Combining system and user belief on classification using the DSmT combination rule

Wilbert van Norden
Defence Material Organisation
CAMS – Force Vision
Den Helder, the Netherlands
W.L.van.Norden@forcevision.nl

Fok Bolderheij
Defence Material Organisation
CAMS – Force Vision
Den Helder, the Netherlands
F.Bolderheij@forcevision.nl

Catholijn Jonker
Man-Machine Interaction Group
Delft University of Technology
Delft, the Netherlands
C.M.Jonker@tudelft.nl

Abstract* – *Having a correct and timely classification solution for objects has become increasingly important as well as increasingly difficult to obtain in new maritime military missions; a decision support system is therefore needed. In decision support systems a challenge lies in how operator and system belief can be reconciled. This paper presents a support system for the classification process using Dezert-Smarandache theory (DSmT) for information fusion. This system is implemented to test these concepts in practise. With this implementation we show that our methodology provides the operator with various levels of interaction with the system. The interface also shows the belief state of the system at any given time, increasing operator trust in the system.

Keywords: Classification, Dezert-Smarandache Theory, Human-Computer Interaction, Reasoning with uncertainty.

1 Introduction

The classification process aboard naval warships is becoming more and more complex, due to three main factors. Missions now are typically executed in littoral waters where rapidly changing environmental conditions make sensor performance harder to predict and enable hostile forces to stay hidden longer. Thus, the reaction time is diminished.

Furthermore, today's missions are characterised by asynchronous threats and missions draw more media attention. The first makes classification itself harder whereas the latter results in high pressure not to make mistakes, see [1].

The last factor refers to developments contradictory in impact: sensor systems are becoming increasingly complex, requiring increasing technical knowledge levels of the operators to optimally deploy the available sensors. On the other hand there is a strive to reduce the ship's complements and to reduce the training time for operators due to budget cuts.

Deploying sensors optimally is important for the overall picture compilation process and therefore for the classification process as well. Sensors can reduce the amount of uncertainty in the compiled picture, thus the input of the classification process is as correct and accurate as possible. When sensors are not deployed in an optimal fashion, much uncertainty occurs on the input of the classification process, reducing classification accuracy. Intelligent classification support can alleviate these problems. This paper focuses on the required cooperation mechanism between operator and system when automated classification is possible. In section two we will start with the background of this research, which is part of a larger ongoing research program. As an information combination methodology we choose to use Dezert-Smarandache theory (DSmT), which is briefly discussed in section three. How this theory can be applied in the field of classification in military command and control systems is discussed in section four. Section five explains how the operator can exert influence in the resulting classification system using DSmT. The implementation of this system is discussed in section six. Finally, sections seven and eight discuss future work and the conclusions of this research.

2 Background

The research presented in this paper is conducted as a follow up on the STATOR[1] project: a collaboration between the Royal Netherlands Navy, the International Research Centre for Telecommunication and Radar of the Delft University of Technology and Thales the Netherlands. Focus of this project was the management of sensor suites on single or multiple platforms and the fusion of the data provided by these sensors. The goal is to develop a decision support system where the operator can communicate with the sensor suite(s) as a whole in operational parameters without the need for technical knowledge. The overall concept used for this is discussed in [2]. We want to manage the whole suite because sensors can give similar and/or complementary data. Exploiting this, means that the sensor suite should be considered as a whole by the sensor manager.

[1] STATOR: Sensor Tuning And Timing on Object Request

2.1 Sensor Management

Previous research, [3], shows that sensor management seeks to compile and maintain a picture of the environment which is complete and accurate. In [4] a three-stage sensor manager is introduced based on this notion. The first stage determines the sensor task that needs to be executed in order to reduce the maximum amount of uncertainty in the compiled picture. Since the uncertainty in classification can be reduced in various ways, [5], and because this uncertainty reduction is important in achieving situation awareness, it seems logical to look at the classification process within the broader study into sensor management. More information about the three-stage sensor manager can be found in [2].

2.2 Classification

Having a *good* and *timely* classification solution is of vital importance to mission success in many fields. Before this can be done however, we need to describe:

1) what classification is;
2) what a *good* classification solution is; and
3) what a *timely* solution is.

The answer to the first question is described in section four, but in short: classification tries to recognise the observed object with as much detail as possible, e.g. recognising that a surface contact is the HNLMS[2] Tromp of the Dutch Air Defence and Command frigate class. Whereas identification, which is generally used for this most detailed classification stage, is the process that tries to solve whether the object is friendly, neutral or hostile.

