Modeling of Expert Knowledge for Maritime Situation Assessment

Yvonne Fischer* and Jürgen Beyerer*†
*Vision and Fusion Laboratory, Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
†Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (IOSB)
Karlsruhe, Germany
Email: yvonne.fischer@kit.edu, juergen.beyerer@iosb.fraunhofer.de

***Abstract*—In today's surveillance systems, there is a need for enhancing the situation awareness of an operator. Supporting the situation assessment process can be done by extending the system with a module for automatic interpretation of the observed environment. In this article, the information flow in intelligent surveillance systems is described and a detailed modeling of the situation assessment process is presented. The main contribution of this article is a probabilistic modeling of situations of interest. The result of this modeling is a Situational Dependency Network (SDN), which represents the dependencies between several situations of different abstraction levels. The focus is on a top-down approach, i.e., the modeling is done in a human-understandable way and can be done by maritime experts. As especially critical situations can change very fast in their characteristics and also they do not happen very often, the machine learning approach is not appropriate for detecting such situations, even if they are very powerful. Therefore, we present an approach, where expert knowledge can be included into a Dynamic Bayesian Network (DBN). In this article, we will show how a DBN can be generated automatically from the SDN. We mainly focus on the determination of the parameters of the model, as this is the crucial point. The resulting DBN can then be applied to vessel tracks and the probability of the modeled situations of interest can be inferred over time. Finally, we present an example in the maritime domain and show that the probabilistic model yields the expected results.**

Keywords—surveillance system; situation awareness; situation assessment; data fusion; dynamic Bayesian networks; probabilistic reasoning.

I. Introduction

This article is based on our previous work that was presented at the ICONS 2012 conference and published in [1]. In this section, we will describe the motivation of our approach.

During the operation of complex systems that include human decision making, the processes of acquiring and interpreting information from the environment forms the basis for the state of knowledge of a decision maker. This mental state is often referred to as situation awareness [2], whereas the processes to achieve and maintain that state is referred to as situation assessment. In today's maritime surveillance systems, the situation assessment process is highly supported through various heterogeneous sensors and appropriate signal processing methods for extracting as much information as possible about the surveyed environment and its elements. They are equipped with powerful sensors like radar systems, the Automatic Identification System (AIS), see [3], or even infrared cameras. All of these sensor systems, either active or passive, are able to detect vessels in the observed area. The surveillance system fuses the estimated positions into consistent tracks, which then can be displayed in a dynamic map. Maritime surveillance systems have been used so far to monitor traffic, to guide passing vessels, and to ensure compliance with traffic regulations. Using these sensor systems is, of course, an essential capability for every surveillance system in order to be able to observe a designated area and to detect and track objects inside this area.

But there is an increasing demand for using their power in security-related applications like the detection of critical situations. However, current systems do not provide any support in the detection of spatio-temporal patterns, i.e., situations. Thus, there is a need for concepts and methods that are able to infer real situations from observed elements in the environment and to project their status in the near future. The challenge of intelligent surveillance systems is therefore not only to collect as much sensor data as possible, but also to detect and assess complex situations that evolve over time as an automatic support to an operator's situation assessment process, and therefore enhancing his situation awareness.

In applications like this, the current approach is to learn the characteristics, i.e., the features of the situation of interest with some training data, and recognizing the situations based on the trained model. There is a lot of machine learning methods that can be used for detecting sequential patterns, see for example [4] or [5]. All methods have in common that they need training data. In order to use such machine learning techniques for estimating maritime situations, there are several challenges that have to be addressed and we will highlight them in the following.

- *Data Collection:* There has to be an access to sensors that are able to collect vessel data. Data of different sensors, e.g., AIS and radar, should be fused into a consistent representation of vessel tracks. The representation should not only contain position data over time, but also additional information like speed, course, MMSI (maritime mobile service identity), beam, length, etc. Since surveillance systems are equipped with these sensors, data collection can be provided easily.
- *Data Labeling:* The collected data has to be labeled in order to perform supervised training. As we want to

detect specific situations, the vessel tracks have to be labeled, i.e., it has to be determined for which time interval the situation is either true or false. The labeling step is done by humans and is therefore extremely time-consuming. Moreover, labeling of higher-level situations is not always straightforward, as critical situations can have very similar patterns than non-critical situations.

- *Data Selection:* It has to be decided, which of the labeled data should be used for training the model. There has to be enough data to represent several variations of the situation. Especially when the model has many parameters, there has to be enough training data in order to avoid overfitting.
- *Model Validation:* It has to be guaranteed that the trained model is able to recognize situations under various circumstances. So there should be a kind of testbed for evaluating the model under real circumstances to determine the reliability and trustworthiness of the results when using the trained model.

Thus, the process of generating training data for one situation of interest is quite complex and time-consuming. Moreover, this process has to be done for every situation of interest. But especially in security-related applications, situations of interest can change very rapidly in their characteristics and they do not happen very often. This means that in most cases, there is only few or even no training data available. Therefore, the machine learning approach is not fully satisfactory for detecting such situations, even if these methods are very powerful.

For human experts instead, it is quite easy to formulate and define the characteristics for new situations of interest. Therefore, our approach is to model this expert knowledge instead of using machine learning approaches. For modeling the expert knowledge and recognizing the situations of interest, we use a probabilistic model, i.e., a Dynamic Bayesian Network (DBN). DBNs, and especially their simplified version, the Hidden Markov Models (HMM) are widely used in machine learning approaches for situation recognition. The potential of these models is for example shown in [6] for maritime surveillance and in [7] for traffic scenarios. It has been shown that these models are able to handle noisy input data like wrong observation values.

However, maritime experts are in general not familiar with such kind of models. They are not able to determine the parameters of the model, i.e., the conditional probabilities, which are the crucial point for the model to yield the expected results. In order to support maritime experts in modeling their own situations of interest, we present an approach for setting the parameters automatically, based on the structure of the DBN. The structure of the DBN is generated based on a Situational Dependency Network (SDN), a human-understandable modeling of the situations of interest. The most challenging task is to determine the parameters in a way that the resulting DBN behaves like the human expert would expect it to behave.

The paper is structured as follows. In Section II, an overview of related work is given. The information flow in intelligent surveillance systems is highlighted in Section III. A detailed description of the situation assessment process is given in Section IV. Section V covers a definition of the term situation and it is shown how situations of interest can be characterized and how they can be represented in a SDN. Section VI starts with a brief review of DBNs and it is shown how the existences of situations of interest can be inferred. In Section VII, the approach of generating the DBN, especially the parameters, is presented. Finally, an application scenario in the maritime domain is given in Section VIII and it is shown that the approach yields the expected results.

