditions for this recognition to work are that the goals and rules of engagement of the opponent are known, and that he has a limited set of COAs to choose from given by the doctrines and rules he adheres to. The opponent's COA can then be deduced reasonably reliably from fused sensor information, such as movements of the participating vehicles. The game component has thus been compiled out of the plan recognition problem. When the goals and resources are not known, these can be modeled as stochastic variables in a BN. However, this is not a strictly correct approach, since the opponent's choice of COA should depend, in an intertwined gaming sense, on what he thinks about our resources, rules of engagement and goals. The situation is essentially a classic Bayesian game, and should be resolved using game algorithms.

3. REALISTIC SITUATION MODELING

It has been suggested that decision-makers often produce simplified and/or misspecified mental representations of interactive decision problems, see, e.g., [34]. Furthermore, most erroneous representations tend to be less complex than the correct ones which, in turn, suggest that decision-makers may act optimally based on simplified and mistaken premises [15]. In this section we discuss and propose the concept of influence diagrams, along with its preliminaries, as a means to specify a reasonably correct representation of the decision problem at hand. An influence diagram is well suited for modeling complex situations. In Section 6, an influence diagram will serve as the underlying model that gives us the basic data needed for the game component.

One goal of artificial intelligence (AI) [45] has been to create expert systems, i.e., systems that can, provided the appropriate domain knowledge, match the performance of human experts. Such systems do not yet exist, other than in highly specific domains, but AI research has inspired important interdisciplinary efforts to solve questions regarding knowledge representation, decision-making, autonomous planning, etc. These results provide a good ground for the construction of C2 decision support systems. Modern expert systems strive for the ideal of a clean separation of its two components; the domain-specific knowledge base and the algorithmic inference engine [14]. Our work proposes generic inference procedures and, thus, targets the inference engine part of the expert system in this regard. During the last decade, the intelligent agent perspective has led to a view of AI as a system of agents embedded in real environments with continuous sensory inputs. We believe that this is a viable way to reason about C2 decision-making and we adopt the agent perspective throughout this paper.

Agents make decisions based on modeling principles for uncertainty and usefulness in order to achieve the best expected outcome. The assumption that an agent always tries to do its best relative to some utility function, is captured in the concept of rationality. The combination of probability theory, utility theory and rationality constitutes the basis for decision theory. The basic elements that we use for reasoning about uncertainty are random variables. General joint distributions of more than a handful of such variables are impossible to handle efficiently, and modeling distributions as Bayesian networks has become a key tool in many modeling tasks.

A BN offers an alternative representation of a probability distribution with a directed acyclic graph where nodes correspond to the random variables and edges correspond to the causal or statistical relationships between the variables. Calculating the probability of a certain assignment in the full joint probability distribution using a BN means calculating products of probabilities of single variables and conditional probabilities of variables conditioned only on their parents in the graph. The BN representation is often exponentially smaller [45] than the full joint probability distribution table and many inference systems use BNs to represent probabilistic information. Another advantage with the BN representation is that it facilitates the definition of relevant distributions from causal links that are intuitively understandable and, in the case of a dynamic BN, develop with time. Successful(?) uses of these networks include the implementation of the "intelligent paper clip" in Microsoft Office [23], although much of its potential functionality was stripped away in the actual deployment.

An influence diagram is a natural extension to a BN incorporating decision and utility nodes in addition to chance nodes, and represents decision problems for a single agent [24]. Decision nodes represent points where the decision-maker has to choose a particular action. Utility nodes represent terminal nodes where the usefulness for the decision-maker is calculated as a function of the values of its parents. These diagrams can be evaluated bottom up by dynamic programming to obtain a sequence of maximum utility decisions.

When designing decision-theoretic systems to be used for C2 decision-making, complex situations arise where one wants to represent knowledge, causality, and uncertainty at the same time as one wants to reason about the situation, simulating different COAs in order to see the expected usefulness of proposed moves. We believe the influence diagram is the right choice for both representation and evaluation and propose a simplified schematic generic diagram in Fig. 1 for the C2 process. C is a discrete random variable representing the consequence of the decisions $D_1,\ldots,D_n$. D_1 represents our own decision and $D_2,\ldots,D_n$ represent the decisions of the other agents. G_1 is a discrete random variable that represents our own goals. U_1 is the utility that we gain after performing decision D_1 depending on the consequence C and our own goals G_1. G_i and U_i are defined similarly for the other agents where $2 \leq i \leq n$.

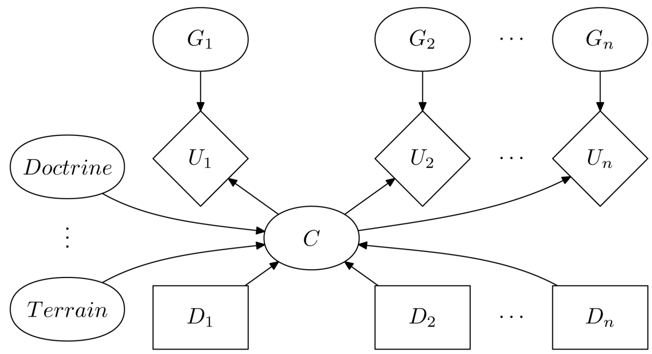

Fig. 1. The C2 process modeled in an influence diagram. Terrain data bases and doctrine are examples of domain-specific subdiagrams that characterize a particular model.

The diagram in Fig. 1 is a simplified representation, to be connected to models—encoded as BNs—of terrain, doctrine, etc., that can be implemented as subdiagrams with causal relationships between different nodes of models. While these subdiagrams are interesting in their own right, they are not the topic of this article. Hence, we have chosen to think of them as existing models that influence the decisions we are modeling.

A problem with the diagram in Fig. 1 is that it does not capture "gaming situations" where one wants to reason about opposing agents that act according to beliefs about one's own actions. Such dependencies are not possible to model in an influence diagram or BN without additional machinery. At this point it should also be noted that the diagram in Fig. 1 should not be considered to be very useful in its own right. Rather, it is a statement of the problem we are trying to solve. Among other things, the diagram is not regular which is a requirement for algorithms that evaluate influence diagrams, see, e.g., [46]. Regularity assumes a total ordering of all of the decisions, a reasonable condition for a single decision-maker who only needs to take his own actions into account.

In this work we use the influence diagram as basic data to develop a generalized technique that solves problems for multiple decision-makers. In Fig. 2 we give an alternative algorithm for evaluation of influence diagrams with multiple agents, inspired from the single agent construction found in [44, 45]. Here, the payoffs for all combinations of alternatives are returned instead of only the alternative with the highest possible payoff.

Input: Influence diagram with decision nodes $D_1, \ldots, D_n$ and utility nodes $U_1, \ldots, U_n$ belonging to agents $1, \ldots, n$ respectively.

Output: An n-dimensional utility matrix A containing, for each possible value combination on the respective agents' decision nodes, the resulting utility vector $(u_1, \ldots, u_n)$.

1. Set the evidence variables for the current world state, i.e., use the percepts that the agent has received to date to assign values to a subset of the random variables in the influence diagram.
2. For each possible value combination of the decision nodes;
   (a) Set the decision nodes to their respective values.
   (b) Calculate the posterior probabilities to the parent nodes of the utility nodes using a standard probabilistic inference algorithm or by the node elimination method of [46].
   (c) Calculate the resulting utility vector for the action combination and store this in the utility matrix A.
3. Return the utility matrix A.

Fig. 2. Algorithm for evaluating an influence diagram where multiple agents make decisions.

