14th International Conference on Information Fusion
Chicago, Illinois, USA, July 5-8, 2011

Situation and Threat Assessment for Urban Scenarios in a Distributed Adaptive System

S.P. van den Broek, P. Hanckmann, and M. Ditzel
TNO
The Hague, The Netherlands
Email: bas.vandenbroek@tno.nl

***Abstract*—An approach for decision support, consisting of situation and threat assessment is described, as part of a distributed and adaptive multi-sensor fusion engine. The decision support module combines the management and assessment components for situations and threats. The situation assessment determines relations between observed and geographical objects. Threat assessment relates the situations and other object information to threats. The threat assessment approach is based on a rule based expert system, implemented in one or more Bayesian networks. Results are presented for an example scenario in an urban environment.**

***Index Terms*—fusion engine; situation assessment; threat assessment; decision support**

I. Introduction

It is increasingly difficult to make efficient use of all information coming from the multitude of sensors and sensor types that are deployed during today's military operations. As a result, information analysts have a difficult task filtering through all this information, which for them is actually loosely structured data, at best [1]. To support the information analysts, as well as the decision making process in general, the higher JDL levels [2]–[4] of information processing are receiving an increasing amount of attention [5]. The focus of the research in this area is how to fuse different types of data and how to present the data to the decision maker, creating improved situation awareness. Given the sheer volume of the available data, automated processing of the data to generate sensible and useful information for a decision maker is inevitable. This also requires automated processing at the situation and impact levels of JDL.

At the end of 2006, the European Defence Agency launched a large research program on Force Protection, to improve the protection of the European armed forces against threats troops are facing in urban environments. This program targets research and technology goals in five capability areas, one of which specifically addresses "data analysis including data fusion from various sources" [6] to improve force survivability by increased situational awareness. Part of the Force Protection program is the project DAFNE: *Distributed and Adaptive multi-sensor FusioN Engine*. The DAFNE project aims to demonstrate the advantages of fusing data of multiple heterogeneous sensors combined with intelligence sources over single-sensor operations in an urban environment. Integral part of the project is the development and implementation of a multi-target, multi-sensor and multi-platform fusion engine, capable of mission dependent tracking, classification, and situation and threat assessment.

The overall architecture of the fusion engine, describing the design goals guiding the system design, the individual modules, their roles and their interfacing is described in [7] and [8]. This paper focuses on the decision support module that is part of this fusion engine. The decision support module contains the functionality to implement automated situation and threat assessment in the fusion engine. The fusion engine developed in the project will process simulated sensor data, created by a simulator capable of running extensive scenarios [9]. For clarity, in this paper less complex sensor data will be used, generated by the simulation of a simple scenario.

This paper is organized as follows. In Section II the functionality offered by a decision support module is discussed. The decision support module combines the management and assessment components for situation and impact level. The general approach taken in the DAFNE project is described as part of the fusion engine in Section III. The situation assessment determines relations between observed and geographical objects and events and will be explained in more detail in Section IV. Threat assessment, which relates the situations and other object information to threats, is described in Section V. In Section VI an example of the decision support is shown using a simple scenario in an urban environment. Finally, in Section VII we conclude the paper.

II. Decision support

Decision support aims at presenting decision makers with structured and coherent information. Originally the focus was on detecting, merging, and presenting information about objects and events. This type of information, which is mostly at the JDL levels 0 and 1, helps the operator in assessing situations. However, as the amount of sensors and sensor data increases the decision maker can neither deduct all interactions, nor can he oversee all possible consequences of the occurring situations. Besides, many of the detected situations are recurring and predictable and pose no threat, but only hinder the operator in focusing at only relevant, possibly threatening, situations.

A future system that fully alleviates an operator from this task, consequently has to implement functionality found at JDL levels 2 and 3, and thus requires automatic situation and impact assessment. Due to the complex environments and the broad nature of possible situations and threats, it is a challenge to build such a system that can work fully independent. However, on a more short term basis, a system can be developed that supports the decision maker by taking over the assessment of recurring and predictable situations. An automated system can then reduce the information load of the information analyst and can directly support the decision maker by providing higher level information.