II. Related Work

Working with heterogeneous sensors, the theories of multi-sensor data fusion [8] offer a powerful technique for supporting the situation assessment process. A lot of research has been done in combining object observations coming from different sensors [9], and also in the development of real-time methods for tracking moving objects [10]. Regarding data fusion in surveillance systems, the *object-oriented world model (OOWM)* is an approach to represent relevant information extracted from sensor signals, fused into a single comprehensive, dynamic model of the monitored area. It was developed in [11] and is a data fusion architecture based on the JDL (Joint Directors of Laboratories) data fusion process model [12]. Detailed description of the architecture and an example of an indoor surveillance application has been published in [13]. The OOWM has also been applied for wide area maritime surveillance [14].

In [15], an overview of different approaches of modeling and recognizing situations in the area of pervasive computing is presented. However, these approaches can also be used in the area of surveillance systems. In his article, Ye et al. divide between two main techniques, specification-based techniques and learning-based techniques. DBNs are not addressed directly in his article, only HMMs, which are a special case of DBNs. However, HMMs are usually used for learning-based approaches. We will not list machine learning approaches here, as they are out of scope of this paper, but we refer to [15] for an extensive discussion on them.

Specification-based techniques are used, when humans model the situations of interest directly. We refer to them as expert modeling approaches. A very recent work using specification-based techniques has been published in [16], in which petri nets are used for modeling and recognizing situations of interest. In [17], situation modeling is done by situational graph trees and situation recognition is performed based on fuzzy-metric temporal logic. However, most of the specification-based techniques are based on ontological modeling, see for example [18], [19]. The reasoning in ontologies is performed by queries based on description logic and there exist several publications on the detection of maritime situations with ontologies, e.g., [20], [21], [22], or [23]. The main drawback in ontological modeling is that it is not possible to deal with uncertain information. For this reason, probabilistic reasoning mechanisms, as provided by Bayesian networks, are often used for situation assessment, e.g., [24] and [25].

Another development is the extension of the Web Ontology Language (OWL), see [26], with a probabilistic representation, which results in the Probabilistic Web Ontology Language

(PR-OWL) presented in [27]. PR-OWL applies the theory of Multi-Entity Bayesian Networks (MEBN) that have been developed by Laskey [28]. MEBN are an extension of Bayesian networks in the form that they allow representation of graphical models with repeated sub-structures. Applications of PR-OWL and MEBNs are presented in [29] for the semantic web, and in [30], [31] for maritime applications. In [28], an algorithm was presented, which creates situation-specific Bayesian networks, based on the modeled MEBN. However, the main drawbacks in the MEBN-approach are that the parameters of the network still have to be inserted manually and that it is not possible to generate DBNs. However, the situation assessment should be able to deal with noisy observations and thus, the method of DBNs should be used. The work of this artice is inspired by the MEBN-approach and addresses the aforementioned problems.

First ideas of modeling situations in surveillance applications have been presented in our previous work in [32]. In [33], a Bayesian network, was applied to observed objects in the maritime domain and the user acceptance of such an automatic situation assessment was shown. Further work has been done in including temporal dependencies into the model, i.e., by defining a DBN for the detection of vessels that are most likely to carry refugees on board, and it was shown that the DBN-approach yields promising results, see [1]. As modeling a DBN is a difficult task and maritime experts are in general not familiar with probabilistic methods, as they are not able to determine the parameters of the model, a method for an automated generation of a DBN was developed and presented in [34] for scenarios on a parking space. A similar approach was then applied to the maritime domain in [35]. This paper extends the previous developed concepts for characterizing situations at different abstraction levels and the methods for generating a DBN.

III. Information Flow in Surveillance Systems

In this section, we will describe the information flow in surveillance systems in a general way. The general information flow for intelligent surveillance systems is visualized in Figure 1, wherein information aggregates are represented by boxes, and processes are represented by circles. The information flow is as follows.

In surveillance applications, the task is to observe a spatio-temporal section of the real world, a so-called *world of interest*. We will term all elements in the world of interest *entities*. By the term entity, not only physical objects like vessels are meant, as entities can also be non-physical elements in the real world like vessel attributes, relations between vessels or situations. Furthermore, not all entities can be observed, as there is no sensor to observe them directly. Thus, entities can represent observable or non-observable elements. All entities together represent the complete state of the world of interest.

The next step in the information flow is the *observation* of entities by several sensors. Sensor systems for observing the real world can be of extremely heterogeneous types, e.g., video cameras, infrared cameras, radar equipment, or radio-frequency identification (RFID) chips. Even human beings can act like a sensor by observing entities of the real world. Observing the world of interest with sensors results in *sensor data*, for example a radar image or a video stream.

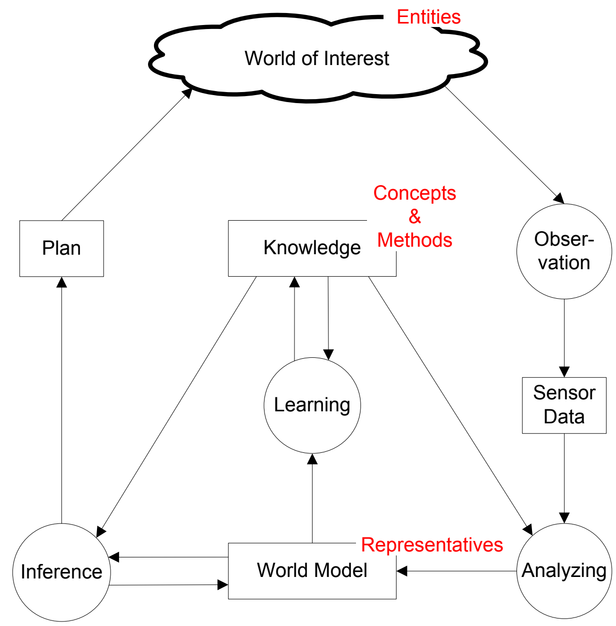


Fig. 1. Information flow in a surveillance system represented by information aggregates *(boxes)* and processes *(circles)*.

Analyzing sensor data is done by means of knowledge and the resulting information is transferred to the world model. Analyzing sensor data includes for example the detection and localization of moving vessels at sea from a video stream. Knowledge contains all information that is necessary for analyzing sensor data, for example specific signal-processing methods and algorithms used for the detection, localization and tracking of vessels in video streams.