A *good* classification is the solution where a sufficient amount of detail is obtained. E.g., the distinction between two types of sea skimming missiles is not very important: both will most likely destroy the ship so risk-wise they are equal. Distinguishing between an airliner and a fighter however, is important because the first constitutes far less risk than the latter. The definition of risk that we use and how it is calculated can be found in [6]. Besides the advantage of reducing the uncertainty in the risk posed by an object, a good classification also improves radar performance in tracking, as shown in [7].

In the military field, the starting point is to assume the worst-case scenario. For incoming objects this means that, at a certain point in time, precautionary actions must be taken. Before this happens, a classification solution could negate the necessity of actions thus preventing collateral damage. A *timely* solution is therefore the solution that reduces enough class uncertainty for deciding on appropriate actions.

In the classification process we search for a good and timely solution. This search space needs to be modelled in order to use automated classification techniques. This model should facilitate the requirements needed to find correct and timely solutions: the model needs to facilitate specific as well as more generic solutions. In section four we will discuss how the search space is modelled in the field of classification.

2.3 Interaction with the operator

Operational command and control systems in use with the Royal Netherlands Navy rely on operator knowledge, especially the classification process. However, new types of missions require more support for the classification process during missions. This does not mean that the operator should not classify: the operator should always be enabled to give the classification solution.

It is known that operators are not always best suited to solve complex problems in time. We therefore want to explore the possibility of more cooperation between the system and the operator, where each is responsible for the task they are best suited for. The operator makes tactical decisions and gives operational information relevant to the mission whereas the system can perform computations to solve the more technical problems. When cooperating, the operator needs to be able to trust the system. In order to earn operator trust, the system must be able to communicate its belief state and be able to explain its actions. The result is an architecture where the systems beliefs are combined with the user's belief in order to find good and timely solutions and where both belief available to the system are visualised.

3 DSmT combination rule

Dezert and Smarandache introduced a theory on combining paradoxical, uncertain, and imprecise information from various sources in [8] and this theory has led to many implementations as can be seen e.g. in [9] and [10]. This theory (denoted as DSmT) is detailed in [8], [9] and [10], and is discussed shortly in this section. We then continue with the Proportional Conflict Redistribution rule (PCR6[3]), [11], [12] and [13].

3.1 General DSmT

In this section we briefly discuss the basic concepts of DSmT that we use in the field of classification, namely the basic model, the belief function and the different combination rules.

3.1.1 The model

Let $\Theta = \{\theta_1, \theta_2, ..., \theta_n\}$ be the frame of discernment with exhaustive elements θ_i. This model is called free when no assumptions are made about the hypotheses θ_i except for the exhaustiveness.

[2] HNLMS: Her Netherlands Majesty; The prefix for all ships of the Royal Netherlands Navy.

[3] In [12] general PCR-rules are described; we however will only discuss and use PCR6.

This model does not fit most real-life problems since some combinations of hypotheses are time dependent or are not valid anymore when more knowledge becomes available. A hybrid model, denoted *M*, can be constructed to deal with these integrity constraints.
The cornerstone of DSmT is the free Dedekind lattice denoted in DSmT as the *hyper-power set*. This hyper-power set, D^{Θ}, is defined as the set of all composite propositions built from elements of Θ with $\cup$ and $\cap$ operators such that:

1. $\emptyset, \theta_1, \theta_2, \ldots, \theta_n \in D^{\Theta}$;
2. If $A, B \in D^{\Theta}$ then $A \cup B \in D^{\Theta}$ and $A \cap B \in D^{\Theta}$;
3. No other elements belong to D^{Θ} except those obtained using rules 1 and 2.

The cardinality of the hyper-power set for $n \geq 1$ follows the Dedekind sequence (i.e., 1, 2, 5, 19, 167, 7580, 7828353,...) as shown in [9]. Tombak et al., describe the analytical form of this sequence in [14]. From the frame of discernment Θ a map is defined $m(.): D^{\Theta} \rightarrow [0,1]$:

$$m(\emptyset) = 0 \text{ and } \sum_{X \in D^{\Theta}} m(X) = 1.$$

The quantity *m(X)* is called the generalised basic belief assignment (gbba) of *X*, also called the generalised mass of *X*.

