

# Implementation and integration of a Bayesian Network for prediction of tactical intention into a ground target simulator

Fredrik Johansson  
School of Humanities and Informatics  
University of Skövde  
Sweden  
fredrik.johansson@his.se

Göran Falkman  
School of Humanities and Informatics  
University of Skövde  
Sweden  
goran.falkman@his.se

*Abstract – Prediction of the enemy’s intention is a main issue of threat analysis, and, hence, will be an important part of the C2-systems of tomorrow. A technique that can be useful for this kind of predictions is Bayesian Networks (BNs). We have developed a BN for prediction of the enemy’s tactical intention, and the implemented BN has been integrated into a ground target simulation framework. The general problem of how to find appropriate prior distributions for BNs has been addressed by developing a tool for data collection, which may make it easier to come up with appropriate prior distributions, by learning conditional probability tables from collected cases, i.e. parameter learning.*

**Keywords:** Bayesian Networks, conditional probability tables, impact assessment, parameter learning, threat analysis.

## 1. Introduction

The Swedish Armed Forces, as well as many other armed forces around the world, are moving towards a network based defense, adapted from the US concept of Network Centric Warfare (NCW). Future tasks for the Swedish Armed Forces will not only involve military responsibilities for the Swedish territory, it will also involve international peace keeping missions and support to civilian authorities in case of large crises [1]. The concept of NCW is based on superior situation awareness and decision superiority, which implies that there is an increased need for effective decision support systems in the command and control (C2) domain. Military operations in general are to a high degree dealing with uncertainty (“the fog of war”), and the new kinds of missions are no exception. In international operations, it is not unusual with very limited information about the enemy, terrain, doctrines etc. This means there is a need for systems which can give decision support under uncertainty.

As stated in [2] (p. 1): “There is a critical need to provide today’s military decision makers with actionable information ... A key requirement to generate this type of actionable information is the ability to predict the adversary’s most likely future

COAs. In today’s asymmetric warfare, such prediction is especially difficult”. Prediction of the enemy’s COAs (Courses of Action) is currently performed manually by intelligence analysts; however, it is becoming increasingly difficult to perform it manually in an accurate and timely fashion [2]. Hence, an important problem that needs more research is how prediction of the enemy’s intention can be more automated.

A technique which can be used for dealing with the uncertainty, present in prediction of the enemy’s intention, is Bayesian Networks (BNs). The usage of BNs to predict the enemy’s intention has been used earlier in e.g. [3,4,5]. In [4], the focus is on prediction of enemy intention by using additional knowledge such as information about the terrain and the enemy’s doctrines in the BN. The BN is used in Matlab-simulations to calculate which company policy (intention) that is most probable, in different scenarios. The approach of using doctrines may be useful in cases where there is a lot of available knowledge about the opponent, but as stated earlier, this is seldom the reality in international operations. In [3], a fuzzy BN is used to decide what the enemy’s intentions are, by investigating information about the unit types that is the output from case-based reasoning in the situation assessment phase.

The usage of BNs for adversary intent modeling has also been investigated in a number of papers written by Santos Jr. et al., e.g. [6,7], where research in user intent inference is presented as a basis for adversarial intent inference.

A potential drawback with the Bayesian approach in general is its sensitivity to the choice of prior distributions, which leads to subjectivity. This also applies to BNs, and, hence, a key problem with BNs is to find ways to put appropriate numbers into the conditional probability tables. There is, as stated earlier, some work done on the usage of BNs for prediction of enemy intention, but the problem of how to fill the conditional probability tables with appropriate values are more or less overlooked. The general approach seems to be assuming that subjective knowledge has

been quantified by experts in “a magical way”. As stated in [5] (p. 26) “... we have advocated that the commander’s perception or subjective knowledge of the situation be exploited to obtain the conditional probabilities, and as pointed out there, this is a problem requiring further research”.

The conditional probabilities are a crucial part, but without having a correct model structure (topology) of the problem domain in question, appropriate conditional probability tables do not help much. The approach to use simple synthetic BNs may be enough for introductory research, but for a real world situation, an expert needs to be involved when the topology of the BN is developed. The main problems addressed by this paper are, therefore, 1) how to find the topology of a BN, reflecting general parameters which influence the enemy’s short-term tactical intention, and 2) how to exploit subjective knowledge in order to learn to quantify conditional probabilities of the BN.