The *world model* is a representation of entities in the world of interest and consists therefore of *representatives*. Every representative has a corresponding entity in the real world. The mapping between entities in the world of interest and representatives in the world model is structure-preserving and can therefore be interpreted as a homomorphism. Specific mappings are defined by concepts, which are part of the knowledge. Concepts are for example used in the analyzing process by defining how an observed vessel is represented in the world model. As the world of interest is highly dynamic and changes over time, the history of the representatives is also stored in the world model.

However, as mentioned before, some entities cannot be observed directly. Therefore an *inference* process is reasoning about non-observable (and also unobserved) entities by means of knowledge. A simple inference process is for example to calculate an object's velocity from the last and current position. A more complex inference process would be to estimate if the intention of an observed vessel is benign or adversarial. Doing this way, the world model is always being updated and supplemented with new information by predefined inference processes. Thus, during operation, the world model tries to estimate a complete representation of the world of interest in every time step.

Summing up, *knowledge* contains all information for analyzing sensor data, updating the world model and supplementing it with new information. Knowledge consists of abstract *concepts* and also of *methods*. Concepts are used for the representation of real-world entities in the world model and methods are used for analyzing data or inferring further information. Characteristics of the knowledge are of course extremely dependent on the application domain. Additionally, knowledge is not static. The content of the world model can be used for acquiring new knowledge by a *learning* process. This could be, for example, a method for structure or parameter learning in probabilistic graphical models.

To close the loop of the information flow, the result of an inference process could also include a *plan* of how to act further in the real world. Thus, the inference process can also act like a decision support, on which the action plan is based. The plan itself can be an action plan for an agent, for example, to call the police, or a sensor management plan, for example, a request for more detailed information from a special sensor.

Finally, we have to mention that the presented information flow in surveillance systems is not intended to act fully automatically. Every process can be designed in a way that a human operator is involved and that he is able to use the system interactively.

In the next section, we will have a more detailed look at a specific inference process, namely the situation assessment process. The situation assessment process tries to estimate predefined situations of interest by using the information of assessed objects over time.

IV. The Situation Assessment Process

By situation assessment, we mean the process of estimating the existence of situations of interest, which is conform to [2] and [12]. We divided the whole process into several sub-processes, as we state that situation assessment does not only consist of the process of recognizing situations, but also of the process of characterizing and modeling situations of interest, based on the current task. The conceptual framework of the process is depicted in Figure 2 and will be described in the following.

During the process of *object assessment*, estimates of objects are created, which do not only include kinematic state estimates like tracking the position and velocity of vessels, but also descriptive attributes of the object. The result of this process are *object representatives*, which are stored in the world model over time. Thus, we will first describe how the concept of an object is defined, i.e., how it is stored in the world model.

The concept of an object is defined as a physical entity of the real world. An object belongs to exactly one object class. As an object has several attributes, an object class is defined as the equivalence class of identical attribute lists. The attributes of the object can be divided into properties and states. Properties are time-invariant attributes, e.g., the length or the name of a vessel. State values can change over time and are therefore time-variant, e.g., the position or the velocity of a vessel. Regarding its spatial position, an object can be mobile, e.g., a vessel, or stationary, e.g., a land border. Thus, for a vessel the position is a state attribute and for the land border the position is a property. The concept of an object is visualized in Figure 3.

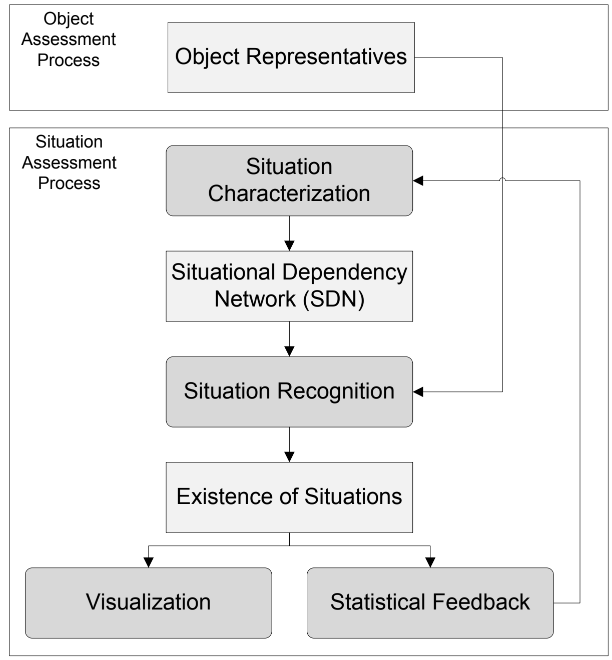


Fig. 2. The process of situation assessment.

As the representation in the world model also has a memory, which means that the past states of an object are stored, the complete history of the observed object is always available. As this operation is very memory-intensive, for practical reasons the world model should be connected to a database and only the latest objects should be hold in memory. Furthermore, the representation of an object in the world model does not only include observed attributes, but also inferred ones. For example, based on observed positions of a vessel, the velocity can be inferred. Furthermore, attribute values can be quantitative or qualitative. For example, the absolute position and velocity of a vessel are quantitative attributes, and the attribute value that a vessel is made of wood is a qualitative one. The selection of the attributes is still up to the user. However, the selected attributes should be dependent on the sensor capabilities and also on the required task of the situation assessment.

The process of situation assessment is divided into the sub-processes of situation characterization, situation recognition, visualization, and statistical feedback, which will be explained in the following. The first sub-process of *situation characterization* includes the a-priori modeling of expert knowledge about situations of interest. Because there are a lot of semantic dependencies between situations, these have to be modeled in order to estimate the existence of the situations correctly. This sub-process results in a *Situational Dependency Network (SDN)*. We will address the process of characterizing situations in detail in Section V.

The sub-process of *situation recognition* analyses the object representatives over time with respect to the existence of the

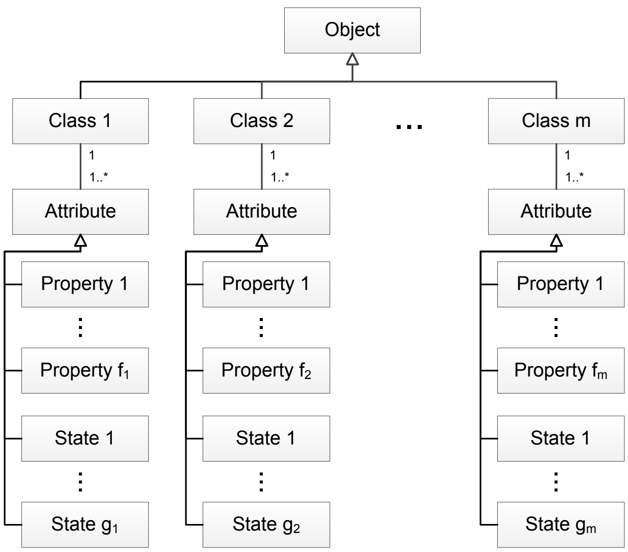


Fig. 3. The concept of an object.

situations of interest. Thus, it applies the SDN to the estimated object properties and states and infers, if the situations of interest are existing or not. The result of this process is a set of existence probabilities, one existence probability for each situation of interest. As we use DBNs for the process of situation recognition, the existence probabilities can be calculated by state-of-the-art inference mechanisms and algorithms that have been developed for DBN. Inference can be performed either exact or approximate. We refer the reader to [36] for detailed information on such algorithms, as this is out of scope of the article. However, many algorithms are available and ready to use, e.g., in specific software [37] or Matlab toolboxes [38]. Based on the resulting probabilities, the existence of situations can be inferred, if the probability value exceeds a certain threshold value.