3.1.2 Combination rules in DSmT

The classic DSmT rule of combination holds when the model is free. When *k* independent sources give their belief masses according to $m_1(.), \ldots, m_k(.)$, the combination rule for $\forall X \in D^{\Theta}$ is given in equation (1).
When the classic rule of combination is used in real-life fusion problems so-called integrity constraints must be taken into account to impose assumptions about the model. In such cases, the hybrid rule of combination for *k* independent sources with belief assignments $m_1(.), \ldots, m_k(.)$ is defined for $\forall X \in D^{\Theta}$ by equation 2.

$$m_c^{\,f}(X) = \sum_{\substack{Y_1,\ldots,Y_k \in D^{\Theta} \\ Y_1 \cap \ldots \cap Y_k = X}} \prod_{i=1}^{k} m_i(Y_i) \qquad (1)$$

$$m_c^{\,DSmH}(X) = \phi(X) \cdot [S_1(X) + S_2(X) + S_3(X)] \qquad (2)$$

In equation (2) all sets are in the canonical form and $\phi(X)$ is the characteristic non-emptiness function of a set *X*, i.e. $\phi(X) = 1$ if $X \notin \emptyset$ and $\phi(X) = 0$ otherwise, where $\emptyset = \{\emptyset, \emptyset_M\}$. $\emptyset_M$ is the set of all elements of D^{Θ} which have been forced to be empty through the constraints of the model *M* and $\emptyset$ is the classical empty set. In equation (2) the following is defined:

$$\begin{cases} S_1(X) = \displaystyle\sum_{\substack{Y_1,\ldots,Y_k \in D^{\Theta} \\ Y_1 \cap,\ldots,\cap Y_k = X}} \prod_{i=1}^{k} m_i(Y_i) \quad ; \\ S_2(X) = \displaystyle\sum_{\substack{Y_1,\ldots,Y_k \in \emptyset \\ [U = X] \vee [(U \in \emptyset) \wedge (X = I_t)]}} \prod_{i=1}^{k} m_i(Y_i) \quad ; \\ S_3(X) = \displaystyle\sum_{\substack{Y_1,\ldots,Y_k \in D^{\Theta} \\ Y_1 \cup \ldots \cup Y_k = X \\ Y_1 \cap \ldots \cap Y_k \in \emptyset}} \prod_{i=1}^{k} m_i(Y_i) \quad ; \end{cases}$$

with $U = u(Y_1) \cup \ldots \cup u(Y_k)$ where $u(Y)$ is the union of all θ_i that compose *Y*. and I_t is the union of all elements in Θ, in other words: total ignorance.
Since it is not always desirable to transfer conflicting mass to relevant ignorance, other combination rules were developed. Here, we choose to use the PCR6 combination rule, [11], to avoid transferring masses to relative ignorance.

3.2 PCR6

The general idea behind the PCR6 rule is to transfer conflicting masses to the non-empty elements that are involved in the conflict as opposed to transfer it to relative ignorance, which is the case in hybrid DSmT. The various PCR rules can be found in [12], and the PCR6 rule, which we use, is discussed in [11] and defines the rule for k independent information sources as equation (3). The different combination rules given in equations (1-3) are illustrated using example 1, from [13].

$$m_c^{\,PCR6}(X) = m_c^{\,f}(X) + \sum_{i=1}^{k} F_i \cdot m_i(X)^2 \qquad (3)$$

In equation (3) the following is defined:

$$F_i = \sum_{\substack{C_1 \\ C_2}} \left(\frac{\prod_{l=1}^{k-1} m_{\sigma_i(l)}\left(Y_{\sigma_i(l)}\right)}{m_i(X) + \sum_{l=1}^{k-1} m_{\sigma_i(l)}\left(Y_{\sigma_i(l)}\right)} \right) \quad \text{with}$$

$$C_1: \quad \bigcup_{j=1}^{k-1} Y_{\sigma_i(j)} \cap X \in \emptyset;$$

$$C_2: \quad \left(Y_{\sigma_i(1)}, \ldots, Y_{\sigma_i(k-1)}\right) \in \left(D^{\Theta}\right)^{k-1};$$

$$\sigma_i(l) \rightarrow \begin{cases} \sigma_i(l) = l & l < i \\ \sigma_i(l) = l + 1 & l \geq i \end{cases}.$$

Example 1

Let the frame of discernment be $\Theta = \{A, B\}$. Two experts have given their opinion as follows: $m_1(A) = 0.6$, $m_1(A \cup B) = 0.4$, $m_2(B) = 0.4$ and $m_2(A \cup B) = 0.7$.