4. AGENT INTERACTION

The decision situation that arises in decision node D_1 in the influence diagram depicted in Fig. 1 is characterized by its dependency on other actors' decisions. Standard AI tools for solving decision-making problems in complex situations, such as dynamic decision networks and influence diagrams, are not applicable for these kinds of situations, as the decisions are intimately related to the other agents' decisions. Game theory, on the other hand, provides a mathematical framework designed for the analysis of agent interaction under the assumption of rationality where one tries to identify the game equilibria as opposed to traditional utility maximization principles. A game component in multi-agent decision-making thus uses rationality as a tool to predict the behavior of other agents.

In higher level C2, i.e., threat prediction in a data fusion context, the need of a game component becomes obvious [55]. Circular relationships are not allowed in influence diagrams or other traditional agent modeling techniques and therefore we cannot make the agents' decisions dependent on each other in the diagram in Fig. 1. On lower level C2 this need is not as obvious, because agents' choices are to a large extent driven by standard operating procedures obtained by training and developed using off-line game analyses. On this level, like in helicopter dogfights, successful developments of strategies have been obtained with look-ahead in extensive form, i.e., perfect information game trees with zero-sum payoffs as reported in [27] or moving horizon imperfect information game trees as reported in [54]. The depth of the game tree corresponds to inference of agents' actions that are dependent on each other, i.e., a series of what-if questions such as "what is the usefulness if agent i performs action c_i and the other agents perform actions $c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_n$ which in turn makes agent i respond with action c_i'," etc. Look-ahead algorithms are typically modeled using a discount factor $\gamma \in (0,1)$ that reduces the utility by γ^d where d is the tree depth. For problems in which the discount factor is not too close to 1, a shallow search is often good enough to give near-optimal decisions [45].

Look-ahead game trees have been used successfully for reasoning in, possibly uncertain, games with perfect information where optimal solutions are obtained with the minimax algorithm. Examples of such games are chess, go, backgammon, and monopoly. In the context of C2 we deal with imperfect information which forces us to solve a more complex game, more similar to poker, since we cannot be sure of exactly where we are in the game tree. Although ordinary minimax algorithms cannot be used in our context it is still likely that the ideas from ordinary game play algorithms, such as the famous alpha-beta pruning [29], can be re-used to some extent. This is interesting as these ideas rest on almost a century of research and experience [33, 45].

Decision-making in environments where multiple agents make decisions based on what they think the other agents might do is a difficult problem, and the use of game theory for agent design has so far been limited due to lack of standard implementation methods. We believe, however, that this barrier will be overcome as more research is focused on the use of game theory for agent design. The widely used AI book by Russell and Norvig [45] added a section on game theory just recently which indicates that the ideas are new and still need to be investigated more thoroughly.

One of the barriers that do exist when using traditional game theory for agent design is that it assumes that a player will definitely play a (Nash) equilibrium strategy. This assumption is certainly true in applications where the game is a designed mechanism, such as the management of (own) mobile sensors [26, 57] or the construction of algorithms for efficient network capacity sharing [4]. However, these situations must be considered a small subset compared with the many situations in everyday life that involve uncertainty about both the other actors and the world as a whole. Over time it has come to be recognized that benevolence is the exception; self-interest is the norm [43]. Particularly, in our C2 application self-interest is the norm that commander training seeks to foster. In this work we aim at solving this problem using the Bayesian game technique, which is described below.

Other problems with game theory for agent design are the lack of methods for combining game theory with traditional agent control strategies [45] and the lack of standard computational techniques for game-theoretic reasoning [33].

In this paper we propose the use of a Bayesian game for modeling higher-level agent interaction in an attempt to obtain better situation awareness in a C2 system. As situation awareness is obtained using fusion techniques we believe that the game component is an integral part of the data fusion process and provides information that is needed in level three data fusion processing according to the JDL model [37, 47, 56]. A Bayesian game is a game with incomplete information, that is, at the start of the game the players may have private information about the game that the others do not know of. Also, each player expresses its prior belief about the other players as a probability distribution over what private information the other players might possess.

5. STRUCTURE OF GAMES AND THEIR REPRESENTATIONS

Recent developments in game theory and AI have made applications with significant game components feasible. Most of the work, however, does not address Bayesian games. Many description methods have been developed with algorithmic techniques being able to solve quite large games if they are of the right type. The extensive form of a game is a tree structure, where a non-terminal node can describe a chance move by nature (random draw) or a move possible for one of the participants, and a leaf node represents the end of the game and its payoff after evolving through the path to it. The immediate descendants of a non-leaf represent the alternative outcomes of a chance move (in which case the node is associated with a probability distribution) or the set of actions available for the player in turn at this point. This is adequate for leisure games like chess, a perfect information game, but the chess game tree does not fit into any computer. A deterministic game with full information (like generalized chess or checkers) can be solved if its game tree can be traversed, by bottom-up dynamic programming.

In games with imperfect information, the exact position in the game tree may not be known to players. This is the case in leisure games of cards, where the hand of a player is only available to her. The determination of optimal strategies must use a game tree where the decision is the same for a whole information set, a set of nodes for a player where the information available to her is the same. As an example, at the first bid of a game of contract bridge, each of the possible distributions of the cards not seen by the player is in the same information set. Bottom-up evaluation does not work, because at the lower levels of the game tree the players have information on the hidden information that was communicated by their opponents' choices of moves (like the initial round of bidding in bridge). This situation is solved by putting the game on strategic form, which means that all combinations of moves for all of a player's information-equivalent nodes in the tree, and all chance moves, are listed with their payoffs. Solutions can be found with numerical methods, linear programming techniques for zero-sum games [11] and solution methods for the linear complementarity problem (LCP) for general games [13]. For the former, a unique mixed (randomized) strategy for each player is a non-controversial definition of the game's solution. For the latter, the Nash equilibrium is the accepted solution concept [42]. A Nash equilibrium always, under general assumptions, exists but is less non-controversial since sometimes several equilibria exist, and there are alternative proposals regarding how to find one that is in a tangible way more relevant than the others. The payoff matrix is typically impossibly large, and games of this type, like standard variants of poker and bridge, have no known optimal solution although interesting approximation algorithms have appeared recently [5]. In the above games, all players know the exact structure and payoff system of the game. This is adequate for many purposes, but not for our application.

The concept of a Bayesian game is fairly complex and different views abound in the literature. With notation from [41], a Bayesian game, Γ^{b}, is defined by

$$\Gamma^{\mathrm{b}} = (N,(C_i)_{i\in N},(T_i)_{i\in N},(p_i)_{i\in N},(u_i)_{i\in N}) \tag{1}$$

where N is a set of players, C_i is the set of possible actions for player $i \in N$, T_i is the set of player i's possible types, p_i is a probability distribution representing what player i believes about the other players' types, and u_i is a utility function mapping each possible combination of actions and types into the payoff for player i. It should be noted that the set notation we use differs from standard mathematical notation. Indices contain one or several players in the set N and hence represent the "player dimension." When there is no subscript at all we actually mean a set with a variable for each player in N which is denoted a profile. The subscript $-i$ denotes the set of all players except for player i, i.e., $N \setminus \{i\}$. The other dimension is defined by the letter itself that can be either lower-case, representing one particular choice, or upper-case, representing the set of all possible choices. Henceforth, C_i is the set of possible actions for player i, $c_i \in C_i$ is one of player i's possible actions, $c \in C$ is a possible strategy profile in the game, and C is the set of all possible strategy profiles that we may encounter in the game.