In this paper, a general BN for prediction of the enemy’s tactical intention in the ground combat is described. We describe how the topology of the model has been developed in cooperation with a military expert, and how the BN has been integrated into the simulation framework GTSIM [1], in order to visualize different scenarios where the enemy’s intent is predicted. The integration with the ground target simulator provides a modeling environment which can be used as a tool for data collecting. This tool may facilitate the encoding of expert knowledge into the conditional probability tables, thereby enabling the predictions to be more automated, even though it can not completely remove the subjective expert knowledge part.

We would like to stress that our intention with the developed BN not is to replace the military decision maker or the intelligence analysts. On the contrary, our intention is to develop a tool that supports the decision maker and reduces the cognitive load.

## 2. Bayesian networks

### 2.1 General description

Information fusion in general and the military domain in particular contains a high degree of uncertainty. The basic elements when reasoning under such conditions are random variables. Since it is impossible to efficiently deal with general joint distributions for more than just a few random variables [8], there is a need for alternative methods, such as BNs.

A BN is represented as a directed graph, where each node is annotated with quantitative probability information. According to [9], a BN is given by:

- A set of random variables (either discrete or continuous) that constitutes the nodes of the directed graph.

- A set of directed edges (arrows) that connects pairs of nodes. If there is an edge from node  $X$  to node  $Y$ ,  $X$  is called parent to  $Y$ .
- For every node  $X_i$ , there is a conditional probability distribution that quantifies the effect that any parent nodes have on the node in question.
- The graph is not allowed to have any directed cycles and from this follows that it is a directed acyclic graph (DAG).

This specification implies that a BN consists of two parts: the topology (structure) of the network and the conditional probability distributions. This combination is enough to implicitly specify the full joint distribution for all the variables in the network.

### 2.2 Topology

The topology is the qualitative part of the network. The topology of the BN consists of a set of random variables (the nodes) and a set of directed edges that connects pairs of nodes. To find the right random variables for describing a specific domain is not an easy task; in fact the task demands very good domain knowledge. For this reason there is often domain experts involved when the topologies are developed.

A process for obtaining the graph can be found in [5]. First all variables are examined in order to find out which variables are root causes and which variables are directly influenced by other variables (see Figure 1). All the root causes are assigned a node each and are called level-1 nodes. Then all variables directly influenced by level-1 nodes (and no other nodes) become level-2 nodes. With this done, links are drawn between variables from level-1 to their direct influences on level-2. Then all level-3 nodes are identified, and so on, until all variables have a place in the graph and all dependencies have been accounted for by edges of the graph.

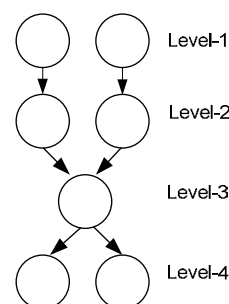


Figure 1: A simple BN illustrating influences between different levels in the network.

All variables have a number of states (at least two) which can be either discrete or continuous. There are also different kinds of variables in a BN, *information variables* and *hypothesis variables*. Information variables are the variables in the network that are directly observable (e.g. data that can be collected from different kinds of sensors). Hypothesis variables on the other hand are variables that are not directly observable

(e.g. enemy intention), and knowledge about their states are inferred from evidence that comes from the information variables.

## 2.3 Conditional probability distributions

The conditional probability distribution is the part of the network where the relationships between different nodes defined in the topology are being quantified. The conditional probability distributions are encoded into the BN by using a set of conditional probability tables, CPTs. Traditionally, CPTs are filled with numbers either by experts estimating the numbers or by setting the numbers based on statistics that are collected from real data in the domain [10]. The level-1 nodes have no parents and therefore their CPTs are not conditional of anything, so in fact they are just prior probabilities. Nodes on the other levels have parents and thereby their conditional tables define how probable the different states of the node are, given the states of their parents.

As stated earlier, it is often difficult to fill the CPTs with appropriate numbers. In cases where there is a large number of data available, the problem can be solved by learning the CPTs from the real existing data, algorithms for this can for example be found in [11]. In the military domain it is often difficult to get access to real data, mainly due to lack of data [12], but also due to restrictions. In these cases, the CPTs must be constructed from subjective expert knowledge.