When assuming Shafer's model of exclusiveness the conflicting mass is $m_{12}{}^{f}(A \cap B) = 0.18$. Using the different combination rules, Table 1 is obtained.

Table 1 Generalised basic belief assignments of two experts and the results of three combination rules.

	A	B	$A \cup B$	$A \cap B$
$m_1(.)$	0.6	0	0.4	0
$m_2(.)$	0	0.3	0.7	0
$m_{12}{}^{f}(.)$	0.42	0.12	0.28	0.18
$m_{12}{}^{DSmH}(.)$	0.42	0.12	0.46	0
$m_{12}{}^{PCR6}(.)$	0.54	0.18	0.28	0

4 Using DSmT in classification

Most classifiers in military applications use classification trees, [15] and [16]. The problem with such an approach is that once the classifier gets stuck on a high level node it cannot classify further, although information is available to make a decision about a lower node. We therefore propose to use a more dynamic approach where 'branching' is done based on the available information. Or in other words, we define a search space where all possible objects are represented and the available information bounds the search space. Defining the search space like this means that elements in the search space are not necessarily exclusive, DSmT therefore seems a good mechanism to apply.

4.1 Classification model

As stated before, getting a good and timely classification of objects in the environment is essential for many mission-critical systems. Important in decision support systems that are to be utilised in this field is the combination of the terms *good* and *timely*. An exact solution is not as important as getting a suitable solution at the right time. For classification this means that the fast classification "helicopter" is preferred over a solution to provide distinction between an Apache and a Seahawk. Certain hierarchy in the classification space is therefore necessary, as well as a system that can operate on and switch between all hierarchical levels. Of course, the interdependencies between the different levels need to be defined: all elements on a certain hierarchical level are mapped to one or more elements at the next higher hierarchical level.

Since exclusiveness is not required in DSmT, we can apply it with the PCR6 combination rule for a hierarchical structured classification space. For this domain we define three hierarchy levels: specific, generic and super classes. Belief can be assigned on any of the hierarchical level(s) and the combination mechanism based on DSmT can then combine beliefs as well as deal with conflicts.

The specific classes in the classification domain represent the different types of objects (e.g., an F16 or an Apache). The set of specific classes is defined as $C = \{\theta_1, \theta_2, ..., \theta_n\}$. This set consists of n exhaustive and exclusive elements.

The set of generic classes, $G = \{\gamma_1, \gamma_2, ..., \gamma_m\}$, consists of m exhaustive and exclusive elements. In this set, objects like 'fighter' and 'helicopter' are represented. Each element of G contains a subset of C.

Finally, the set of super classes, $S = \{\sigma_1, \sigma_2, ..., \sigma_k\}$, consists of k exhaustive elements. Each element from this set contains a subset of G. For the classification problem we consider the union of C, G and S as the frame of discernment ($\Theta = C \cup G \cup S$).

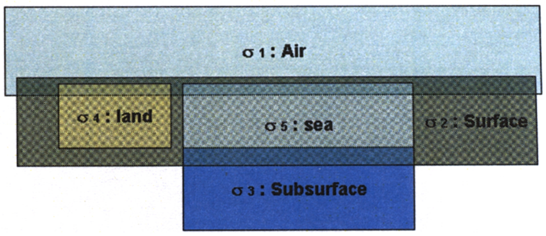

Figure 1 The different domains represented as Venn diagram of S

Within the classification domain for military applications we define S using the different domains. This produces set S with five elements defined as the domains: σ_1 *air*, σ_2 *surface* and σ_3 *subsurface*. Furthermore, σ_4 and σ_5 represent the sub-domains *land* and *sea* of the surface class respectively. The Venn diagram of these five elements is given in Figure 1. Overlaps with the surface domain are included since both air and subsurface objects can operate in the surface domain, e.g. a surfaced submarine or low flying helicopter. Example 2 illustrates how these hierarchical levels are used in the classification domain.

Example 2

Let C be given by:

θ_1:	Seahawk;	θ_6:	Apache;
θ_2:	F-16;	θ_7:	M-frigate;
θ_3:	Walrus-class;	θ_8:	K-class;
θ_4:	ADCF;	θ_9:	F-14;
θ_5:	Leopard II;	θ_{10}:	Boeing 747.