Recent studies focus on which data to use, how to fuse this data and how to present it to the decision maker. Effort is spread over a multitude of techniques including, but not limited to: rule based expert systems, decision trees, neural networks, and Bayesian networks [10]. It is important to be able to determine the importance of specific sensors and data sources. When using neural networks it is difficult to extract this type of information. Decision trees need to be retrained every time a new data source is added or when the context of a data source changes (e.g., a sensor is repositioned) [10]. Bayesian networks are used in a variety of studies such as [11], [12]. They capture the methods used by information analysts when assessing a situation, and classifying a situation as a threat. In both studies Bayesian networks are created based on world knowledge about platforms, roles, and behaviors of objects; in these cases respectively planes and ships. The studies show good results in taking over repetitive tasks from information analysts such that the focus can be switched to classifying less common threats. The threats can be identified as anomalous behaviors, or in specific cases as a specific threat. In the last case a label can be attached to a class with the threat type estimation.

III. Fusion Engine Architecture

In the DAFNE project, the focus is on enhancing situational awareness during military operations using a fusion engine able to combine data from the sensors employed in an urban warfare scenario. The fusion engine architecture aims at scalability (operation with heterogeneous sensors), flexibility (different environments without major modifications), extendibility (future proof, including dynamic situation and threat assessment techniques) and distributedness (increasing global awareness). In [7] and [8] the philosophy behind the architecture of the fusion engine and its resulting design are explained. The diagram in Figure 1 shows a representation of the resulting design, ordered by JDL levels. The tracker and classification modules fuse all information coming from the different sensors, resulting in a track history, describing all observed objects, possibly with annotated labels and classes. Labels are descriptions based on first order features when they can be measured directly, or on second or higher order features when they can be derived from lower order features. An example of a first order feature is a named color based on RGB values, and for a second order feature is an action label like ‘stopping’ based on the first order feature ‘velocity’. The tracks are input to the Decision Support module, which is divided into a Situation Assessment and a Threat Assessment part. Mapping this to the well known JDL data fusion levels (see for example [3], [4]), the Situation Assessment corresponds to JDL level 2 and the Threat Assessment to JDL level 3 (Impact Assessment).

The DAFNE fusion engine consists of three modules: a tracker, classifier, and decision support module. Also the context of the modules and their management are taken into account. Each module is discussed in more detail in the following paragraphs.

The multi-feature **tracker** associates observations from different sensors to single objects. This does not only include kinematic data, but also other measured features (e.g., color, size), or previously determined labels or classes.

The **classifier** derives labels from the associated observations. These may be derived by combining different features (classification by pattern recognition) or by combining labels from classification processing in sensors. Class information from earlier assessments or from different sensors is processed and combined into a new best estimate, resolving contradictions and correcting errors to the engine’s best ability.

The **Decision Support** tries to relate observed objects to relevant expected situations and threats in support of decision making algorithms or decision makers. It combines the management and assessment components for situations and threats and their respective settings in a single module. At all times, the module has access to the latest information from the previous modules, as well as to the information provided in the past. Relations between objects are considered to be situations. This enables complex combinations to be evaluated and assessed as a threat. For example, consider an object that fits part of a rule, but was not assessed to be dangerous. This object may still cause an alert when a relation appears with another object, which was indicated as dangerous because it came from a suspicious area.

Naturally, the system’s modules are used in a specific **context**, which may be useful for improving the modules’ performance. To that end, each module uses a description of the world to add new meaning to the information it gets as input. For example, the classifier uses relations between classes and features to classify objects. In decision support, the locations of geographical objects are used to link the position of objects to threats.

Some of this context information is mission independent (e.g., shape of a car), while other context information is mission dependent (e.g., position of roads and buildings). By making this description of the world available via a context database external to the modules, the system can be adapted to new environments and other missions. Only the mission dependent context information needs to be updated. This is especially relevant for the decision support modules.

The design includes **Management Modules** for all levels. In its most basic form, these modules translate context information about the world into a form a module can use (indicated as settings in Figure 1). They allow management of the system, by tuning this translation using the settings of the module at a higher level. For example, certain threat definitions (as Threat Assessment settings) may indicate that assessments related to particular geographical objects are required, which the Situation Assessment Management Module can then provide to the Situation Assessment. Although implementation of full system management is not planned, a basic form is used in Decision Support as described in the next sections.