The definition given above is a flat representation given originally in [21]. It seems as if it only states first-order beliefs of players about each other, but this is not a fair perspective. We want to consider all types of higher-order knowledge, such as what player 1 believes that player 2 believes that player 1... believes. This type of information can indeed be modeled in a standard Bayesian game, under quite general conditions, as shown in a strictly mathematical and non-algorithmic argument in [38]. On the other hand, the amount of information required to perform such modeling can be infinite and thus not extractable from, or actually used by, experts and decision-makers. Bayesian games can have infinite type sets even in simple cases like natural analyses of bargaining situations. We will restrict our attention to games with finite type sets and players, since otherwise general solution algorithms do not exist (games with infinite type sets must be analyzed manually to bring about a finite solution algorithm).

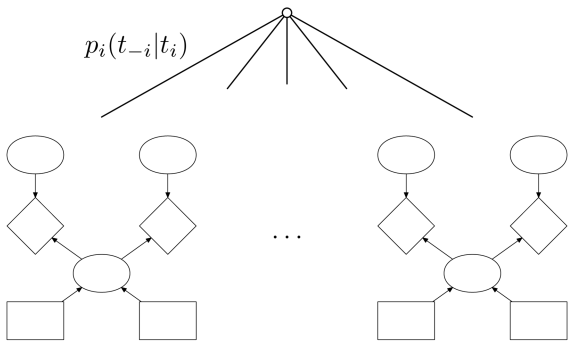

Fig. 3. Architecture overview. Models are represented by influence diagrams that yield payoff values for a Bayesian game.

An important class of Bayesian games is games with consistent beliefs. In this case the player's belief, conditional on his type, about other players' types are all derivable from a global distribution over all players' types by conditioning, i.e., $p_i(t_{-i} \mid t_i) = p(t_{-i} \mid t_i)$. Hence, this class is a subclass of imperfect information games. The assumption of consistent beliefs is both required and natural for most applications; it simply means we should model the players using all information we currently have in our possession. Although game theory means we solve the game for all players at the same time, the solution is still obtained from one particular decision-maker's view of the situation. Therefore, consistent versus inconsistent beliefs becomes more of a philosophical question and we will assume consistent beliefs throughout this work.

Inputs: 1) A list of influence diagrams; one influence diagram for each possible agent model that needs to be considered. Decision nodes $D_1, \ldots, D_n$ and utility nodes $U_1, \ldots, U_n$, belonging to agents $1, \ldots, n$ respectively, need to exist in all diagrams. 2) A common prior probability distribution P over the possible agent models.

Output: Solution proposals for the influence diagram decision variables $D_1, \ldots, D_n$ in the form of mixed strategy Nash equilibria.

1. Let each influence diagram correspond to a Bayesian game type profile $t \in T$, representing that each influence diagram corresponds to different beliefs regarding the participating agents' private information.
2. Formulate the Bayesian game

$$\Gamma^{\mathrm{b}} = \big(N, (C_i)_{i\in N}, (T_i)_{i\in N}, (p_i)_{i\in N}, (u_i)_{i\in N}\big)$$

so that;

(a) N, the set of players, corresponds directly to the set of participating agents,

(b) C_i corresponds to the set of actions available to agent i in decision node D_i in the influence diagrams,

(c) T_i contains the possible types for player i; induced by T according to item 1 above,

(d)

$$p_i(t_{-i} \mid t_i) = \frac{P(t)}{\sum\limits_{s_{-i}\in T_{-i}} P(s_{-i}, t_i)} \quad \text{(consistent beliefs)},$$

(e) $u_i : C \times T \to \mathbf{R}$ is given by the algorithm in Fig. 2, i.e., the algorithm in Fig. 2 needs to be run for each type profile $t \in T$ to obtain the respective utilities given a certain model.

3. Calculate one or more solutions to the Bayesian game in the form of mixed strategy Nash equilibria.
4. Equilibria in the game correspond directly to solution probability distributions over the decision variables $D_1, \ldots, D_n$ in the original influence diagrams. These distributions are returned as solution concepts—not necessarily to be executed, but to further enhance a commander's predictive situation awareness.

Fig. 4. Summary of the game component.

6. THE GAME COMPONENT

In this section we define the proposed information fusion game component using notation from [41]. A brief concept sketch is given in Fig. 3 and a more formal summary is given in Fig. 4 which, in turn, uses the algorithm depicted in Fig. 2. The objective has been to specify an architecture that is suitable for threat prediction in the C2 domain. The most important criteria for the specification of such an architecture are that the agents' decisions are based on their belief regarding the other agents' private information, and that the architecture is made up from an underlying well-established and realistic probabilistic model of the situation. We achieve the former criterion by the use of a game with incomplete information, and the latter criterion by using an influence diagram for representing our model of the current situation awareness.

A top-down perspective on the architecture can be seen in Fig. 3, depicting a probability distribution over the possible worlds. Each such world is modeled in an influence diagram, such as the diagram outlined in Fig. 1, containing nodes for the goals (G_i), the possible courses of actions (D_i), and the payoff (U_i) for each respective agent. Apart from these variables, each influence diagram is connected to model specific subdiagrams containing environmental descriptions, doctrine and other properties specific to the model in question. An important observation regarding the model in Fig. 1 that motivates the use of game theory is the fact that this model, seen as an ordinary influence diagram, does not account for situations when agents' try to make decisions that are influenced by other agents' decisions. That is, it is not capable of representing circular causal relationships between D_1 and D_2. To account for this gaming perspective we therefore think of the possible world states as Bayesian game type profiles. Utilities are obtained for each such type profile by using its correlated influence diagram to create a strategic form game, i.e., utilization of the algorithm in Fig. 2 which for each combination of the decision profile $D_1, \ldots, D_n$ calculates utilities $U_1, \ldots, U_n$.

Using our prior belief regarding which model is accurate, we then obtain a Bayesian game for the whole decision problem. Calculation of equilibria in the Bayesian game yields solutions for the decision variables $D_1, \ldots, D_n$ in the form of mixed strategy Nash equilibria. A more formal description of the scheme can be found in Fig. 4.

Assuming consistent beliefs, the solution to a Bayesian game is obtained by introducing a new root node called a historical chance node that is used to implement the Bayesian property of the game. A historical chance node differs from an ordinary chance node in that the outcome of this node has already occurred and is partially known to the players when the game model is formulated and analyzed. For each set of possible types, the edges from the root node in the game correspond to the model that is used if the players were of this type. We say that a player i believes that the other players' type profile is $t_{-i} \in T_{-i}$ with subjective probability $p_i(t_{-i} \mid t_i)$ given that player i is of type $t_i \in T_i$. Again, note that the subscript $-i$ is standard notation for the set of all players except for player i, i.e., t_{-i} is a list of types for all the other players.

For each type profile $t \in T$, an influence diagram, as in Fig. 1, describes the decision situation using random state variables. The different models differ in properties that cannot be seen in Fig. 1, consisting of other random variables describing for example terrain, doctrine, and belief regarding all kinds of properties that do not rely on other participating agents' decisions. In the context of our Bayesian C2 game the historical chance node is thus a lottery over the possible models that are represented as influence diagrams.