Let G be given by:

γ_1:	Helicopter;	γ_4:	Frigate;
γ_2:	Fighter;	γ_5:	Tank;
γ_3:	Submarine;	γ_6:	Airliner.

For this example $\gamma_1 \cap C = \{\theta_1, \theta_6\}$, $\gamma_2 \cap C = \{\theta_2, \theta_9\}$, $\gamma_3 \cap C = \{\theta_3, \theta_8\}$, $\gamma_4 \cap C = \{\theta_4, \theta_7\}$, $\gamma_5 \cap C = \{\theta_5\}$ and $\gamma_6 \cap C = \{\theta_{10}\}$ holds.

Two classifiers give information on their belief as given in Table 2 and Table 3 respectively. The third classifier assigns its generalised belief as $m_3(\sigma_1) = m_3(\sigma_2) = 0.4$ and $m_3(\sigma_3) = 0.2$.

Table 2 The classification solution from source 1

X	$m_1(X)$	X	$m_1(X)$	X	$m_1(X)$	X	$m_1(X)$
θ_1	*0.150*	θ_6	*0.150*	γ_1	*0.250*	σ_1	*0.150*
θ_2	*0.005*	θ_7	*0.040*	γ_2	*0.002*	σ_2	*0.100*
θ_3	*0.000*	θ_8	*0.000*	γ_3	*0.000*	σ_3	*0.000*
θ_4	*0.036*	θ_9	*0.005*	γ_4	*0.041*	σ_4	*0.010*
θ_5	*0.009*	θ_{10}	*0.005*	γ_5	*0.005*	σ_5	*0.040*
				γ_6	*0.002*		

Table 3 The classification solution from source 2

X	$m_2(X)$	X	$m_2(X)$	X	$m_2(X)$	X	$m_2(X)$
θ_1	*0.075*	θ_6	*0.075*	γ_1	*0.175*	σ_1	*0.130*
θ_2	*0.010*	θ_7	*0.075*	γ_2	*0.002*	σ_2	*0.120*
θ_3	*0.020*	θ_8	*0.020*	γ_3	*0.010*	σ_3	*0.005*
θ_4	*0.075*	θ_9	*0.020*	γ_4	*0.100*	σ_4	*0.020*
θ_5	*0.010*	θ_{10}	*0.020*	γ_5	*0.010*	σ_5	*0.025*
				γ_6	*0.003*		

4.2 Model constraints

Due to the fact that C and G contain exclusive elements all the intersections of these elements in the hyper-power set can be constrained. Furthermore, due to the structure of S given in Figure 1 some intersections in the superclasses can be discarded as well. The PCR6 rule is used to redistribute all masses assigned to these intersections. We call the constraints that are made at this point, the model constraints since they are caused by the modelling of the classification space.

With PCR6 we have a well defined, 'easy' to implement solution to transfer all model constraints in example 2. This is straightforward because these constraints all deal with similar situations, as was the case in example 1, e.g. the model dictates that an object cannot belong to the air and the subsurface domains. The mass associated with that possibility is proportionally transferred to the air domain and the subsurface domain.

Using this combination scheme in example 2 to resolve the modelling conflicts we obtain the results shown in Figure 2 and Figure 3.

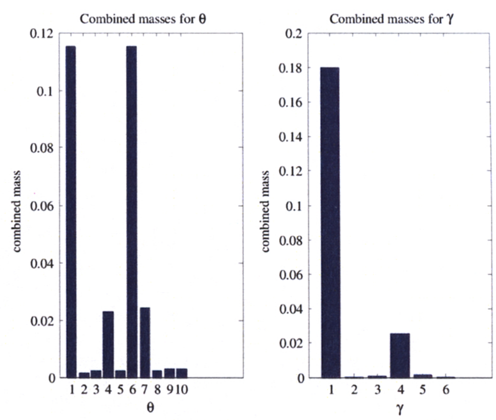

Figure 2 The histograms of the combined belief masses for C and G after applying PCR6

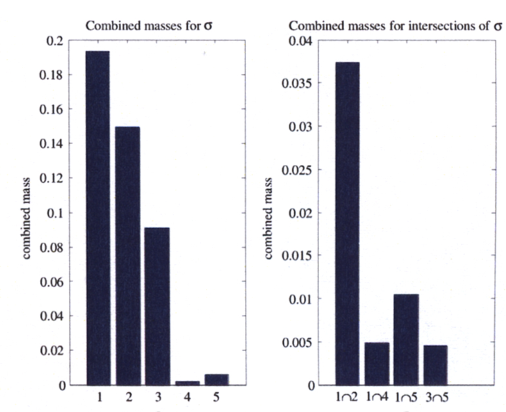

Figure 3 The histograms of the combined belief masses for S after applying PCR5 to resolve model constraints.