IV. Situation Assessment

A. Functionallity

The Situation Assessment module maps what is known about observed objects to observables that are used by Threat Assessment (see Figure 2). These observables are linked to properties of the object (such as labels and classes) or describe relations between objects. For this, information from the Track History is used to relate observed objects to each other, and to relevant geographical objects, such as areas or buildings. The relevant geographical objects are provided by mission dependent context information. The result is new information on relations between already existing tracks, which is stored in the Situation History. Examples are persons who have met each other, cars that are in a convoy, a car being near a building, etc. By storing the history, relations in the past become available (e.g., a person was in the vicinity of an explosives factory). If relevant, track information can also be predicted into the (near) future, enabling prediction of relationships, such as a car is approaching an area. Relations to geographical objects are determined, evaluating different functions based on distance to a point or position relative to a polygon describing the object by name (e.g., Park Street) or the objects by type (e.g., Road). An example is "NEAR Embassy", which evaluates if the object is within a defined distance of a polygon defining the Embassy. Another example is "ON road", which evaluates if the object is within one of several polygons defining objects of type "road". Other possible relations not based on location are possible, defining for example established communication between persons based on mobile phone traffic.

B. Situation Context and Management

Situation Context represents situation specific data, that give mission specific a-priori information, e.g., the locations of high value targets (for instance an embassy), or the locations of hostile areas. It also gives mission independent a-priori information, e.g., functions related to object relations, such as when does an object qualify as "near" another object.

Situation Management allows for an automatic adjustment of the Situation Settings (see Figure 1), based on the requested Threat Settings. For example, when a user indicates that only threats in a certain area are of interest, the Situation Management component could make Situation Settings only provide information about geographical objects in this area, or ask situation and threat assessments from other, connected DAFNE nodes (as the DAFNE nodes support distributed assessments). It also controls the information about what particulars constitute a certain relationship. For example, a "near" relation could be defined differently for different types of geographical objects (e.g., within a given range around a point, or within a specified distance of a polygon). A full implementation of the management component is not within the scope of the current research. However, the component is explicitly defined to enable future modifications and extensions of the Fusion Engine towards fully dynamic fusion management.

C. Implementation

The current implementation of Situation Assessment consists of the following steps:

1) *Create list of observables*
   At initialization, a list of the required assessments (the observables) is created from the selected rules. The different types (i.e., label, class) and relations (NEAR, ON) are automatically extracted from the rule definitions (see Section V).
2) *Get relevant context*
   A list of relevant geographical objects is obtained from the Situation Context, and the corresponding observables (i.e., NEAR Embassy) are defined based on this information.
3) *Get current state*
   At each evaluation time, all tracked objects that are possibly relevant are evaluated. For all tracks, the track state (especially information such as location) has to be predicted to the current evaluation time, if the track state is not already available for the current evaluation time.
4) *Assess situations*
   The situations (a list of observables, represented by situation observables in Figure 2) are assessed for each object. For labels and classes, the belief in the object having the label, or belonging to a class, is the belief in the observable being true. For example, for labels it is checked if they are present, and if the belief is high enough. For class labels, the approach is similar. For the relations, the defined functions (NEAR, ON, etc.) are evaluated using the objects estimated location and the geographical entities descriptions. The functions result in a belief value, allowing for a range between 0 and 1 to indicate an area where the assessment is not certain. Relations to other tracked objects can be computed in a similar way, e.g., defining a NEAR function as a maximum distance between objects.
5) *Update situation information*
   The resulting assessment is set in the Threat Assessment module (see next section). The assessment result is a declaration of a situation being true, with indicated belief. For this, the initial belief value is thresholded, resulting in an assessment being true (with the indicated belief), or false (with 1-belief).

In the current implementation, context (geographical information and assessment function parameters) are read and set at initialization as well. Furthermore, all objects are evaluated. For a real situation this will create a computational load that is presumably too high, especially when evaluating object-to-object relations. Situation Management can help in this case, by asking the track history to only provide tracks that are close enough to any other relevant object. This distance