The Bayesian property of the game might seem trivial at first glance, but the historical chance node at the root of the tree poses a serious concern to us. To establish Nash equilibria for the game the normal representation in strategic form is needed, but the algorithm for the creation of this relies on the players being able to decide their strategies before the game begins, which is not true in a Bayesian game that is represented with a historical chance node. The solution, due to Harsanyi [21], is to reduce the game to Bayesian form and compute its Bayesian equilibria. Such an equilibrium consists of a probability distribution over actions for each player and each of this player's types. This can in principle be accomplished by solving an LCP to obtain a mixed strategy for each type of each player. Although in game-theoretic studies, Bayesian games are often defined with infinite type and action spaces, we classify actions discretely after doctrines the players are trained to follow, and if the intuitive type of a player is a continuous variable we discretize it.

At level two, for each node represented by a distinct type profile $t_{-i} \in T_{-i}$, the node is the start of the model that the type profile $t_{-i} \in T_{-i}$ gives rise to. To represent this model we use a game on strategic form; that is, a game with players N, actions $(C_i)_{i \in N}$, and utility functions $(u_i)_{i \in N}$.

The (still Bayesian) game relates to the influence diagram in Fig. 1 in that N represents the n agents that are about to make decisions $D_1, \ldots, D_n$, C_i represents the actions available for agent i in decision node D_i, and u_i is the utility that is obtained in the diamond shaped utility node U_i which is, in turn, depending on the random variables C and G_i denoting the world consequence and the agent's goals respectively.

7. EQUILIBRIA AND COMPLEXITY

While modeling and representing a C2 situation is interesting in its own right, a primary concern is the use and interpretation of the model. In game theory the concept of Nash equilibria defines game solutions in the form of strategy profiles in which no agent has an incentive to deviate from the specified strategy. Without doubt, defining equilibria is the foremost goal in game theory. Fortunately, this means that we can lean on well-established results in our effort to find equilibria for the C2 situation.

For a Bayesian game, Harsanyi [21] defined the Bayesian equilibrium to be any set of mixed strategies for each type of each player, such that each type of each player would be maximizing his own expected utility given that he knows his own type but does not know the other players' types. Mathematically speaking, a Bayesian equilibrium for a Bayesian game Γ^b, as defined in (1), is any mixed strategy profile σ such that, for every player $i \in N$ and every type $t_i \in T_i$,

$$\sigma_i(\cdot \mid t_i) \in \operatorname*{arg\,max}_{\tau_i \in \Delta(C_i)} \sum_{t_{-i} \in T_{-i}} p_i(t_{-i} \mid t_i) \times \sum_{c \in C} \left(\prod_{j \in N-i} \sigma_j(c_j \mid t_j) \right) \tau_i(c_i) u_i(c,t). \quad (2)$$

Here, $\Delta(C_i)$ denotes the set of probability distributions over the set C_i, i.e., the set of possible mixed strategies that player i can choose from, and $\sigma_i(\cdot \mid t_i)$ is the, possibly mixed, strategy of player i in type t_i.

Existence of a Bayesian equilibrium solution in mixed strategies follows from the famous existence theorem for general games, which is due to Nash [42]. Solution methods for general-sum game-theoretic problems are however intractable for the generic case. The most well-known solution method, the Lemke-Howson algorithm [36, 49], solves a linear complementarity problem [13]. The computational complexity for finding one equilibrium is still unclear. We know, according to Nash's theorem [42], that at least one equilibrium in mixed strategies exists but it is problematic to construct one. The Lemke-Howson algorithm exhibits exponential worst case running time for some, even zero-sum, games. However, this does not seem to be the typical case [49]. Interior point methods that are provably polynomial are not known for linear complementarity problems arising from games [49]. Methods amounting to examining all equilibria, such as finding an equilibrium with maximum payoff, have unfortunately been proven **NP**-hard [19], so for these kinds of problems no efficient algorithm is likely to exist.

The standard way of calculating equilibria in a game in extensive form is to transform the game into strategic form. However, the creation of the matrix for the strategic form typically causes a combinatorial explosion. This is due to each value in the matrix representation of a strategic form game representing the payoff for a complete strategy. Hence, even though a game tree typically contains widely different decision alternatives in different subtrees the decisions in the other subtree still need to be considered. Therefore the strategic form matrix dimension grows for each node that is traversed. In a series of articles [30, 31, 48] published during the last decade the sequential form as a replacement for the strategic form has provided a representation suitable for efficient computation of equilibria in an extensive imperfect game with chance nodes. The idea is to replace the game's strategies with new strategies based on sequences ranging from the root node down to the leaves. That is, each sequence represents a possible course of events in the game. As the creation of the matrix for the sequence form relies on payoffs that are already in the tree the problem complexity is reduced from a **PSPACE**-complete problem into a problem that is linear in the size of the tree. However, it should be kept in mind that general game trees often share decision alternatives and, hence, do not exhibit a full scale combinatorial explosion. In totally symmetric problems, as investigated in for example [8], the choice of game representation therefore does not affect the computational tractability significantly. Also, as mentioned above a pre-requisite for the sequential method to be effective is that the game is in extensive form to start with. Referring to the information fusion game component, as outlined in Section 6, this is problematic since the algorithm depicted in Fig. 2 results in a strategic game. However, using an additional chance node denoting the common model prior, it is possible to hinder this combinatorial explosion by transforming the whole game component into one large influence diagram. This influence diagram can then be utilized to create the game tree directly using the multi-agent influence diagram conversion algorithm in [32] which, in turn, is a straightforward extension of the single-agent decision tree algorithm found in [44].

As indicated, the incentive for us to actually use the sequential method when developing the information fusion game component has so far been limited, but the relation between the sequential method and its potential savings must be kept in mind when developing the game component further. A model incorporating a series of ordered decisions, or perhaps a hierarchy of decisions as outlined in [7], is likely to benefit significantly from this representation. More information on this topic regarding so-called MAIDs, an acronym for multi-agent influence diagrams, and their relation to the information fusion game component can be found in Section 10.

Although game-theoretic methods are, in most cases, computationally infeasible in theory, computation of optimal solutions still seems to be tractable in reasonably sized C2 decision problems [8]. Moreover, despite the intractability of finding all optimal solutions there exist fast algorithms that often finds all, or nearly all, solutions.

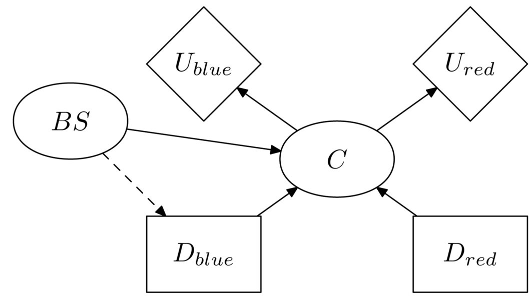

Fig. 5. Influence diagram depicting an example scenario with a blue player and a red player. The Boolean node BS denotes the blue player's private information that gives rise to two blue player types in the game.

8. A SMALL EXAMPLE

In this section the gaming perspective is illustrated with an example of a situation where the commander wishes to reason about two possible models.