Recall the simple BN topology from Figure 1 and let the two level-2 nodes be information variables ( $I_1$  and  $I_2$ ) connected to a hypothesis variable on level-3. If the information variables  $I_1$  and  $I_2$  both have three discrete states, the CPT for the hypothesis variable will consist of nine rows, where each row corresponds to a combination of the parents states (see Table 1). This CPT is probably straightforward for an expert to fill with numbers (depending on the domain and which the variables are). The problem is that the number of rows in the CPT for the child node grows exponentially with its number of parents. If all of the  $n$  parents have three states each, the child's CPT will consist of  $3^n$  rows. This implies that if there are six parents with three states each, the child's CPT will consist of  $3^6=729$  rows. To fill in this table by hand is probably a non-trivial task, and the task gets more difficult when the number of states or parents grows even more.

**Table 1: CPT for the level-3 hypothesis variable**

$I_1$	$I_2$	True	False
State1	State1	0.60	0.40
State1	State2	0.55	0.45
State1	State3	0.50	0.50
State2	State1	0.30	0.70
State2	State2	0.50	0.50
State2	State3	0.45	0.55
State3	State1	0.35	0.65
State3	State2	0.40	0.60
State3	State3	0.50	0.50

There are techniques such as the Noisy-OR gate [11], which in some cases can be used to reduce these exponentially problems into linearly problems. Another approach that has been used successfully in the military C2-domain is the weighted sum algorithm [12,13]. These kinds of techniques are invaluable where they can be used, but they still need to be complemented with tools that simplify the collecting of data from experts. An example of this is the tool for data collecting described in chapter 4.2.

## 3. A BN for prediction of the enemy's tactical intention

### 3.1 Parameters and topology

An open interview has been performed with an officer from the Swedish Army Combat School (Markstridsskolan) for the purpose of finding general parameters which can be used for prediction of the enemy's tactical intention in different ground combat scenarios. The results from the interview showed that the officer thought it would be difficult to find good general parameters which could be applied to many different scenarios, due to the fact that the Swedish operational defense builds on the foundation of maneuver warfare where *critical vulnerabilities* are very important. The concept of critical vulnerabilities refers to that the own forces should be used towards points where the enemy is weak or where the enemy does not expect attacks. These points, or vulnerabilities, vary from situation to situation, which makes it hard to use them as general parameters; instead they should be identified and added to specific scenarios. However, a number of general parameters were identified from the interview with the officer and from the discussion with the military expert from EMW:

An obvious parameter is the *enemy intention*. It is the posterior probabilities for its states that should be calculated, and thereby, this should be a hypothesis variable in the BN. Another parameter is the *distance* between the enemy and different potential targets. (Note that targets in this paper correspond to own forces, buildings etc. due to the fact that we try to predict what the enemy's intention are, and hence, try to see it from the enemy's perspective). It is also interesting whether the enemy is moving towards a potential target or if he is moving away from it, thus, the temporal parameter *direction* has been included in the model.

There is also the aspect of a target's *protection value*, e.g. an airport will probably have higher protection value than a road block. Furthermore, different types of units are suitable for doing different tasks; hence, the *enemy type* and *target type* are included in the model. Another parameter is the utility the enemy believes will be the outcome of an attack towards a target; this parameter will from now on be defined as a target's *attraction*. The attraction reflects the principle of

Maximum Expected Utility (MEU) and can be seen as a combination of the enemy type, the target type and the target's protection value. Therefore, the above parameters are set as parents to a target's attraction (see Figure 2). The attractions for the different targets are in their turn parents to the enemy intention node. The direction and distance nodes are effects of the enemy intention and, therefore, they have enemy intention as their parent. These nodes, together with the relationships among them, constitute a first simple general topology for prediction of the enemy's tactical intention. Figure 2 shows this BN topology for two un-instantiated targets, but, of course, the BN grows dynamically with the number of targets. Notice that perfect sensors are assumed in this model. In real-world applications where sensors are non-perfect, e.g. enemy type will become an intermediate node, and an information node that reflects the actual sensory output regarding enemy type will be added on top of it.