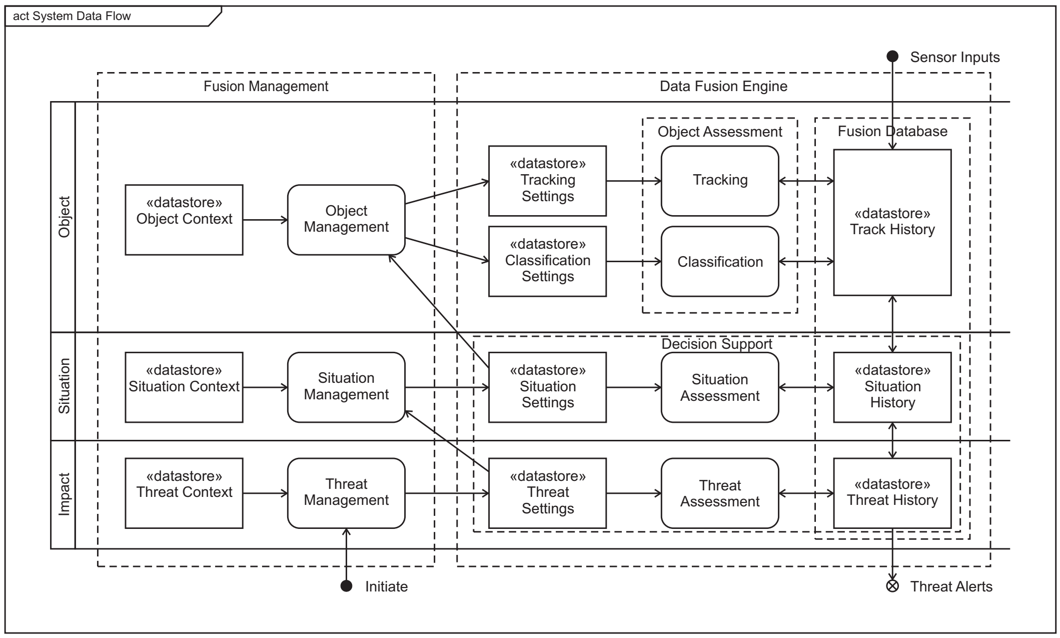


Figure 1. Fusion Engine Architecture. The Decision Support module containing Situation Assessment and Threat Assessment evaluates object level information from tracking and classification against threat rules, using context information.

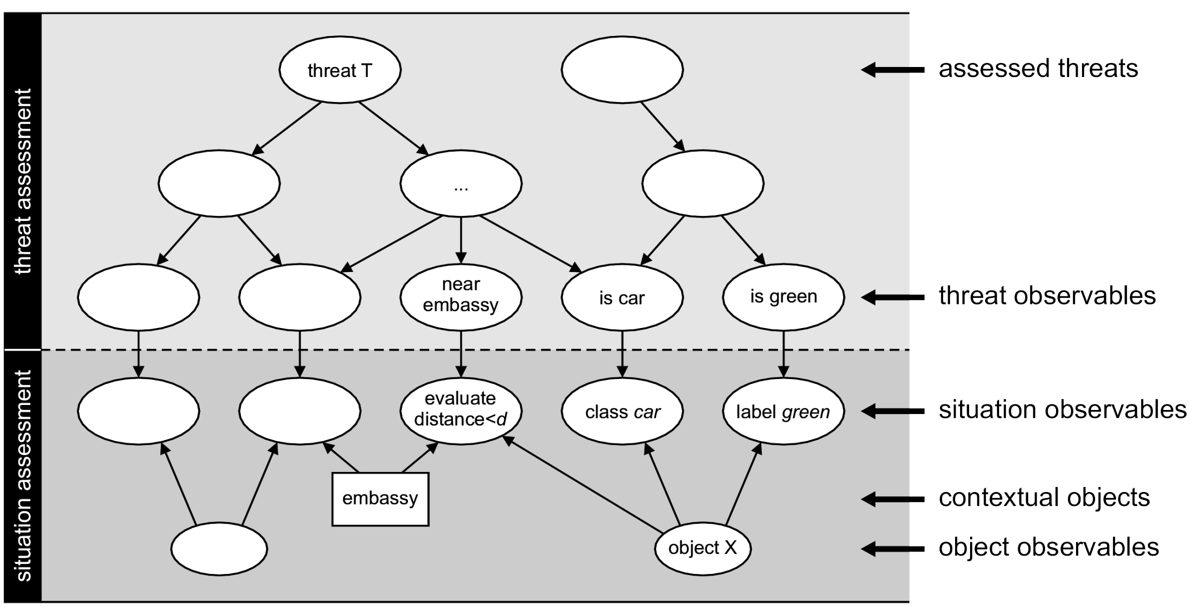


Figure 2. Schematic representation of the situation and threat assessment. Situation assessment assesses observables based on the current object estimates (classified track information). Situation observables are used to infer relevant threats.

should include uncertainties due to objects moving since the last track update until the current evaluation time, as well as being large enough to enclose possibly intricate polygon shapes of geographical objects. Relevant objects in this case are all geographical entities that are included in threat rule definitions, as well as all observed objects that have been marked as relevant due to earlier assessments. For example, an observed object may become relevant once it has been inside a certain area (e.g., a part of a city) or in a building. Relations to this object are now relevant, to later assess rules such as 'a car is near the embassy, which was in contact with a person who was in a suspect building'. Situation Management may further be implemented as a way to dynamically change context, for example when a user changes the list of relevant threat rules, or changes relevancy of geographical objects.