This calculation, combined with an appropriate *visualization*, allows for a prompt assessment of the whole situation and thus for prompt decisions. Inferred situations of interest can be visualized in a dynamic map, where the involved vessels are highlighted. Also the calculated existence probability of the situations of interest can be visualized additionally. Of course, visualization is a crucial point of supporting situation awareness. Even if the results are calculated correctly, the visualization of them to the user during operation can be done in a wrong way, e.g., by presenting him too much information. This results in an information overload and thus, reduces his situation awareness [2]. However, evaluating different visualization methods for situations of interest is out of scope for this article and is therefore not discussed in detail.

Furthermore, a *statistical feedback* can be calculated from the set of existence probabilities over time. This can be used for refining the process of situation characterization and thus is a process for integrating the user in the situation assessment process. The feedback can have several objectives, for example, it could suggest a refinement of the SDN or could indicate situations that have not been detected at all, i.e., ask the user if the situation is still of interest.

V. Characterizing Situations of Interest

In this section, we will describe how situations can be modeled as random variables and how their existence is defined. We will introduce two different situation types and how the dependencies between several situations can be modeled. Finally, we will show how a complete SDN can be modeled.

A. Situation Modeling

Before we are able to characterize situations of interest, we have to define the term situation. In [15], a situation is defined as follows:

> "A situation is defined as an external semantic interpretation of sensor data. Interpretation means that situations assign meanings to sensor data. External means that the interpretation is from the perspective of applications, rather than from sensors. Semantic means that the interpretation assigns meaning on sensor data based on structures and relationships within the same type of sensor data and between different types of sensor data."

This definition corresponds to our understanding of the term situation, but we try to give a more formal definition of a situation. First of all, we state that a situation at time point t is always connected with an external semantic statement, which is either true or false. The semantic statement is always based on a temporal sequence of a specific constellation of modeled objects and their attributes.

We define $O^1, O^2, \dots, O^n$ as the set of objects that are relevant for the semantic statement. We define $A_1^i, A_2^i, \dots, A_{m_i}^i$ as the set of the relevant attributes of the object O^i, with $i = 1, \dots, n$. If an object or an attribute is relevant or not is induced by the semantic statement of the situation and has to be defined by the user. For example, for the situation that a vessel is fast, only it's velocity is relevant and it's heading can be ignored. We can then define the configuration space $\mathcal{O}$ as

$$\mathcal{O} = \mathop{\times}_{i=1}^{n} \mathop{\times}_{k=1}^{m_i} r(A_k), \tag{1}$$

where $r(A_k)$ denotes the range of the attribute values of A_k. $\mathcal{O}$ has then the dimension $\dim \mathcal{O} = \sum_{i=1}^{n} m_i$.

We set $\Omega = \mathcal{O} \times T$, where T represents the time and define a situation S_t at time t as the mapping

$$S_t : \Omega \to \{0, 1\}. \tag{2}$$

We say that the set $\tilde{\Omega} = \widetilde{\mathcal{O}} \times \widetilde{T} \subseteq \mathcal{O} \times T$ is the support of the Situation S_t, if

$$S_t(\omega) = \begin{cases} 1, & \text{if } \omega \in \widetilde{\Omega}, \\ 0, & \text{if } \omega \notin \widetilde{\Omega}. \end{cases} \tag{3}$$

Example 1: The semantic statement is that two objects are close to each other, e.g., a yacht is close to a tanker. Relevant objects are A and B, with a one-dimensional position value

x_a and x_b, respectively. We chose a one-dimensional position for visualization purposes. It is $\Omega = \mathcal{O} \times t$, where t is a time point. Then it is

$$\omega \in \widetilde{\Omega} \Leftrightarrow |x_a - x_b| \leq r, \tag{4}$$

where r is the threshold value. The support of this situation is visualized in Figure 4.

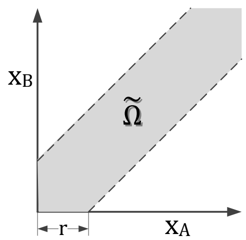


Fig. 4. The support of the situation that object A is close to object B.

Example 2: The semantic statement is that two objects are approaching each other, e.g., a vessel is approaching a specific harbor. Relevant objects are A and B, with a one-dimensional position value x_a and x_b, respectively, where x_b is static. It is $\Omega = \mathcal{O} \times T$, where T is a discrete time interval. Then it is

$$\omega \in \widetilde{\Omega} \Leftrightarrow |x_a - x_b|_t < |x_a - x_b|_{t-1}, \forall t, t-1 \in T. \tag{5}$$

The support of this situation is visualized in Figure 5.

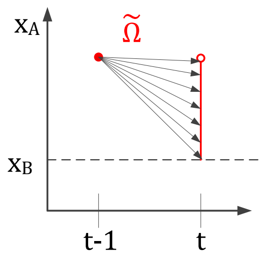


Fig. 5. The support of the situation that object A is approaching object B, with a selected value x_A for time point $t-1$.

B. Existence of Situations

We say that a situation exists, if the semantic statement is true, and that it does not exist, if the semantic statement is false. Thus, a situation exists, if the observed objects can be assigned to the relevant objects of the configuration space and for their attribute values it holds $\omega \in \widetilde{\Omega}$.

For modeling the existence of situations, we interpret the situation, i.e., the mapping $S_t : \Omega \to \{0,1\}$ as a binary random variable. It is P a probability measure on $S_t(\Omega)$, thus a probability distribution of S_t. Then the existence of a situation S_t at time point t is given by $P(S_t = 1)$, or shortly $P(S_t)$. Thus, when performing situation recognition, we are interested in the probability $P(S_t)$.