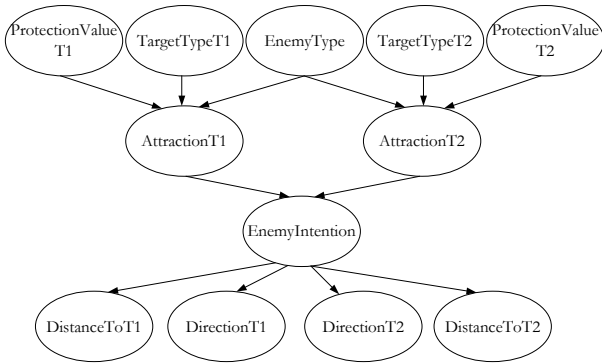


Figure 2: General BN topology for two alternative targets.

### 3.2 Conditional probability distributions

The nodes sets of states are as follows:

- ProtectionValue: {High, Medium, Low}
- TargetType: {Artillery, Maintenance, Infantry}
- EnemyType: {Armor, Mech. Infantry, Infantry}
- Attraction: {High, Medium, Low}
- EnemyIntention: { $T_1$ ,  $T_2$ , ...,  $T_n$ , No Attack}
- Distance: Continuous (discretized into 8 intervals)
- Direction: {Towards, From}

As can be seen in Figure 2, the number of parents to enemy intention grows with the number of alternative targets. If there are  $n$  alternative targets, the CPT for enemy intention will consist of  $3^n$  rows (since all the parents each have three possible states). This highlights the problem with how to come up with appropriate numbers in large CPTs. The approach here has been to constrain the number of possible targets to three. In most real-world cases, the number of hypotheses will, however, be larger than that, and therefore tools and methods must be developed to cope with these kinds of problems. The actual numbers of the CPTs have been set in cooperation with the officer from the Swedish

Army Combat School, but will not be presented here since the actual numbers are not the focus of this paper.

## 4. Adding BN functionality to a ground target simulator

GTSIM (Ground Target Simulator) is a simulation framework developed for use as a test bed for different methods and concepts in the field of information fusion [1]. The source code for GTSIM is written in Java and, therefore, the Netica Java API has been used to add BN functionality into GTSIM. One reason for the integration of the probabilistic model into a ground target simulator is that threat analysis is not a standalone problem in a real-world application; it is based on output from processed sensor data, clustering, etc. Therefore, the simulation environment of GTSIM provides a good test bed for the use of BNs in the military domain. Though, in this case, the most important reason for the integration is that the simulation and visualization of different scenarios makes it easier to collect subjective expert knowledge which can be used to fill the CPTs for the BN with appropriate numbers.

### 4.1 Developing a BN package for the ground target simulator

In the ground target simulator, a scenario generator package is used in order to generate a scenario. By defining waypoints that ground based units or flying sensors should pass, complete routes for the units and sensors are built up. Ability to define coordinates, type and protection value for static targets (such as artillery and infantry units which constitutes different targets that the enemy may attack) have been added to the scenario generator.

One of the key elements in the ground target simulator is the world state simulation which contains a digital terrain database and a scenario based ground target model [1]. Sensor simulations (such as GMTI-radar, SAR-system, etc.) collect data from the world state simulation and generate real-time sensor data which are used to build up a common operational picture, presented by a presentation package. In this paper, the sensors that have been used are simulated to be perfect, in the sense that they can see everything. This reduces the complexity, i.e. uncertainty is only in the future instead of both in the present and the future. However, GTSIM provides the possibility to simulate the uncertainty in the present situation which can be used in future work. The user who sees the common operational picture can choose to add information (type, coordinates and protection value) about the static targets that the enemy may attack into the system. An example of this can be seen in Figure 3 where a red force armour unit (to the right) and three blue force units representing alternative targets (to the left) are visualized.