V. Threat Assessment

A. Functionallity

The Threat Assessment module combines the results from the Situation Assessment module (as situation observables in Figure 2) into an assessment of threats (the threat observables). Information about the object (labels and classes, for example, a green, stopping jeep) and its relationships in the past (was in a suspect part of a city) and present (is near the embassy) are used to evaluate the threat rules, to deduce a threat level. Threat Management defines the rules that lead to a situation being flagged as a threat, and is provided as (user) input to the system.

B. Context and Management

Context for the Threat Assessment is the information defining the threats to assess. Part of this information may be mission independent information (such as the rule definition: 'a vehicle approaching a forbidden area with high speed is suspect'), but most of the information will be mission dependent. This includes information describing which rules are relevant, which areas are not-public, or suspect, and what buildings need to be guarded. Mission dependent information may also include a-priori information, indicating how prevalent a certain situation is. This could include the probability of an actual attack, or what kind of classes are present (e.g., occurrence of different types of vehicles).

C. Implementation

Threat assessment itself is not a major topic of the DAFNE project. Simple rules are sufficient to evaluate threats, resulting in flags being set if the belief in a threat is high enough.

As implementation, Bayesian belief networks are chosen, which are created manually and tuned by hand. As implementation of the Bayesian belief network, the SMILE reasoning engine (together with the GenIe modeling environment) developed by the Decision Systems Laboratory (University of Pittsburgh) is used [13]. Each threat rule to be evaluated is defined by a belief network (see the example in Figure 3). A threat node is linked (directly or indirectly) to a number of nodes representing the threat observables in Figure 2. Each node indicates a belief in the threat or situation being true. A-priori probabilities are set to define belief in the situation being true in case of a threat or not. For each Situation node, an observable node is created, linked to the situation observables in Figure 2. These nodes indicate how the observation relates to the situation. For example, although an observation 'green' is done, the observation may be wrong. The belief in the observation gives an estimate of the probability that the observation is correct. This is set in the node, in the form of a confusion matrix, indicating the belief of an object being green and not being green, given the assessment it is green or not. The assessment itself and the values in the confusion matrix are the values set by the Situation Assessment module, as described in the previous section. For example, a belief in 'green' of 0.1, would result in the observation node being set to 'false' with the confusion matrix being set to have value 0.9 on the diagonal (i.e., the assessment being correct) and 0.1 off-diagonal.

The belief networks for all threats are updated after Situation Assessment changes the observables, and the resulting belief value of the threat node is read as output of the assessment of the threat.

The currently used networks are basic, and tuned to behave like the simple rules that are intended, while still leaving some probabilistic behavior. For example, a hard rule could have been evaluated by just thresholding beliefs of all observables and checking if all situations are true. By implementing the rules as belief networks, it is now possible to combine different rules, and for example raise an alarm if either a car is on a blacklist, or if several other assessments are true. Tuning the network is done by specifying the a-priori probabilities of the nodes. For this reasearch it is sufficient to make 'educated guesses' for the values of the a-priori probabilities.

Implementing a rule in a network can be done in many ways. Ancestry of nodes in this case is chosen in such direction that a small number of values need to be set for all situation nodes (namely, what the belief is in the situation being true or not, if the threat is true or not). Intermediate nodes may make rules more consistent, but are not used in these examples. To use intermediate nodes and bigger, more complex networks in general, a tool is recommended to automatically set the many values and connections in the network. However, this tool is not a part of this project.

It is also possible to define actions to be taken linked to a threat having a high probability. Related to these actions are the costs of making correct or wrong decisions. For example, not stopping an attack or acting on a false alarm may both be costly. The network can take those costs into account and determine what action to take. However, defining the costs is not straightforward and highly dependent on the action (e.g., blocking a possible attacker at a distance as opposed to shooting). For this reason, as output of threat assessment the probability of the threat is used.