5 Interacting with the user

Imposing user constraints will be somewhat more complex than imposing model constraints, since more elements are involved. Looking at example 2, say that the operator imposes that the object is not a subsurface contact, $m(\sigma_3) \to 0$. The structure of the model then forces all masses assigned to the underlying elements of that domain to zero. The exception to this basic scheme is the surface domain: when set to zero all underlying elements that also belong to the air or the subsurface domain should not be set to zero due to the chosen modelling of the classification space, e.g. when an operator indicates that it is not a surface contact, the low flying helicopter should still be under consideration. Furthermore, the operator has the freedom to impose constraints on either *C*, *G* or *S*.

5.1 Transferring user constraints

Using PCR6 to transfer user constraints means that all conflicts are redistributed to elements involved in the conflict. However, this works when constraints are placed on intersections of two of more elements: the generalised masses are then redistributed to the single elements. Where should generalised masses be transferred to when e.g. the user constrains θ_{10} in example 2? A general approach could be to transfer it to the element at a higher hierarchical level, in this case γ_1. This fits the modelling of the classification space. The belief held on a specific class will transfer to the more generic class when the first is contradicted. Now however, the problem occurs when an element at the highest hierarchical level is constrained: where should that mass be transferred to?

Table 4 Transfers between elements of *S*.

Set to zero	Transfers to
σ_1	σ_2
$\sigma_2\,(\sigma_4, \sigma_5)$	σ_1, σ_3
σ_3	σ_2, σ_5
σ_4	σ_2
σ_5	σ_2, σ_3

This problem can be solved by proportionally transferring masses at the highest level in accordance with Table 4, which in turn is based on the overlaps of the different domains as can be seen in Figure 1. E.g., when the user indicated that an object does not belong to the air domain, the generalised mass for that domain is transferred to the surface domain only, since that is the most generic and most likely other candidate due to the overlaps between the two.

Applying this method results in an accumulation of generalised masses at the highest hierarchical level since the masses of all underlying elements of the constraint domain are transferred as well. We therefore propose to transfer the masses of those underlying elements to elements at the same hierarchical level that are children of the element that the mass was supposed to go following Table 4.

5.2 Conflict

Constraining the subsurface domain in example 2 agrees with the information given by the two other sources, in other words: the operator agrees with the system, as shown in Figure 3. When the user imposes a similar constraint on the air domain, another problem arises: the system disagrees with the operator. This conflict can be interpreted and used in three ways.

Firstly, the operator might be wrong. In this case, the conflict can be used as a alert for the operator to review the case. The conclusion can go both ways. If the operator agrees with the system this is of added training value for the operator. If the system is wrong, this instance can be used to train the automated classification algorithms.

Secondly, the conflict can be used to trigger sensor measurements. The conflict between system and operator can be traced to a single classifier that causes most of the conflict. By looking at the input of that particular classifier, we can tell which uncertainty of its input should be reduced in order to try and reduce the amount of conflict. With that information a sensor task can be generated for the sensor manager.

And lastly, when conflict is not resolved by such sensor tasks, two other options must be considered: 1) the object is classified correctly but is behaving very unexpectedly; or 2) a sensor system is degraded and is giving wrong measurements. Either way, the amount of conflict is used to alert the operator should this occur.

6 Implementation

The theories from the previous sections have been implemented as a part of the ongoing research in sensor management at CAMS – Force Vision. The simulation environment and the scenarios that were developed in the STATOR project have been expanded with classifiers to test the fusion of system belief and operator input. Note that the classifiers used are simple ones and will later be replaced by more intelligent classifiers that e.g. take temporal aspects into consideration. This section describes the architecture of the fusion system and the interface for classification.