At a certain point in battle, a blue (male) unit controls an asset (equipment or territory). When a red (female) unit appears on the scene the blue unit knows immediately whether its own forces are inferior or superior. The red unit on the other hand, does not know anything regarding the capabilities of the blue unit. The blue unit has the choice to engage in battle or to remain passive. If he remains passive the red unit will use her sensors to detect whether he is superior or not and if he is inferior she will force him to give up the asset. On the other hand, if the blue unit chooses to engage the red unit she will be faced with an opportunity to retreat or to engage. If the blue unit is superior and the red unit chooses to engage him, he will both defeat the red unit and keep control of the asset. If the blue unit is inferior and the red unit chooses to engage him he will lose both the battle and the asset. If the red unit retreats the blue unit will keep control of the asset whether he is superior or not. The central part of the corresponding influence diagram is shown in Fig. 5. The random variable *BS* (Blue Superior) constitutes evidence for the blue decision-maker but not for the red decision-maker, denoted with the dotted arrow from *BS* to D_{blue}. The node *BS* is also a parent to the world consequence node *C* because it determines the outcome of an engagement and thus the state of the world. The *C* node then affects the decision-makers' respective utility nodes where, in this case, $U_{\text{blue}} = -U_{\text{red}}$ since the game is zero-sum. It is vital to understand the difference between evidence variables and query variables to fully grasp the example (and the game component as a whole). For the blue player, the variable *BS* is evidence which, in turn, gives rise to one "blue superior game model" and one "blue inferior game model." For the red player, *BS* is just an ordinary random variable with an associated conditional probability table. The chance node *C*, on the other hand, can never have its value set as an evidence variable as it is referring to a future state.

TABLE I
Payoff Matrices for the Myerson Card Game

t_1 = 1.a (superior)

Player 1 \ Player 2	M	P
R	2, −2	1, −1
F	1, −1	1, −1

t_1 = 1.b (inferior)

Player 1 \ Player 2	M	P
R	−2, 2	1, −1
F	−1, 1	−1, 1

If the value of 1) winning the battle and 2) controlling the asset are worth one utility unit respectively, the game becomes similar to the card game of Myerson [41]. As indicated in the situation description, we follow the convention that odd-numbered players are male and even-numbered players are female. This is common practice in game theory and has no deeper meaning. At the beginning of the game both players put a dollar (the asset) in the pot. Player 1 (the blue force) looks at a card from a shuffled deck which may be red (he is superior) or black (he is inferior). Player 2 (the red force), on the other hand, does not know the color of the card but maintains a belief of this in the form of a probability distribution in her influence diagram, i.e., a belief of the possibility of player 1 being superior or inferior. Player 1 moves first and has the opportunity to fold (F) or to raise (R) with another dollar, i.e., remain passive or engage in battle. If he raises, player 2 has the opportunity to pass (P) or to meet (M) with another dollar in the pot, i.e., retreat or engage in battle.

We let $\alpha \in (0,1)$ denote player 2's belief of player 1 being superior. In this example, player 1 also knows the value of α, i.e., the players' beliefs are consistent. The situation can then be modeled with a Bayesian game Γ^{b}, as defined in (1), with $N = \{1,2\}$, $C_1 = \{F,R\}$, $C_2 = \{M,P\}$, $T_1 = \{1.\mathrm{a}, 1.\mathrm{b}\}$, $T_2 = \{2\}$, $p_1(2 \mid 1.\mathrm{a}) = p_1(2 \mid 1.\mathrm{b}) = 1$, $p_2(1.\mathrm{a} \mid 2) = \alpha$, $p_2(1.\mathrm{b} \mid 2) = 1-\alpha$ and $(u_1(c_1,c_2,t_1), u_2(c_1,c_2,t_1))$ as in Table I.

Solving the game using the technique described by Harsanyi [21] involves introducing a historical chance node, a "move of nature," that determines player 1's type, hence transforming player 2's incomplete information regarding player 1 into imperfect information. The Bayesian equilibrium of the game is then precisely the Nash equilibrium of this imperfect information game. The Harsanyi transformation of Γ^{b} is depicted in Fig. 6 on extensive form.

Note that there are two decision nodes denoted "2.0" that belong to the same information set, representing the uncertainty of player 2 regarding player 1's type. Also, note that the move labels on the branch following the "1.a" node do not match the move labels on the branches following the "1.b" node, representing that player 1 is able to distinguish between these two nodes. The normal way of solving such a game is to look at the strategic representation, as seen in Table II.

In order to solve the game, first note that Fr is dominated by Rr and that Ff is dominated by Rf

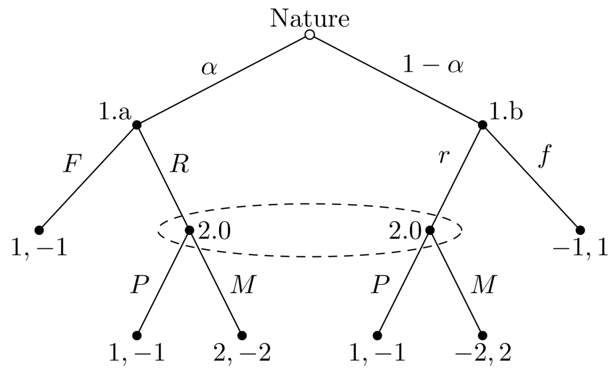

Fig. 6. The Harsanyi transformation of the game in Table I.

TABLE II
The Strategic Form of the Game in Fig. 6

Player 1 \ Player 2	M	P
Rr	$4\alpha-2, 2-4\alpha$	$1,-1$
Rf	$3\alpha-1, 1-3\alpha$	$2\alpha-1, 1-2\alpha$
Fr	$3\alpha-2, 2-3\alpha$	$1,-1$
Ff	$2\alpha-1, 1-2\alpha$	$2\alpha-1, 1-2\alpha$

regardless of the value of α, i.e., player 1 will always raise if in a superior position. Second, if $3/4 \le \alpha < 1$ we have that P dominates M so that player 2 will always choose to pass, which, in turn, implies that player 1 will always choose to raise. Hence, $([Rr],[P])$ is the one and only equilibrium strategy profile for $3/4 \le \alpha < 1$. For $0 < \alpha < 3/4$ there are no equilibria in pure strategies (just check all four remaining possibilities) and we have to look for equilibria in mixed strategies. Let $q[Rr] + (1-q)[Rf]$ and $s[M] + (1-s)[P]$ denote the equilibrium strategies for players 1 and 2 respectively, where q denotes the probability that player 1 raises with a losing card and s the probability that player 2 meets if player 1 raises. A requirement for an equilibrium for player 1 is that his expected payoff is the same for both Rr and Rf, i.e., $s(4\alpha-2) + (1-s)1 = s(3\alpha-1) + (1-s)(2\alpha-1) \Rightarrow s = 2/3$. Similarly, to make player 2 willing to randomize between M and P, M and P must give her the same expected utility against $q[Rr] + (1-q)[Rf]$ so that $q(4\alpha-2) + (1-q)(3\alpha-1) = q1 + (1-q)(2\alpha-1) \Rightarrow q = -\alpha/(3(\alpha-1))$.

We can now use the equilibrium strategy of the imperfect information game in order to derive the Bayesian equilibrium of the game Γ^{b}. A Bayesian equilibrium specifies a randomized strategy profile containing one strategy $\sigma_i(\cdot \mid t_i)$ for all combinations of players and types. Hence, the unique Bayesian equilibrium of the game Γ^{b} is $\sigma_1(\cdot \mid 1.\mathrm{a}) = [R]$, $\sigma_1(\cdot \mid 1.\mathrm{b}) = q[R] + (1-q)[F]$, $\sigma_2(\cdot \mid 2) = 2/3[M] + 1/3[P]$ for $0 < \alpha < 3/4$ and $\sigma_1(\cdot \mid 1.\mathrm{a}) = [R]$, $\sigma_1(\cdot \mid 1.\mathrm{b}) = [R]$, $\sigma_2(\cdot \mid 2) = [P]$ for $3/4 \le \alpha < 1$.

Although this simple game presents a solution that is not entirely trivial, it is simpler than our full family of games in that it is zero-sum with only two players and thus has a unique Nash equilibrium that is computationally easy to find.