Due to this modeling, situations are characterized by information collected over a time-period, but they only exist at a special point in time. Their existence in the next time-point has to be verified again. The time-period itself is induced by the semantic statement of the situation. In example 2, it is enough to take $[t-1, t]$ into account. But for other situations it may be necessary to expand the time-period, e.g., for the situation that the vessel was in a suspicious area during the last 2 hours.

C. Different Situation Types

We now face the problem that the semantic statement of situations can be arbitrary complex and on a high abstraction level, e.g., a vessel is a suspicious smuggling vessel. Thus, the recognition of the situation depends on various characteristics and the direct modeling of the support of the situation is not always possible. Because of this fact, we can differentiate between the following two cases:

- The existence of a situation can imply the existence of other situations.
- The existence of a situation can lead to the existence of another situation.

These dependencies can be used for the recognition of situations. To use them, we will define two types of situations, namely elementary situations and abstract situations:

- *Elementary situations*: The support of the situation can be modeled directly. The existence of the elementary situation S_t can be modeled as the deterministic mapping

$$P(S_t|\omega) = \begin{cases} 1, & \text{if } S_t(\omega) = 1, \\ 0, & \text{if } S_t(\omega) = 0. \end{cases} \tag{6}$$

- *Abstract situations*: It is not possible to model the support of the situation directly. The existence of the abstract situation S_t is dependent of the existence of n other (elementary or abstract) situations $S_t^1, S_t^2, \ldots, S_t^n$

$$P(S_t|S_t^1, S_t^2, \ldots, S_t^n) = \frac{P(S_t, S_t^1, S_t^2, \ldots, S_t^n)}{P(S_t^1, S_t^2, \ldots, S_t^n)}. \tag{7}$$

However, with an increasing number of dependent situations, it is not possible for an expert to model the joint probability distribution $P(S_t, S_t^1, S_t^2, \ldots, S_t^n)$. The solution for this is to use the chain rule of probability by using the aforementioned dependencies, e.g., for two situations S^1, S^2 this would be:

$$P(S^1, S^2) = P(S^1|S^2)P(S^2) = P(S^2|S^1)P(S^1). \tag{8}$$

Thus, it is sufficient to model the conditional probabilities between dependent situations. An attempt to visualize these dependencies is depicted in Figure 6. Of course, in real applications, it is often not straightforward to identify the dependencies between situations. In practice, the identification of dependencies have to be done by experts together with system developers.

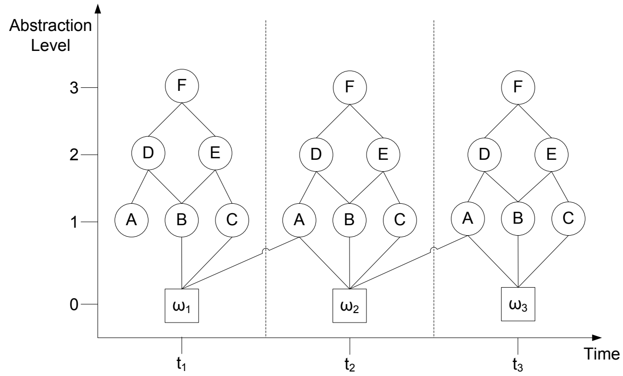


Fig. 6. Visualization of dependencies and abstraction levels, visualized without direction of dependencies.

D. Situational Dependencies

After defining situations formally, we have to consider their semantic interpretation, especially their relationships among each other. In [15], Ye et al. distinguish between five different types of relationships: generalization, composition, dependence, contradiction and temporal sequence, which we will repeat shortly in the following:

- *Generalization:* A situation is more general than another situation, if the occurrence of the latter implies that of the former.
- *Dependence:* A situation depends on another situation if the occurrence of the former situation is determined by the occurrence of the latter situation.
- *Composition:* A situation can be decomposed into a set of smaller situations, which is a typical composition relation between situations.
- *Contradiction:* Two situations can be regarded as mutually exclusive from each other if they cannot co-occur at the same time in the same place on the same subject.
- *Temporal sequence:* A situation may occur before, or after another situation, or interleave with another situation.

Our approach is now to model these relationships in a SDN. For the SDN, we divide the relationships into two main categories: sufficient and necessary conditions. We will explain them in the following:

- *Necessary condition:* A situation A is necessary for another situation B, if the existence of B implies the existence of A, i.e.,

  $$B \xrightarrow{\text{N}} A.$$

  If we have more than one necessary situations $A_1, \ldots, A_n$, we have

  $$B \xrightarrow{\text{N}} A_1 \wedge A_2 \wedge \ldots \wedge A_n.$$

- *Sufficient condition:* A situation A is sufficient for another situation B, if the existence of A implies the existence of B, i.e.,

  $$A \xrightarrow{\text{S}} B.$$

  If we have more than one sufficient situations $A_1, \ldots, A_n$, we have

  $$A_1 \vee A_2 \vee \ldots \vee A_n \xrightarrow{\text{S}} B.$$

Thus, we can always interpret the arrow from A to B as follows: If situation A exists, then situation B exists. Or in logical notation: $A \Rightarrow B$, namely A implies B. $A \xrightarrow{\text{N}}$

Compared to the five different types of relationships defined in [15], we can state the following:

- *Generalization:* The generalization is in our case modeled as the sufficient condition.
- *Dependence:* The dependency is in our case modeled as the necessary condition.
- *Composition:* The composition is in our case modeled through the necessary condition with more than one necessary situation.
- *Contradiction:* The contradiction is not explicitly modeled in our case, but should, of course, be represented by the semantics of the model. This means, the parameters of the DBN should be determined in a way that the two situations cannot exist simultaneously.
- *Temporal sequence:* The temporal sequence is not yet addressed in our model so far. However, we can extend the model by defining a specific sequence of existences of different situations as a situation of interest. We can recognize this sequential situation, for example, by comparing it to the results of the Viterbi-algorithm, which calculates the probability of the most likely sequential situation.

E. Situational Dependency Network

We further assume that in real world applications, sensor observations will be noisy. Noisy observation data can appear as wrong observations or as missing observations. It is also possible that some modeled elementary situations cannot be observed because there is no sensor available that would be able to observe the necessary information. Of course, the fact of noisy observations has a big influence on the situation recognition process.

Thus, we want to achieve a kind of inertia in the process of situation recognition. The inertia of the process supports the following statements:

- If we observe an area of interest over time, a single observation should not yield to the recognition of a situation, as it could be a noisy observation.
- If a situation of interest exists in the time step before, it is very likely that it exists in the current time step, also if different observations are made.