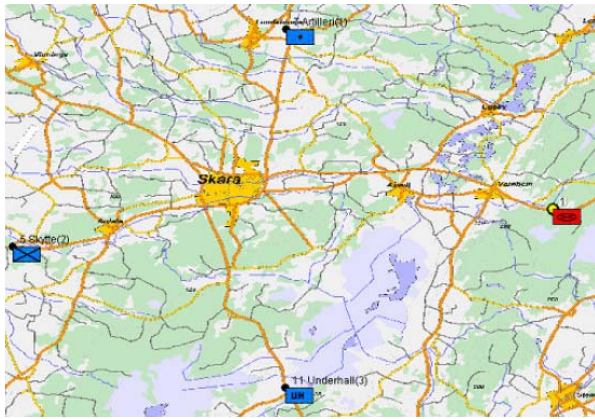


Figure 3: One red force armor unit and three alternative blue force targets (artillery, infantry and maintenance).

A class in the BN package creates a BN out of the common operational picture. The class dynamically reads in the static targets and instantiates them as nodes in the network. The nodes are also used as states for the enemy intention node, together with the state No Attack. The CPTs are then read in and the BN is compiled by another class in the package. After that, findings of protection value, enemy type and target type are set. As stated earlier, the enemy type will be known due to the assumption of perfect sensors, but its states will in a real-world application be uncertain to some degree (the same holds true for the direction and distance nodes). A separate thread is then created, which each fifth second reads in the position of the enemy from the common operational picture, calculates the distance to the different targets and calculates if the enemy unit is moving towards or away from the targets. The results from these calculations are used to draw inferences about how probable the different states of the enemy intention node are. The resulting probabilities are presented to the user in form of a bar chart. Pseudo code for the implementation of the BN can be seen in Figure 4.

```

readInTargets()
constructTopology()
setCPTs()
compileNet()
getBeliefs()
while (not stopped) do
  for i ← 1 to nrOfTargets do
    calculateDistance(EUCLIDIAN, i)
    setDistance(i)
    calculateDirection(i)
    setDirection(i)
  end
  doInference()
  updateChart()
end

```

Figure 4: Pseudo code for the implementation of the BN.

As can be seen in Figure 4, it is possible to calculate the Euclidian distance, but there is also a possibility to calculate the road distance, i.e. use road data from the underlying digital terrain database to calculate the distances.

## 4.2 A data collecting tool for parameter learning

Modifications have been done to the ground target simulator, in order to make it possible to use it as a tool for collection of expert knowledge data. With the modifications, the simulations can be paused automatically in each of the enemy's waypoints (or manually in an arbitrary moment). A dialog box opens, and the user can fill in estimations of the probabilities for the different targets, together with the alternative of no attack.

Information about the current world state (fetched from the common operational picture) is then added as a row in a text file, together with the estimated probabilities for the different targets. In this way, a multi-case file is built up, where each column corresponds to a specific parameter and each row corresponds to a specific case (see Table 2).

Table 2: Example of a part of a case file

Hi	Inf	Med	Inf	Arm	T o	.3	.5	.2
Low	Art	Med	Mai	Mech Inf	F r	.1	.3	.6
Hi	Inf	Low	Art	Inf	T o	.8	.1	.1

By using the support in Netica for parameter learning, the topology in combination with the data in the case files can be used to learn the CPT for each node in the BN. Thereby the learned network can be used to predict the enemy's tactical intention in a new case drawn from a population similar to the cases it learned from.

The data collecting tool may also be used for validating the developed model. By showing simulations of different scenarios for a military expert and letting him or her estimate the probability for the different targets to be attacked by the enemy, the difference between the output from the model and the expert's estimations can be calculated. This difference can be seen as a measure of how correct the model is, i.e. by using a score function such as the mean squared error we can use the expert's estimations as a target value and calculate how good the predictive model is.

During an interview with an officer from the Swedish Army Combat School, he pointed out that it will be necessary to complement the model with the parameter unit size (with states such as platoon, company and battalion) before any model validation can be done. In the current model, all units are of the same size, but the quantitative size is not defined. According to the officer, it is unlikely that the enemy will attack a target if he is not numerically superior to the defender. For this reason, no real model validation has been done; however, a simulation has been shown for another military knowledgeable person. The person's probability estimations for six different points in time were collected and compared with the output from the BN for the same time. No large general conclusions can be drawn from a single scenario, even though,

comparing the model's ordering of the probability for the different targets being attacked with the persons ordering, five out of six waypoints had the same ordering.