9. SOLUTION INTERPRETATION

Nash equilibria, in the form of mixed strategies, as a solution to decision problems require a moment of thought. On the one hand, it is easy to argue that the equilibrium strategy is theoretically sensible. After all, the notion of Nash equilibria, building on the concept of rationality, defines precisely this. By using the idea of Bayesian games we are able to create alternative models regarding agents that are in some way "irrational." Thus, by using Bayesian games we can counterattack any objections on the existing model by simply extending the model with a new submodel that models the objection in question. Of course, this also requires assigning a prior probability to the new submodel and re-evaluating the prior probabilities for the existing submodels, which makes sense if someone comes up with an objection (which is interpreted as a new model that we have not thought of before). If the objection is independent of the existing models, normalization is the natural way to re-assign probabilities. Otherwise it is natural to let the prior probability of the new model be represented by a reduction of prior probabilities of the model or the models that it depends on. In most cases we believe that it is appropriate to have a separate model for the "uncertain case" that takes care of whatever we have not thought of. In that case the new submodel, provided it is independent of other existing models, typically reduces our overall uncertainty regarding the situation and thus causes a reduction of prior probability for the earlier mentioned "uncertain case" submodel. Models that take care of the rest, i.e., that represent options or possibilities that we are not yet aware of, are often found in proposed architectures for multi-agent modeling, see for example [20] where irrational behavior as well as lack of information is modeled in so called "no information models."

On the other hand, although representing the theoretically rational course of action, the Nash equilibrium poses several concerns regarding its interpretation. Looking at the example scenario in Section 8, it is interesting to see how q and s varies depending on α which is shown in the diagram in Fig. 7, i.e., how the solution to our decision problem varies depending on our subjective beliefs regarding the opponent being superior or inferior. How do we convince a commander that he should decide what to do by throwing a die that varies depending on $q(\alpha)$? He probably understands that he is bluffing, and that it is in general disadvantageous both to always bluff and to never bluff. Without knowing

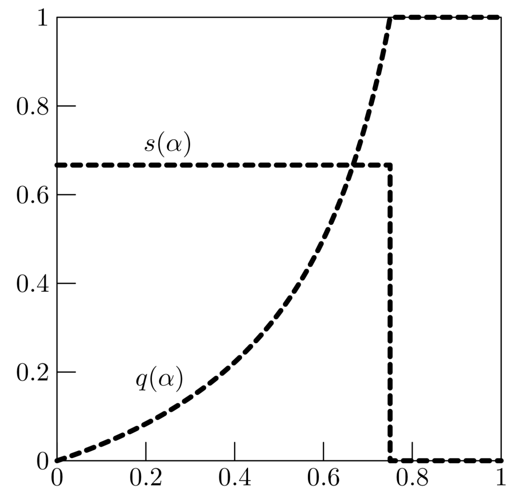

Fig. 7. The graph shows how the game-theoretic solution $s(\alpha)$ to the decision problem in Section 8 varies in a non-intuitive manner depending on the player's speculation regarding the other player being inferior or superior.

the background to the solution it is not trivial to understand why player 1 should raise with a losing card with probability $q(\alpha)$ in Fig. 7. Perhaps even more strange is that player 2's counterattack, the probability $s(\alpha)$ to meet, is kept constant at $s(\alpha) = 2/3$ until $\alpha = 3/4$ when it suddenly goes down to zero. So there is a discontinuity in the optimal strategy when α varies, although at the discontinuity the optimal utilities vary continuously. Hence, an error in the α estimate has no large utility effect although the equilibrium solution strategies may vary significantly. The conclusion regarding the Myerson card game is that a simple problem gives us a solution that is difficult to understand intuitively and that may or may not, dependent upon the decision-maker's objective, raise questions regarding robustness. This is quite typical, see for example [6] for another example, and we need to address the question of how to use the solution in a sensible way. To actually throw the die is part of the solution and if this is not performed the commander is not rational and, hence, will be outperformed by a rational opponent that is capable of modeling this behavior. It is probably easier to accept the opponent's randomized strategy as a prediction. Then the optimality of one's own randomized strategy is fairly easy to establish. As can be seen in Fig. 7, however, such a prediction must be analyzed for discontinuities that indicate potential issues related to strategy robustness.

To outperform someone by exploiting his plan is called outguessing. It is tempting to use an estimate of the risk of exploitation as a basis for decision-making so that the (risk-compensating) Nash equilibrium mixed strategy is chosen when the risk is high and the pure strategy with the highest payoff is chosen when the risk is low. An approach in this direction using hypergame theory, which is fundamentally heretical to the concepts of game theory, is proposed in [53].

10. RELATED WORK

Development of game tools is an active area in AI. In the Gala system of [33], tools exist for defining games with imperfect information. A tractable way to handle games with recursive interaction in strategic form was developed in [20], where the potentially infinite recursion of beliefs about opponents is represented approximately as a finite depth discrete utility/probability matrix tree defining the players' beliefs about each other. The solution emerging from this modeling is not a Bayesian game equilibrium, however.

There is a significant body of work on multi-agent interactions in the intelligent agents literature. A survey of methodological and philosophical problems appears in [20]. The principle of bounded rationality can be taken as an excuse to use simpler solution concepts than Bayesian game solutions. In our case, there is no reason to assume that the opponent is not rational—there would be few excuses if he turned out to be so. This does not mean that it is not necessary to take advantage of opponents' mistakes when they occur. Plans must foresee this and have opportunities of opponent mistakes as a part; but these options should not be executed until the evidence of the mistake is sufficient. The recursive modeling of multi-agent interaction of [20] (mostly developed for cooperative rather than competitive interaction) is thus not appropriate in our application. The proposal in [27] is to use game theory with zero-sum game tree look-ahead for C2 applications. Although this approach was successful for analysis of lower level game situations, we have argued above that it is not enough in a complete higher level C2 tool.

In [32] the concept of a multi-agent influence diagram (MAID) is defined, which in a similar manner to our information fusion game component partitions the decision and utility variables by agent so that utilities and decisions of many agents can be described. The key idea behind the MAID framework is to use the graph structure to explicitly state strategic relevance between decision variables which, in turn, is being used to break up a large game into a set of singly connected components (SCCs) which can be solved in sequence. The complexity of equilibria computation in the full game is therefore reduced to the complexity of equilibria computation in the largest SCC in the MAID. In some games, where the maximal size of an SCC is much smaller than the total number of decision variables, the MAID representation provides exponential savings over existing solution algorithms. In the worst case, however, the strategic relevance graph forms a single large SCC and the MAID algorithm simply solves the game in its entirety, with no computational benefits. The influence diagrams in the information fusion game component outlined in Section 6 are unfortunately examples of such large SCCs. As it turns out, the whole game component could be alternatively represented by a MAID with a single large SCC provided an additional chance node, representing the "move of nature," was added to connect the models to each other.

An extension of the MAID framework is the NID—Network of Influence Diagrams. In the version described in [18], several MAIDs—or other game representations—can be connected in a directed acyclic graph, where outgoing arcs are labeled with a probability distribution. This allows us to define situations where agents do not all use the same model, but there is no way to describe in an acyclic graph a situation where there is mutual uncertainty and inconsistent beliefs about the game structure and the opponents' goals.

11. CONCLUSIONS

In higher level command and control (C2) we can be certain that large efforts are directed towards predicting the beliefs, desires, and intentions of the adversary—and there will not be a common agreed upon model of the situation and its utilities. In fact, the complex nature of any C2 decision situation makes it necessary to go beyond any proposed theoretical model and question how, if at all, it can be used in practice. Adding conflict, where opposing parties try to outguess each other, complicates things even further with the necessary addition of a gaming perspective—putting stress on a decision situation that is complex already from the beginning.