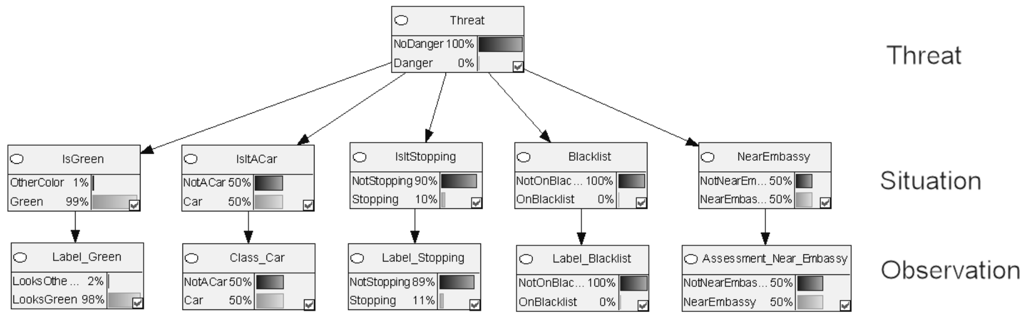


Figure 3. Example of a Bayesian network representing a rule for the Threat Assessment. The Threat node probability is the primary output. The bottom row of nodes is set as assessment of the situations represented by the row above.

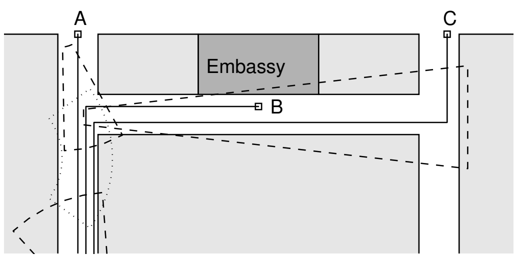


Figure 4. Example scenario for the protection of an Embassy. An alarm should be triggered if a car is stopping near the embassy, probably not being green and on a blacklist. Sensor coverage is indicated by the dashed and dotted lines.

VI. Example Scenario

In this section an example scenario (see Figure 4) is used to illustrate the decision support described in the previous chapter. This simplified urban scenario includes several streets and buildings. One of the buildings is a high value object to be protected (the Embassy). Regarded threats to the embassy are posed by cars nearby, that are either on a black list or slowing down and not embassy cars (which are assumed to be green). Several radars and cameras are distributed along the streets. Some of the cameras provide not only location measurements, but also classification as car or other type of object, and an indication of color. At one location, a camera system checks license plates against a blacklist. In this example, several cars are observed: (A) a blue car on the blacklist, not going past the embassy, (B) a blue car, not on the blacklist, stopping near the embassy, and (C) a green car on the blacklist going past the embassy. Even though the Management Modules are not implemented, they are included as first steps in the process described here, as their function is similar to the manual steps.

- *Threat Management*
  A Threat Management module might select or adjust rules for threats based on requirements from a user. In this case, the rule is created manually in the form of the network shown in Figure 3. The situations are given a-priori beliefs indicating that, in case of a threat, an object is certainly a car (belief near 1), stopping near the embassy, most probably not green. In case of a threat, the car is possibly (but not necessarily always) on the blacklist. If it is not a threat, it certainly is not on the blacklist. The bottom row of nodes indicates to Situation Management how to assess the corresponding situations, but values are not set yet. For example, the Assess_NEAR_Embassy indicates that the function NEAR needs to be assessed with respect to the geographical object named Embassy.
- *Situation Management*
  Functionality of a Situation Management module is partly included in the implementation, as it creates a list of assessments based on the definition of the rule, and includes information from the geographical context. In this case a function NEAR is defined as the maximum distance to the center of the geographical object named Embassy.
- *Object Management*
  The object and sensor level modules are not the aim of the research described in this paper. Object Management would provide the Classification Module with information how to classify a car, and how the velocity of a car is related to stopping.
- *Sensor data*
  In this example, simple sensors are simulated, adding a small error to the locations of the objects. Labels indicating 'green', 'on blacklist' and class 'car' are assumed to be indicated by the cameras with a certain belief.
- *Tracking and Classification*
  The tracker module tracks the objects by associating observations of objects from different sensors (such as the labels and class) in time. A label 'stopping' is added based on simple assessments of speed and acceleration.
- *Situation Assessment*
  For each evaluation instance, all situations are assessed, and the resulting values are set in the network nodes. In this example, the label and class information does not change in time, but differs for the three cars. At each evaluation, the current track location is used to re-compute the belief in the object being near the embassy. Resulting belief values for the three different cars are shown in Figure 5.