## 5. Discussion

It would have been good if recent real-world data could be used for learning the parameters and the topology in the network, but since it is difficult to find that kind of data that we are interested in [12], the approach of using estimated probabilities for different simulated scenarios can be seen as a good alternative. One might argue that it would be better to use historical data from real-world campaigns due to the fact that we can find the answer to what alternative the opponent really chose to attack. The problem with this approach is that modern warfare does not look the same as the warfare did hundred years ago: the means of communication has changed; weapon ranges are longer nowadays, etc. Therefore, the approach of learning from simulated scenarios has been suggested in this paper.

The model described here is just a first simple, general prototype of a BN for prediction of the enemy's intention. Many simplifications have been done, such as the assumption of perfect sensors, the constraint of maximally three targets, all of them must be static and there is just one enemy unit (with an undefined size). However, these simplified assumptions highlight what should be done in future work and the model serves as a ground which can be extended with more situation specific nodes or other general nodes such as unit size. Extensions to the BN are straightforward to implement due to the fact that the work of integrating the BN into the ground target simulator is already done, and thereby just small modifications have to be done, in order to add more nodes to the BN.

The use of BNs has been assumed in this paper; however, there are other approaches such as Dempster-Shafer theory [14,15] which also may be useful for calculating the probability for different enemy courses of action. GTSIM is well suited for comparing different techniques and, hence, it would be interesting to implement a Dempster-Shafer package into GTSIM, with the aim to compare its utility with the implemented BN package.

## 6. Conclusions and future work

In this paper, the possibility to integrate a general BN for prediction of the enemy's intention into a ground target simulator has been shown. BNs rely on subjective probabilities, which are not undisputed. Simulations of different scenarios make it easier to collect experimental data, and therefore, can be used to evaluate the robustness of the implemented model and for validation of the models correctness.

The BN presented in this paper is very general; however, a general model will in a specific situation not

be as useful as a model which is tailor-made for the situation. Therefore, research needs to be done on how to find critical vulnerabilities and how to add them to the BN. There is also a need for increasing the uncertainty by using ordinary sensors instead of the perfect sensors assumed in this paper. Other important aspects are how to deal with a larger number of enemy units and how to generate hypotheses that are used for the hypothesis testing.

Case-based reasoning (CBR) [16,17] is an interesting approach to hypothesis generation and testing (see cf. [18,19,20]). Future work could include the application of CBR techniques to the data in the collected case files in order to both infer initial CPTs from the recorded context and the estimated probabilities, and to identify interesting objects [21] or critical vulnerabilities.

The second part of the implementation, the data collection tool, can also be used for further research. We have argued for the use of a data collection tool for parameter learning, but the practical use of it has not yet been fully evaluated. One important question is how many test cases are needed for parameter learning in the network. In this case, CBR could be used for generating realistic test cases (scenarios), both from actual cases and from prototypical (or stereotypical) cases generated from real cases [22,23], thereby reducing the number of necessary test cases. Another aspect needed to be investigated is how sensitive the model is to small changes of the subjective probabilities in the test cases. If the model is very sensitive to small changes, it can be discussed how useful the particular model actually is.

Before the developed model can be tested and validated thoroughly and in-depth, it has to be complemented with the parameter unit size for both the different targets and the enemy's units. The numbers of the CPTs have so far been rough, and need to be adjusted by e.g. parameter learning or sensitivity analysis. Nevertheless, the output of the implemented model for different scenarios seems promising, since the BN's ordering of threat level for the different targets matches well with a human user's ordering. In the long run, it is the level of threat for the different targets that is important, not the actual numbers, i.e. it does not matter if the probability is 91% or 94% for the enemy attacking a specific target, it is the fact of a very high probability for the target being attacked that matters.

## Acknowledgements

This research has been supported by a grant from the Knowledge Foundation (project number: 2003/0104) to the Information Fusion research program at the University of Skövde ([www.infofusion.se](http://www.infofusion.se)). We would also like to thank Mikael Johannesson, University of Skövde, for his help and supervision of the master's thesis [24], which this paper is based upon.