In this paper we propose a way to overcome the barriers between theory and practice, taking into account opponent modeling as well as current state-of-the-art C2 situation modeling principles. We characterize the proposed architecture as an information fusion game component to emphasize the inherent dependencies between the gaming perspective and the process of fusing sensor data into a comprehensible situation picture. It is our belief that game theory should not be considered just another tool in the decision-maker's toolbox. Rather, it is the science of agent interaction itself, i.e., we consider game theory to be the whole toolbox as well as a statement of the information fusion threat prediction problem.

Game-theoretic tools have a potential for situation prediction that takes uncertainties in enemy plans and deception possibilities into consideration. The idea behind Bayesian games is particularly interesting, and needed, from the viewpoint of a commander facing a real setting decision problem; it combines several models of the situation, thus making it possible to consider such diverse factors as opponent irrationality or the decision-maker's intuition by incorporating these ideas as separate models. However, Bayesian games, as well as game theory in general, still have shortcomings when representing realistic, potentially large and complex, situation descriptions—at least compared to the expressiveness and ease of understanding obtained with the current state-of-the-art single agent description within AI, i.e., a Bayesian network representation of the situation. Hence, the natural extension in order to make the Bayesian game truly useful for other problems than leisure games is to maintain several influence diagram representations of the possible models and let the game's utility functions consist of the utilities that can be calculated with the use of the respective influence diagrams.

For a situation picture to be truly useful for a commander, it should convey both awareness of the current situation as well as predictive awareness regarding likely future courses of events. Hence, prediction of future courses of events must be considered of utmost importance when commencing development of the next generation's C2 systems and, henceforth, in higher level fusion the game component is both important and needed.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the help of Per Svensson at the Swedish Defence Research Agency for commenting on drafts at different stages.

REFERENCES

[1] D. S. Alberts
Network centric warfare: Current status and way ahead.
Journal of Defence Science, **8**, 3 (Sept. 2003), 117–119.

[2] D. S. Alberts, J. J. Garstka and F. P. Stein
Network Centric Warfare: Developing and Leveraging Information Superiority.
Washington, D.C.: CCRP, DoD, 2nd ed., 1999.

[3] S. Arnborg, H. Artman, J. Brynielsson and K. Wallenius
Information awareness in command and control: Precision, quality, utility.
In *Proceedings of the Third International Conference on Information Fusion* (FUSION 2000), Paris, France, July 2000, ThB1/25–32.

[4] R. Azoulay-Schwartz and S. Kraus
Negotiation on data allocation in multi-agent environments.
Autonomous Agents and Multi-Agent Systems, **5**, 2 (June 2002), 123–172.

[5] D. Billings, N. Burch, A. Davidson, R. Holte, J. Schaeffer, T. Schauenberg and D. Szafron
Approximating game-theoretic optimal strategies for full-scale poker.
In *Proceedings of the 2003 International Joint Conference on Artificial Intelligence* (IJCAI 2003), 2003, 661–668.

[6] J. Brynielsson
Using AI and games for decision support in command and control.
Decision Support Systems, in press.

[7] J. Brynielsson and S. Arnborg
Bayesian games for threat prediction and situation analysis.
In P. Svensson and J. Schubert (Eds.), *Proceedings of the Seventh International Conference on Information Fusion* (FUSION 2004), Stockholm, Sweden, vol. 2, June 28–July 1, 2004, 1125–1132.

[8] J. Brynielsson and S. Arnborg
Refinements of the command and control game component.
In *Proceedings of the Eighth International Conference on Information Fusion* (FUSION 2005), Philadelphia, PA, July 2005.

[9] J. Brynielsson, M. Engblom, R. Franzén, J. Nordh and L. Voigt
Enhanced situation awareness using random particles.
In *Proceedings of the Tenth International Command and Control Research and Technology Symposium* (ICCRTS), McLean, VA, June 2005.

[10] J. Brynielsson and K. Wallenius
Game environment for command and control operations (GECCO).
In *Proceedings of the First International Workshop on Cognitive Research With Microworlds*, Granada, Spain, Nov. 2001, 85–95.

[11] V. Chvátal
Linear Programming.
New York: W. H. Freeman and Company, 1983.

[12] T. P. Coakley
Command and Control for War and Peace.
Washington, D.C.: National Defense University Press, 1991.

[13] R. W. Cottle, J-S. Pang and R. E. Stone
The Linear Complementarity Problem.
Boston, MA: Academic Press, 1992.

[14] R. G. Cowell, A. P. Dawid, S. L. Lauritzen and D. J. Spiegelhalter
Probabilistic Networks and Expert Systems.
Statistics for Engineering and Information Science, New York: Springer-Verlag, 1999.

[15] G. Devetag and M. Warglien
Playing the wrong game: An experimental analysis of relational complexity and strategic misrepresentation.
CEEL Working Paper 0504, Computable and Experimental Economics Laboratory, Department of Economics, University of Trento, Italy, 2005.

[16] E. Durschmied
The Hinge Factor: How Chance and Stupidity Have Changed History.
London: Coronet Books, 1999.

[17] H. Friman and B. Brehmer
Using microworlds to study intuitive battle dynamics: A concept for the future.
In *Proceedings of Command and Control Research and Technology Symposium* (CCRTS), Newport, RI, June 29–July 1, 1999.

[18] Y. Gal and A. Pfeffer
A language for modeling agents' decision making processes in games.
In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems* (AAMAS'03), Melbourne, Australia, July 2003, 265–272.

[19] I. Gilboa and E. Zemel
Nash and correlated equilibria: Some complexity considerations.
Games and Economic Behavior, **1**, 1 (Mar. 1989), 80–93.

[20] P. J. Gmytrasiewicz and E. H. Durfee
Rational coordination in multi-agent environments.
Autonomous Agents and Multi-Agent Systems, **3**, 4 (Dec. 2000), 319–350.

[21] J. C. Harsanyi
Games with incomplete information played by "Bayesian" players.
Management Science, **14**, 3,5,7 (1967–1968), 159–182, 320–334, 486–502.

[22] G. E. Horne
Maneuver warfare distillations: Essence not verisimilitude.
In P. A. Farrington, H. B. Nembhard, D. T. Sturrock and G. W. Evans (Eds.), *Proceedings of the 1999 Winter Simulation Conference*, 1999.

[23] E. Horvitz, J. Breese, D. Heckerman, D. Hovel and K. Rommelse
The Lumière project: Bayesian user modeling for inferring the goals and needs of software users.
In G. F. Cooper and S. Moral (Eds.), *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence* (UAI-98), Madison, WI, July 1998, 256–265.

[24] R. A. Howard and J. E. Matheson
Influence diagrams.
In R. A. Howard and J. E. Matheson (Eds.), *Readings on the Principles and Applications of Decision Analysis*, Menlo Park, CA: Strategic Decisions Group, 1984, 721–762.

[25] R. Johansson and R. Suzić
Particle filter-based information acquisition for robust plan recognition.
In *Proceedings of the Eighth International Conference on Information Fusion* (FUSION 2005), Philadelphia, PA, July 2005.

[26] R. Johansson, N. Xiong and H. I. Christensen
A game theoretic model for management of mobile sensors.
In *Proceedings of the Sixth International Conference on Information Fusion* (FUSION 2003), Cairns, Australia, July 2003, 583–590.