For achieving this inertia in the process, we have to extend our model with recursive, i.e., temporal arrows. We will add the temporal arrow for abstract situation of interest, whose elementary situations are assumed to be noisy, i.e., visually draw an arrow from the situation at time point t to the same situation at time point $t+1$. The temporal arrows can then be used in the DBN and result in a filtering effect of the existence probability. As we will show later, the strenght of the filtering effect can be adjusted by different weights. These weights will have an influence on how many observations have to be made to justify the existence of a situation or how many noisy observations can be made without rejecting the existence.

Example 3: In this example, we have three situations:

- Situation *area*: The vessel was in a suspicious area, known for smuggling activities.
- Situation *AIS*: The vessel is not sending any self-identification signal like AIS.
- Situation *smuggling*: The vessel is a suspicious smuggling vessel.

The first and the second situations (area and AIS) are elementary situations and the third one (smuggling) is an abstract situation, which we are interested in, and it is dependent on the two others. The dependencies are as follows. If a vessel was in a suspicious smuggling area, it is very likely that the vessel is a smuggling vessel. Thus, the area situation is a sufficient condition for the smuggling situation and we draw an arrow from area to smuggling. If a vessel is a smuggling vessel, then it is pretty sure that it does not send any identification signal. Thus, the AIS situation is a necessary condition for the smuggling situation and we draw an arrow from smuggling to AIS. As we assume noisy observations, we add a temporal arrow to the smuggling situation. The overall SDN is depicted in Figure 7, where the temporal arrow is indicated with a red T in the lower right corner of the node. Note that the same SDN can be used, even if the suspicious area itself can change.

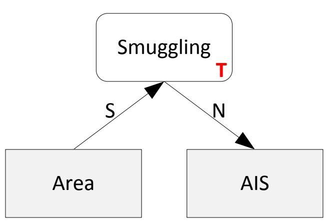


Fig. 7. Example of a situational dependency network.

Finally, we can present the general approach for modeling the SDN in Algorithm 1 and the calculation of the abstraction level in Algorithm 2.

VI. Recognizing Situations of Interest

Due to this modeling, the SDN can be interpreted as a probabilistic graphical model, namely a DBN. In a simple Bayesian network, the basic idea is to decompose the joint probability of various random variables into a factorized form. We will now describe how a DBN is defined and how the existence probabilities of the modeled situations can be inferred.

Algorithm 1 Creating a SDN

Require: set of situations and known pairwise dependencies
Ensure: SDN
  model situations as nodes
  for all nodes **do**
    if support can be modeled directly **then**
      declare it as elementary situation
    else
      declare it as abstract situation
    end if
  end for
  for all dependencies **do**
    if situation A is sufficient for situation B **then**
      model the dependency as an arrow from A to B
    end if
    if situation A is necessary for situation B **then**
      model the dependency as an arrow from B to A
    end if
  end for
  for all abstract situations **do**
    if abstract situation is dependent on an elementary situation with noisy observations **then**
      add a temporal arrow to the abstract situation
    end if
  end for

Algorithm 2 Calculating abstraction levels

Require: SDN
Ensure: level of abstraction for each node
  for all elementary nodes **do**
    abstraction level =1
  end for
  i=2
  initialize $\widetilde{\boldsymbol{S}}$ with a non-empty set
  while $\widetilde{\boldsymbol{S}} \neq \emptyset$ **do**
    set $\widetilde{\boldsymbol{S}}$ to the set of abstract situations that are only dependent on situations with abstraction level $i-1$
    set all elements of $\widetilde{\boldsymbol{S}}$ to abstraction level i
    set $i=i+1$
  end while

A. Dynamic Bayesian Networks

In a Bayesian network, random variables $X^1, X^2, \ldots, X^n$ are depicted as nodes and conditional probabilities as directed edges. The joint probability can then be factorized as

$$P(X^1, \ldots, X^n) = \prod_{i=1}^{n} P(X^i | Pa(X^i)), \tag{9}$$

where $Pa(X^i)$ is the set of parents of the node X^i. If $Pa(X^i)$ is an empty set, then X^i is a root node and $P(X^i|Pa(X^i)) = P(X^i)$ denotes its prior probability.

A DBN [39] is defined as a pair $(B_0, 2TBN)$, where

- B_0 defines the prior distribution $P(\boldsymbol{X}_0)$ over the set $\boldsymbol{X}_0$ of random variables, and
- $2TBN$ defines a Bayesian network over two time

slices with

$$P(\boldsymbol{X}_t|\boldsymbol{X}_{t-1}) = \prod_{i=1}^{n} P(X_t^i|Pa(X_t^i)), \quad (10)$$

where X_t^i is a node at time slice t and $Pa(X_t^i)$ is the set of parent nodes, which can be in the time slice t or in the time slice $t-1$.

Note that in the definition of a $2TBN$, $Pa(X_t^i)$ is never empty, i.e., every node in time slice t has at least one parent node and, therefore, the left side of equation (9) differs from the left side of equation (10). An example of a $2TBN$ with 3 nodes in each time slice is shown in Figure 8.

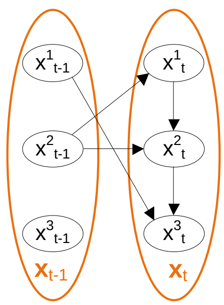


Fig. 8. An example of a $2TBN$ defining dependencies between two time slices and dependencies between nodes in time slice t adopted from [10]. Note that a $2TBN$ does not define the dependencies between nodes in time slice $t-1$.

The joint probability distribution of a DBN can then be formulated as

$$P(\boldsymbol{X}_{0:T}) = P(\boldsymbol{X}_0) \cdot \prod_{t=1}^{T} \prod_{i=1}^{n} P(X_t^i|Pa(X_t^i)), \quad (11)$$

with $P(\boldsymbol{X}_{0:T}) = P(\boldsymbol{X}_0, \ldots, \boldsymbol{X}_T)$.

As we want to model a network of situations by a DBN, the structure of the network has to fulfill the following assumptions:

- Stationarity: the dependencies within a time slice t and the dependencies between the time slices $t-1$ and t do not depend on t.
- 1st order Markov assumption: the parents of a node are in the same time slice or in the previous time slice.
- Temporal evolution: dependencies between two time slices are only allowed forward in time, i.e., from past to future.
- Time slice structure: The structure of one time slice is a simple Bayesian network, i.e., without cycles.

If any of these assumptions are not fulfilled, the network is not a DBN and inference algorithms could not be applied.