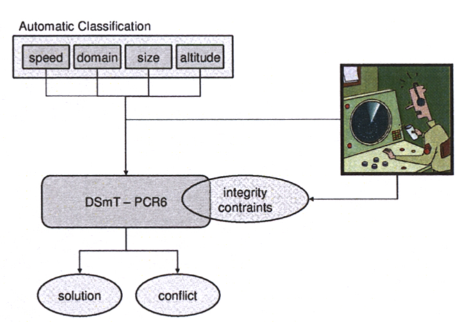

Figure 4 Basic system architecture to combine classification solutions

6.1 System architecture

In section 5.2 we saw that the user and the system interact on several levels. At the highest level, the operator defines the mission, which influences the parameters of the classifiers. This level is discarded for now since it is a

component in the mission planner and not of the classification system itself. More information on this subject can be found in [17].
Two levels are left that do need to be implemented. Firstly, the level where the operator constrains the model for each individual detected (or expected) object. The second level is the operator classifying objects manually. Making this separation means that two separate combination rules must be used. The resulting system architecture is shown in Figure 4. Five different (simple) classifiers were implemented that try to classify objects based on:

1) speed;
2) domain;
3) object size;
4) altitude; and
5) area (for this classifier the mission settings define where objects are more likely to occur).

This system has two outputs. The first, of course, is the combined belief that is assigned to the various elements of the classification space. The second, the amount of conflict, is important for feedback purposes, as mentioned in section 5.2. This feedback informs the operator about decisions he makes and on how well this fits into the systems belief. The amount of conflict can be used as a trigger for the sensor manager.

6.2 Classification Interface

Besides the influence the operator has on the classification solution, it is also important for the operator to know what the current state of belief is in the system. The resulting interface, shown in Figure 5, gives the operator input possibilities and displays the current belief state of the system. In this figure, the colour indicates the amount of belief the system has in a certain element on all hierarchical levels. The operator can exclude (sets of) elements, which makes those elements transparent. The amount of conflict is also tracked. Furthermore, the user can classify an object by clicking the appropriate element. The system belief is then combined with the operator's solution using a DSmT combination rule as shown in Figure 4.
In the interface, the amount of conflict is also displayed. The amount of conflict determines the colour in which the numeric value is displayed, which alerts the operator if it increases too much. The conflict level is only determined by the constraints the operator places on the classification combination rule. When the operator classifies an object himself, the combined value to that class indicates how much the system classifiers agree with that solution; the colour intensity of that particular node also indicates that value as additional information source. This indication can prevent the operator from tunnel vision, i.e. it will emphasize contradicting evidence when the operator does not have enough time to notice that.

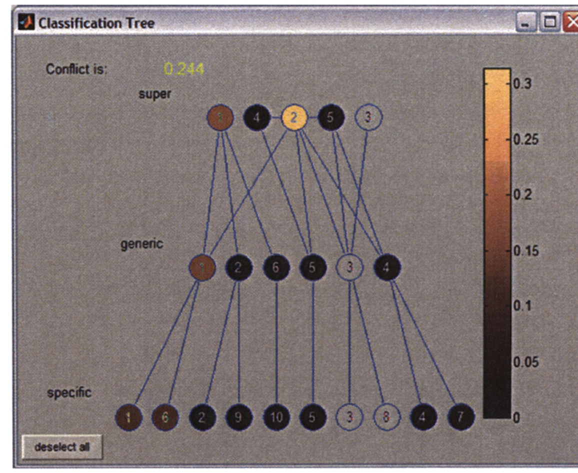

Figure 5 Screenshot of the interface where the operator can view the system belief on the three hierarchical levels.

7 Future Work

In [3] command and control concepts and sensor management concepts were tested in a simulation environment. We will implement the classification systems discussed here in that environment. Another addition to that environment will be the implementation of the automated classifiers discussed in [17] and the temporal aspects within the classification process.
By expanding the implementation of the overall concepts in more complex scenarios, we will be able to conduct serious gaming tests to validate the cooperation mechanisms discussed in this paper.
Sensor management is the overall subject of ongoing study in the Royal Netherlands Navy. Mechanisms to generate sensor task requests based on the reasoning process with uncertainties need also be developed. These requests should, of course, fit into the overall sensor management concepts. As scheduling mechanism for these functions we expect to use the results from [18]. The combination will result in an overall new command and control concept.

8 Conclusions

This work shows that it is possible to use DSmT to combine operator and system belief on classification solutions. Modelling the classification space to comply with DSmT fits the real life situation in the military domain.
Operator input is very important in military applications. The interface we presented here enables an operator to exert influence on various levels in a flexible way. Furthermore, the interface gives the operator insight into the belief state of the system. The result of this insight will give the operator reason to trust the system, which leads to better cooperation.