## References

- [1] H. Warston and H. Persson, *Ground surveillance and fusion of ground target sensor data in a Network Based Defense*, In Proceedings of the 7<sup>th</sup> International Conference on Information Fusion, 1195-1201, 2004.
- [2] A. Pawlowski, S. Gigli and F. Vetesi, *Situation and Threat Refinement Approach for Combating the Asymmetric Threat*, MSS NSSDF Conference, San Diego, CA, 2002.
- [3] C. G. Looney and L. R. Liang, *Cognitive Situation and Threat Assessments of Ground Battlespaces*, Information Fusion, 4:297-308, 2003.
- [4] 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, 2003.
- [5] B. Das, *Representing Uncertainties Using Bayesian Networks*, Information Technology Division Electronics and Surveillance Research Laboratory, 1999.
- [6] B. Bell, E. Santos Jr., S. Brown, *Making Adversary Decision Modeling Tractable with Intent Inference and Information Fusion*, In Proceedings of the 11<sup>th</sup> Conference on Computer Generated Forces and Behavioral Representation, 2002.
- [7] S. Brown, E. Santos Jr., B. Bell, *Knowledge Acquisition for Adversary Course of Action Prediction Models*, Proceedings of the AAAI 2002 Fall Symposium on Intent Inference for Users, Teams, and Adversaries, 2002.
- [8] J. Brynielsson and S. Arnborg, *Refinements of the Command and Control Game Component*, In Proceedings of the 8<sup>th</sup> International Conference on Information Fusion, 2005.
- [9] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2<sup>nd</sup> ed., Prentice Hall, 2003.
- [10] D. Heckerman, *A Tutorial on Learning with Bayesian Networks*, In Learning in Graphical Models, M. Jordan, ed. MIT Press, Cambridge, MA, 1999.
- [11] R. Neapolitan, *Learning Bayesian Networks*, Prentice Hall, 2004.
- [12] L. Falzon and J. Priest, *The Centre of Gravity Network Effects Tool: Probabilistic Modelling for Operational Planning*, Command and Control Division Information Sciences Laboratory, 2004.
- [13] B. Das, *Generating Conditional Probabilities for Bayesian Networks: Easing the Knowledge Acquisition Problem*, <http://arxiv.org/abs/cs.AI/0411034> (last accessed April 25, 2006).
- [14] G. Shafer, *A Mathematical Theory of Evidence*, Princeton University Press, 1976.
- [15] P. Smets and R. Kennes, *The transferable belief model*, Artificial Intelligence, 66:191-234, 1994.
- [16] C. K. Riesbeck and R. C. Schank, *Inside Case-Based Reasoning*, Lawrence Erlbaum, 1989.
- [17] J. L. Kolodner, *An introduction to case-based reasoning*, Artificial Intelligence Review 6(1):3-34, 1992.
- [18] C. G. Looney and L. R. Liang, *Battlespace situation assessment via clustering and case-based reasoning*, In Proceedings of the 17<sup>th</sup> International Conference on Computers and Their Applications, 172-175, 2002.
- [19] B. Yu, K. Sycara, J. Giampapa and S. Owens, *Uncertain information fusion for force aggregation and classification in airborne sensor networks*, In Proceedings of AAAI-04 Workshop on Sensor Networks, July 25-26, 2004.
- [20] R. Grinton, S. Owens, J. Giampapa, K. Syara, C. Grindle and M. Lewis, *Terrain-based information fusion and inference*, In Proceedings of the 7<sup>th</sup> International Conference on Information Fusion, 2004.
- [21] S. H. Liao, *Case-based decision support system: Architecture for simulating military command and control*, European Journal of Operational Research 123(3), 558-567, 2000.
- [22] J. Dowell, *Generative case structure for training scenarios*, In Proceedings of IEEE Conference on Human Interfaces in Control Rooms, Cockpits and Command Centres, 154-158, 2001.
- [23] K. Uchiyama, Y. Iwai, T. Ichinoseki and K. Kayama, *Applying artificial intelligence theory to helicopter SAF simulation based on HLA/RTI(1)*, In Proceedings of Computational Intelligence and Multimedia Applications, 210-214, 2001.
- [24] F. Johansson, *Prediktering av fiendeintention, baserat på bayesiansk hypotesprövning*, Master's thesis, HS-IKI-MD-05-302, University of Skövde, 2005.