- *Threat Assessment*
  The network describing the rule with the values of the bottom nodes is updated (by Situation Assessment). The resulting belief value of the Threat node is an output of the system. Figure 5 shows the belief values as function of time.

The graphs in Figure 5 show the system response for the cars A, B, and C.

Car A is never assessed to be a threat. Although there is a high belief that car A is on the blacklist. The threat belief is low because the car is never near the Embassy.

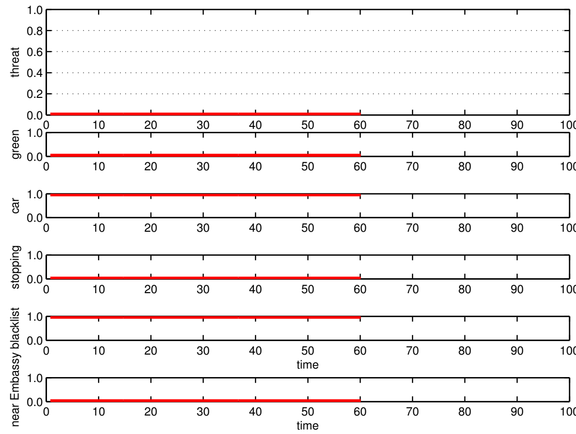
Car B is believed to be a threat. This belief is high, because of three reasons: 1) the car is, at some point, near the Embassy, 2) the car slows down near the embassy, and 3) the car is not green (and thus not an Embassy car). The belief that this car is a threat is then approximately 0.6. If the car would have been on the blacklist, this belief would have been higher. The graph shows that the assessment on 'stopping' fails a few times due to inaccuracies in the sensors data.

Car C, is green (and thus has a probability to be an Embassy car) and is not stopping. It is believed that the car is a threat when it is near the Embassy, because it is blacklisted. The belief in this car being a threat is of a rather short duration, because the car passes the Embassy and is, as a result, only near the Embassy for a short period of time. From the graph it is also clear that car C passes the Embassy shortly before car B arrives.

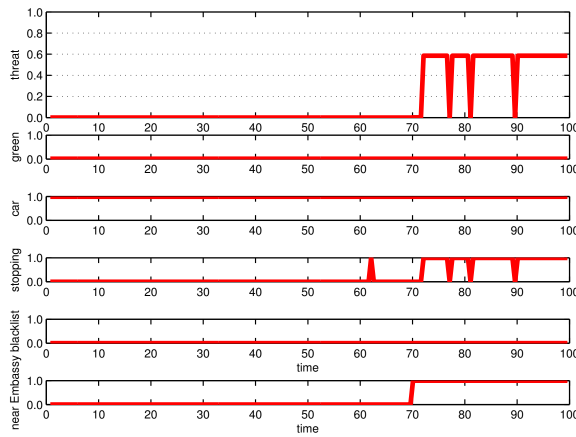
The example above is basic and easy to assess using the provided figures. However, this system is designed to be capable of assessing more complex scenarios. A more complex scenario could be as follows. Car A is in a suspected area, then drives into the city. Car B is in the city and waiting for car A. Car B and car A meet. Car B moves towards the Embassy and plans to slow down in front of the Embassy. The situation and treath assessment concequently can evaluate this scenario as follows. Car stops near the Embassy which previously is related to a car that was in a suspected area.