B. Inferring Existence Probabilities

Due to the dependency between elementary and abstract situations and the fact that we can feed the DBN with evidence, i.e., observations, only via the elementary situations, we can calculate the joint probability recursively in time by

$$P(\boldsymbol{S}^*_{0:T}, \boldsymbol{E}_{1:T}) = P(\boldsymbol{S}^*_0) \cdot \prod_{t=1}^{T} P(\boldsymbol{S}^*_t|\boldsymbol{S}^*_{t-1}) P(\boldsymbol{E}_t|\boldsymbol{S}^*_t), \quad (12)$$

where $\boldsymbol{E}$ denotes the set of elementary situations filled with evidences, and $\boldsymbol{S}^*$ denotes all defined situations $\boldsymbol{S}$ in the DBN without the collected evidence nodes, i.e., $\boldsymbol{S}^* = \boldsymbol{S} \backslash \boldsymbol{E}$.

By using this kind of recursive calculation, we can make different calculations over time, which we list in the following. Note that $\widetilde{\boldsymbol{S}}$ is now an arbitrary set of abstract situations.

- Filtering: $P(\widetilde{\boldsymbol{S}}_t|\boldsymbol{E}_{1:t})$ gives a solution to the existence probability of a set of situations $\widetilde{\boldsymbol{S}}$ at the current time,
- Prediction: $P(\widetilde{\boldsymbol{S}}_{t+k}|\boldsymbol{E}_{1:t})$ (with $k > 0$) gives a solution to the existence probability of a set of situations $\widetilde{\boldsymbol{S}}$ in the (near) future,
- Smoothing: $P(\widetilde{\boldsymbol{S}}_k|\boldsymbol{E}_{1:t})$ (with $0 < k < t$) gives a solution to the existence probability of a set of situations $\widetilde{\boldsymbol{S}}$ in the past,
- Most likely explanation: $\text{argmax}_{\widetilde{\boldsymbol{S}}_{1:t}} P(\widetilde{\boldsymbol{S}}_{1:t}|\boldsymbol{E}_{1:t})$ gives a solution to the most likely sequence of situations $\widetilde{\boldsymbol{S}}_{1:t}$.

Due to this modeling, the existence probability of an arbitrary set of abstract situations can be calculated in a recursive way at each point in time. A situation is then represented in the world model, if the corresponding existence probability is larger than an instantiation-threshold. If the existence probability in the next time step is below a deletion-threshold, it is assumed that the situation does not exist any longer and its representation is removed from the world model. This way, the world model tries to keep an up-to-date representation of the existing situations of the real world. However, determining a meaningful instantiation- and deletion-threshold is still an open task.

VII. Generating Situation-Specific DBNs

In this section, we will describe how we generate a DBN from our SDN model in order to be able to apply the aforementioned inference calculations. As we may have modeled a lot of situations in our SDN, and the operator may be only interested in a subset of them, we will generate a DBN with only the necessary nodes. We will term the generated DBN a *situation-specific DBN*, which we adopted from the MEBN-approach presented in [28].

A. SS-DBN Structure

For generating the structure of the SS-DBN, we will make use of the structure of the pre-modeled SDN. In general, we will use exactly the same structure, but only use the situations that are relevant for the selected situations of interest. The general approach is the described in Algorithm 3. Thus, we get a reduced version of our SDN by making use of the predefined abstraction levels. This way, we simply get our SS-DBN structure by selecting the nodes with a lower abstraction level.

Algorithm 3 Generating the SS-DBN structure

Require: SDN
Ensure: DBN-structure
  set **S** as selected situations of interest in the SDN
  while S $\neq \emptyset$ **do**
    for all incoming and outgoing arrows of every situation s in **S** **do**
      if abstraction level of the connected node is lower than the one of s **then**
        add node to DBN
      else
        ignore them
      end if
      set **S** as the set of all added nodes
    end for
  end while
  for all temporal nodes **do**
    add a reflexive arrow
  end for

B. Criteria for good Parameters

The main challenge is now to set the parameters in a way that the network behaves as we would expect it to behave. In the following, we will list some criteria, which the DBN should fulfill:

- *Asymptotic behavior:* How does the DBN behave if we observe the same values over time? Especially if we make no observations at all, we do not know anything about the existence of the situation and the resulting existence probability should be converging to 0.5.
- *Switching behavior:* How does the DBN behave if observations change their values significantly? If we first observe values that support the existence of the situation of interest and then we make the opposite observations, the resulting probabilities of the situation should also change, i.e., the situation should switch its state from true to false.
- *Robustness:* How does the DBN behave if we have wrong or missing observations? As said above, the resulting probability of the situation of interest should not be very sensitive for noise.

Having determined these criteria, we will evaluate several parameter settings with respect to them.

C. SS-DBN Parameters

We have to define parameters for every node, i.e., a priori probabilities for nodes with no incoming arrows, and conditional probabilities for nodes with incoming arrows. We will treat the temporal arrows the same as all other incoming arrows. The conditional probabilities then correspond to $P(X_t^i|Pa(X_t^i))$ in equation (10). As a first rule for setting the parameters, we have the following:

- If the situation has no incoming arrows, i.e., is a root node, the prior probabilities are all set to 0.5.
- If the situation is a temporal situation, we set the probabilities inside the first time point to 0.5.

In many cases, elementary situations have only one incoming arrow. This is due to the fact that at the lowest level, there are the observation values of the DBN and they can always be considered as necessary conditions. Thus, we have to differentiate between two different types: either the parent node is temporal or not. For non-temporal nodes, the higher level situation can be interpreted as a semantic annotation to the observed value and therefore, we can set the conditional probabilities in a deterministic way:

- If the elementary situation A has one incoming arrow from a non-temporal situation B, the probabilities are set deterministic like in Table I.

TABLE I. CONDITIONAL PROBABILITY TABLE (CPT) FOR ELEMENTARY SITUATION WITH NON-TEMPORAL PARENT.

$P(A\|B)$	B	$\neg B$
A	1.0	0.0
$\neg A$	0.0	1.0

If we have a temporal parent, we will apply a non-deterministic conditional probability table. We will evaluate some different parameter settings before choosing them. For this small evaluation, we will only consider two nodes A and B, where the parent B is a temporal node. We are interested in good values for the CPT $P(A|B)$. Thus, we fix the CPT parameters for $P(B_{t+1}|B_t)$. We set the CPT values in a mirrored way, i.e.,

$$P(A|B) = P(\neg A|\neg B) \text{ and } P(\neg A|B) = P(A|\neg B). \quad (13)$$

This is because we do not want to have different influence when observing true or false. Note that we only have to change one value for our evaluation, namely $p = P(A|B)$, as $P(\neg A|B) = 1 - P(A|B)$. For the temporal node, we fix the value $P(B_{t+1}|B_t) = 0.9$, see right side in Table II.