References

[1] H. Gerner, *Politiek plot op zee: Information Operations tijdens de NATO oefening Noble Midas 2007*, Marineblad, Vol. 118, No. 1, February 2008, pp. 17-21, (in Dutch)

[2] Tanja van Valkenburg- van Haarst, Wilbert van Norden, Fok Bolderheij, *Automatic Sensor Management: challenges and solutions*, Proc. of the SPIE Defence and Security conference, Optics and Photonics in Homeland Security (6945-33), Orlando (FL), March 2008

[3] Fok Bolderheij, Frans Absil, *Mission-Oriented Sensor Management*, Proc. of 1st conference on cognitive systems with interactive sensors; Paris, March 15-17, 2006

[4] Fok Bolderheij, Frans Absil, Piet van Genderen, *Risk-Based Object-Oriented Sensor Management*, Proc. 8th Int. Conf. on Information Fusion, Philadelphia, (PA), July 25-29 2005

[5] Wilbert van Norden, *Intelligent task scheduling in sensor networks*, Delft University of Technology, Masters thesis with the Man-Machine Interaction group, April 2005

[6] Fok Bolderheij, Piet van Genderen, *Mission Driven Sensor Management*, Proc. 7th Int. Conf. on Information Fusion, Stockholm, June 28 – July 1 2004, pp. 799-804

[7] Y. Bar-Shalom, T. Kirubarajan, C. Gokberk, *Tracking with classification-aided multiframe data association*, IEEE Transaction on Aerospace and Electronic Systems, Vol.41, No.3, July 2005, pp. 868-878

[8] F. Smarandache, J. Dezert, *An Introduction to the DSm Theory for the Combination of Paradoxical, Uncertain, and Imprecise Sources of Information*, Proc. 13th International Congress of Cybernetics and Systems, Maribor, July 6-10, 2005,

[9] F. Smarandache, J. Dezert, *Advances and Applications of DSmT for Information Fusion*, [Internet]. Rehoboth (MA): American Research Press; 2004 [cited 2005 May 16], Available from: http://www.gallup.-unm.edu/~smarandache/DSmT-book1.pdf

[10] F. Smarandache, J. Dezert, *Advances and Applications of DSmT for Information Fusion (Collected Works), Volume 2*, [Internet]. Rehoboth (MA): American Research Press; 2006 [cited 2008 March 18], Available from: http://www.gallup.unm.edu/~smarandache/DSmT-book2.pdf

[11] A. Martin, C. Osswald, *A new generalization of the proportional conflict redistribution rule stable in terms of decision*, Chapter 2 in *Advances and Applications of DSmT for Information Fusion (Collected Works), Volume 2*, edited by F. Smarandache and J. Dezert, American Research Press, Rehoboth (MA), 2006

[12] F. Smarandache, J. Dezert, *Proportional Conflict Redistribution Rules for Information Fusion*, Proc. 8th Int. Conf. on Information Fusion, Philadelphia (PA), July 25-29 2005

[13] M. Florea, J. Dezert, P. Valin, A-L. Jousselme, F. Smarandache, *Adaptive combination rule and proportional conflict redistribution rule for information fusion*, Proc. of 1st conference on cognitive systems with interactive sensors; Paris, March 15-17 2006

[14] M. Tombak, A. Isotamm, T. Tamme, *On logical method for counting Dedekind numbers*, Lect. Notes on Comp.Sci., 2138, pp. 424-427, Springer-Verlag, 2001

[15] Bionda Mertens, *Reasoning with uncertainty in the situational awareness of air targets*, Delft; 2004. Masters Thesis with the Man-Machine Interaction Group of the Delft University of Technology

[16] Krispijn Scholte, *Dynamic Bayesian networks for reasoning about noisy target data*, Den Helder: Royal Netherlands Naval College, Research Report for the Combat Systems Department, December 2005

[17] Wilbert van Norden, Fok Bolderheij, Catholijn Jonker, *Classification support using confidence intervals*, Proc. of the 11th International Conference on Information Fusion, Cologne, June 30 – July 3 2008

[18] Jeroen de Jong, Wilbert van Norden, *Application of hybrid metaheuristics in sensor management*, Aerospace Science and Technology, Vol. 11, No.4, May 2007, pp. 295-302