VII. Conclusions

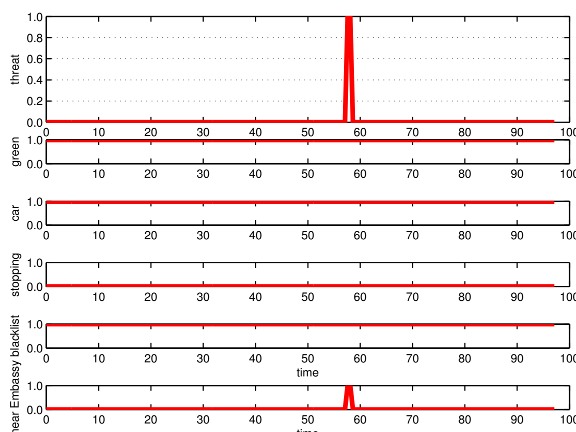
A general approach for a decision support system as part of a larger fusion engine is described in this paper. It allows for a Decision Support Module that can be used for a wide range of scenarios. For the current stage of the research, only a basic threat assessment is planned, which is currently implemented using Bayesian Belief networks describing simple rules. The link between this threat description and Situation Assessment are a set of basic observables, associated to labels and classes (to be provided by Object Assessment) and many possible functions describing relations between objects. A future automated system management and optimization is possible by using management modules linked to their corresponding assessment modules, which provide the needed context (such as geographical context and rules) filtered by the requirements at a higher level. A provided example shows the system in use for a simple scenario. This makes the system flexible and able to adapt to new environments and missions. Scalability



(a) Car A



(b) Car B



(c) Car C

Figure 5. Decision Support output for the example scenario for protection of an Embassy. Belief values are shown as function of time for the cars passing the Embassy. The top graphs show the final threat assessment, other graphs indicate the assessed situations. Car A is blacklisted and not stopping. Car B is stopping, not blacklisted, but does not have the expected color. Car C is of expected color, and passing by, but on the blacklist.

of the system to large number of objects and rules, is mainly covered by the distributed design of the fusion engine [7]. Further, situation management allows for situation and threat assessment to only process objects that are relevant for the threats to be evaluated.

ACKNOWLEDGMENTS

The work presented in this paper was partially done as part of the DAFNE project and is sponsored by the European Defence Agency (EDA) under grant number A_0380-RT-GC. The authors would like to thank the partners in the DAFNE consortium for their valuable comments and contributions regarding the architecture of the fusion engine [7].

REFERENCES

[1] J. J. Salerno, M. Sudit, S. J. Yang, G. P. Tadda, I. Kadar, and J. Holsopple, "Issues and Challenges in Higher Level Fusion - Threat and Impact Assessment and Intent Modeling - A Panel Summary," *13th International Conference on Information Fusion*, Jul. 2010.

[2] D. L. Hall and J. Llinas, "An introduction to multisensor data fusion," in *Proceedings of the IEEE*, vol. 85, January 1997, pp. 6 – 23.

[3] N. Steinberg and C. Bowman, "revisions to the JDL data fusion process model," in *proceedings of the 1999 national symposium on sensor data fusion*, May 1999.

[4] A. Steinberg and C. Bowman, "rethinking the JDL data fusion levels," in *proceedings of the national symposium on sensor data fusion jhuapl*, June 2004.

[5] A. N. Steinberg, "An Approach to Threat Assessment," *7th International Conference on Information Fusion*, Jul. 2005.

[6] European Defence Agency, "EDA defence R&T joint investment programme on force protection (JIP-FP)," http://www.eda.europa.eu/genericitem.aspx?id=184.

[7] M. Ditzel, S. van den Broek, P. Hanckmann, and M. van Iersel, "DAFNE - A Distributed and Adaptive Fusion Engine," in *Lecture Notes in Artificial Intelligence*, May 2011.

[8] M. Ditzel, L. Kester, and S. van den Broek, "System Design for Distributed Adaptive Observation Systems," in *14th International Conference on Information Fusion*, Jul. 2011.

[9] P. Pohanka, J. Hrabovsky, and M. Fiedler, "Sensors simmulation environment for sensor data fusion," in *14th International Conference on Information Fusion*, July 2011.

[10] F. W. Kong, G. W. Ng, Y. S. Tan, and C. H. Tan, "Evolving Fuzzy Neural Networks in Adaptive Knowledge Bases to support Task-Oriented Decision Making for Sensor Management," *10th International Conference on Information Fusion*, Jul. 2007.

[11] T. Lampinen, J. Ropponen, and T. Laitinen, "Joint Threat Assessment with Asset Profiling and Entity Bayes Net," *12th International Conference on Information Fusion*, Jul. 2009.

[12] R. O. Lane, D. A. Nevell, S. D. Hayward, and T. W. Beaney, "Maritime anomaly detection and threat assessment," *13th International Conference on Information Fusion*, Jul. 2010.

[13] Decision Systems Laboratory of the University of Pittsburgh , "GeNIe modeling environment and SMILE reasoning engine," http://dsl.sis.pitt.edu.