[27] A. Katz and B. Butler
"Game commander"—Applying an architecture of game theory and tree lookahead to the command and control process.
In *Proceedings of the Fifth Annual Conference on AI, Simulation, and Planning in High Autonomy Systems* (AIS94), Gainesville, FL, 1994.

[28] G. A. Klein
Recognition-primed decisions.
In W. B. Rouse (Ed.), *Advances in Man-Machine Systems Research*, Greenwich, UK: JAI Press, 1989, 47–92.

[29] D. E. Knuth and R. W. Moore
An analysis of alpha-beta pruning.
Artificial Intelligence, **6**, 4 (Winter 1975), 293–326.

[30] D. Koller, N. Megiddo and B. von Stengel
Fast algorithms for finding randomized strategies in game trees.
In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing* (STOC), Montréal, Québec, Canada, May 1994, 750–759.

[31] D. Koller, N. Megiddo and B. von Stengel
Efficient computation of equilibria for extensive two-person games.
Games and Economic Behavior, **14**, 2 (June 1996), 247–259.

[32] D. Koller and B. Milch
Multi-agent influence diagrams for representing and solving games.
Games and Economic Behavior, **45**, 1 (Oct. 2003), 181–221.

[33] D. Koller and A. Pfeffer
Representations and solutions for game-theoretic problems.
Artificial Intelligence, **94**, 1 (July 1997), 167–215.

[34] D. M. Kreps
Game Theory and Economic Modelling.
Clarendon Lectures in Economics, Oxford University Press, 1990.

[35] J. Kuylenstierna, J. Rydmark and H. Sandström
Some research results obtained with DKE: A dynamic wargame for experiments.
In *Proceedings of the Ninth International Command and Control Research and Technology Symposium* (ICCRTS), Copenhagen, Denmark, Sept. 2004.

[36] C. E. Lemke and J. T. Howson, Jr.
Equilibrium points of bimatrix games.
Journal of the Society for Industrial and Applied Mathematics, **12**, 2, (June 1964), 413–423.

[37] J. Llinas, C. Bowman, G. Rogova, A. Steinberg, E. Waltz and F. E. White, Jr.
Revisiting the JDL data fusion model II.
In P. Svensson and J. Schubert (Eds.), *Proceedings of the Seventh International Conference on Information Fusion* (FUSION 2004), Stockholm, Sweden, vol. 2, June 28–July 1, 2004, 1218–1230.

[38] J-F. Mertens and S. Zamir
Formulation of Bayesian analysis for games with incomplete information.
International Journal of Game Theory, **14**, 1 (Mar. 1985), 1–29.

[39] J. Moffat
Complexity Theory and Network Centric Warfare.
Washington, D.C.: CCRP, DoD, 2003.

[40] K. P. Murphy
Dynamic Bayesian Networks: Representation, Inference and Learning.
Ph.D. thesis, University of California, Berkeley, 2002.

[41] R. B. Myerson
Game Theory: Analysis of Conflict.
Cambridge, MA: Harvard University Press, 1991.

[42] J. F. Nash
Non-cooperative games.
Annals of Mathematics, **54**, 2 (Sept. 1951), 286–295.

[43] S. Parsons and M. Wooldridge
Game theory and decision theory in multi-agent systems.
Autonomous Agents and Multi-Agent Systems, **5**, 3 (Sept. 2002), 243–254.

[44] J. Pearl
Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.
San Mateo, CA: Morgan Kaufmann Publishers, Inc., 1988.

[45] S. J. Russell and P. Norvig
Artifical Intelligence: A Modern Approach.
Upper Saddle River, NJ: Prentice Hall, 2nd ed., 2003.

[46] R. D. Shachter
Evaluating influence diagrams.
Operations Research, **34**, 6 (Nov.–Dec. 1986), 871–882.

[47] A. Steinberg, C. Bowman and F. E. White, Jr.
Revisions to the JDL data fusion model.
In *Proceedings of SPIE AeroSense* (Sensor Fusion: Architectures, Algorithms, and Applications III), Orlando, FL, vol. 3719, Apr. 1999, 430–441.

[48] B. von Stengel
Efficient computation of behavior strategies.
Games and Economic Behavior, **14**, 2 (June 1996), 220–246.

[49] B. von Stengel
Computing equilibria for two-person games.
In R. J. Aumann and S. Hart (Eds.), *Handbook of Game Theory with Economic Applications*, Elsevier Science, vol. 3 of Handbooks in Economics, 2002, 1723–1759.

[50] R. Suzić
Representation and recognition of uncertain enemy policies using statistical models.
In *Proceedings of the NATO RTO Symposium on Military Data and Information Fusion*, Prague, Czech Republic, Oct. 2003.

[51] P. Thunholm
Military Decision Making and Planning: Towards a New Prescriptive Model.
Ph.D. thesis, Department of Psychology, Stockholm University, Sweden, June 2003.

[52] U.S. Army
Intelligence Preparation of the Battlefield.
Field Manual FM 34–130, Headquarters, Dept. of the Army, Washington, D.C., July 1994.

[53] R. Vane and P. Lehner
Using hypergames to increase planned payoff and reduce risk.
Autonomous Agents and Multi-Agent Systems, **5**, 3 (Sept. 2002), 365–380.

[54] K. Virtanen, J. Karelahti and T. Raivio
Modeling air combat by a moving horizon influence diagram game.
Journal of Guidance, Control, and Dynamics, **29**, 5 (Sept.–Oct. 2006), 1080–1091.

[55] K. Wallenius
Support for situation awareness in command and control.
In P. Svensson and J. Schubert (Eds.), *Proceedings of the Seventh International Conference on Information Fusion* (FUSION 2004), Stockholm, Sweden, vol. 2, June 28–July 1, 2004, 1117–1124.

[56] F. E. White, Jr.
A model for data fusion.
In *Proceedings of the First National Symposium on Sensor Fusion*, Orlando, FL, vol. 2, Apr. 1988.

[57] N. Xiong and P. Svensson
Multi-sensor management for information fusion: Issues and approaches.
Information Fusion, **3**, 2 (June 2002), 163–186.

Joel Brynielsson was born in 1974 in Stockholm, Sweden, where he earned his M.Sc. degree in computer engineering and his Ph.D. degree in computer science, both from the Royal Institute of Technology. He is currently an assistant professor at the Swedish National Defence College.

His research interests are in information and uncertainty management in command and control systems with papers appearing in journals and conferences devoted to information fusion, decision support, command and control, operations research, microworld research, and modeling and simulation.

Stefan Arnborg took the civilingenjör (M.Sc.E.) exam in engineering physics at the Royal Institute of Technology (KTH) in 1968, and the Ph.D. in information processing at KTH in 1972.

He worked as a supercomputer programmer at the Courant Institute (1966), with object-oriented programming systems at the Norwegian Computing Centre (1969–1970) and CAP Sweden (1971), as an operations research analyst at the Swedish Defence Research Establishment (FOA, 1972–1979), and as a distributed systems and computer guru at Philips Financial Terminal Systems (PEAB, 1979–1982). He spent sabbatical terms at the University of Oregon (1986, 1989, and 1993) and was part-time advisor to the Swedish Institute of Computer Science (SICS). He is the director of the theory group and of the M.Sc.E. program in computer engineering. He has been a professor of computer science at KTH since 1982.

His research interests are algorithms and complexity, verification, computer algebra, data engineering and uncertainty management, data mining, human brain informatics, command and control, and aesthetics in engineering education.