We evaluate five different values for p, and we show three different results, namely the probability $P(B|A)$ over ten time steps. Figure 9 shows the resulting probability for similar observations, switching observations, and for some false observations. For the first observation sequence (Figure 9a), we would like to result in a high probability, so we discard $p = 0.6$. For the second one (Figure 9b), we would like to have a probability, that switches, but not rapidly, so we discard $p = 0.99$. For the third sequence (Figure 9c), we would like to have a probability that is not too sensitive to false observations, so we finally choose $p = 0.7$. We therefore have the following rule:

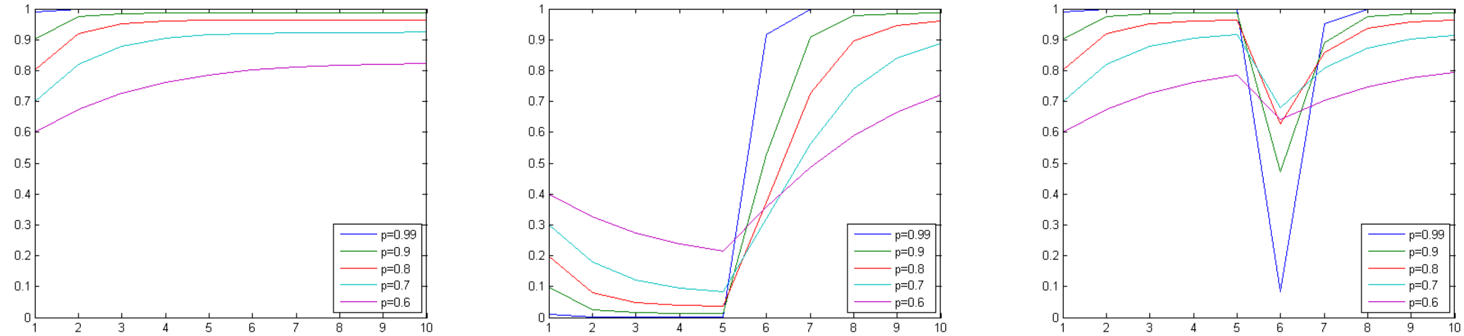
- If the elementary situation A has one incoming arrow from a temporal situation B, the probabilities are set like the ones on the left side in Table II.

TABLE II. CPT FOR ELEMENTARY SITUATION WITH TEMPORAL PARENT (LEFT SIDE) AND CPT FOR TEMPORAL PARENT (RIGHT SIDE).

$P(A\|B)$	B	$\neg B$
A	0.7	0.3
$\neg A$	0.3	0.7

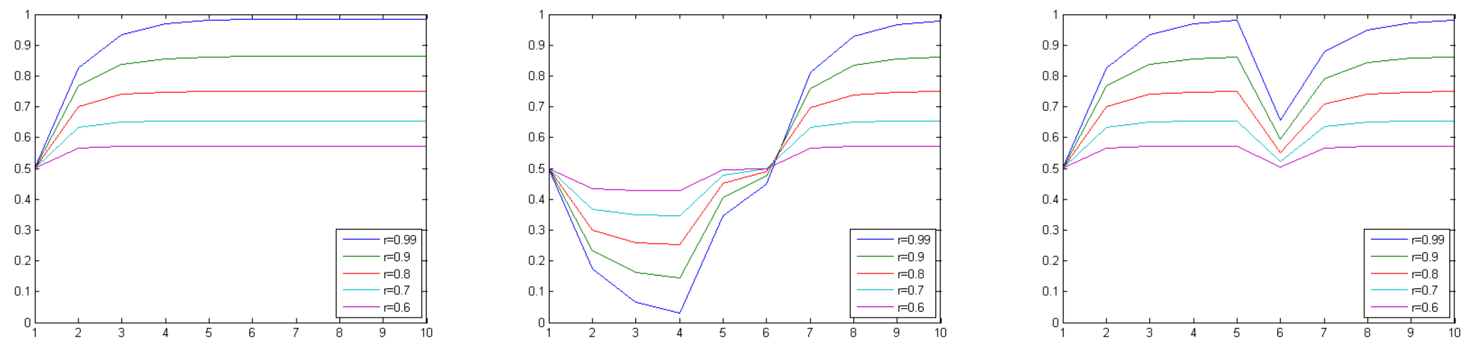
$P(B_{t+1}\|B_t)$	B_t	$\neg B_t$
B_{t+1}	0.9	0.1
$\neg B_{t+1}$	0.1	0.9

For all abstract situations, we apply an approach that uses a weighted CPT-construction. Let Y be the abstract situation, and $X_1, \ldots, X_k$ the situations with an arrow to Y.



(a) $A = 1111111111$ (b) $A = 0000011111$ (c) $A = 1111101111$

Fig. 9. $P(B|A)$ over ten time steps with different observation sequences.



(a) $A = 1111111111$, $B = 1111111111$ (b) $A = 0000111111$, $B = 0000001111$ (c) $A = 1111101111$, $B = 1111111111$

Fig. 10. $P(C|A, B)$ over ten time steps with different observation sequences.

For this approach, we have to determine two different types of parameters.

- Relative influence of variables: This is modeled by different weights λ_i with $\sum_{i=1}^{k} \lambda_i$.
- Absolute influence of variables: This is modeled by a stretch value $r \in [0.5, 1]$.

For the relative influence, we can use a weighting of the influence, namely

$$P(Y = 1|X_1 = x_1, \ldots, X_k = x_k) = \sum_{i=1}^{k} \lambda_i x_i. \quad (14)$$

Then it is $P(Y = 1|X_1 = x_1, \ldots, X_k = x_k) \in [0, 1]$ and $P(Y = 1|X_1 = x_1, \ldots, X_k = x_k) = 0$ for $x_i = 0$ and $P(Y = 1|X_1 = x_1, \ldots, X_k = x_k) = 1$ for $x_i = 1$.

The aim of the absolute influence is to reduce the interval $[0, 1]$ to the values of $[1 - r, r]$. Thus, the overall strength of the variables should be reduced. We define

$$f(x) = (2r - 1) \cdot x + (1 - r). \quad (15)$$

Then it is $f(x) \in [1 - r, r]$ for $x \in [0, 1]$. Thus, we can apply f on $P(Y = 1|X_1 = x_1, \ldots, X_k = x_k)$ and we have $P(Y = 1|X_1 = x_1, \ldots, X_k = x_k) \in [1 - r, r]$. In summary, we have to determine $k + 1$ parameters, namely $\lambda_1, \ldots, \lambda_k, r$, instead of 2^{k-1} for the CPT of $P(Y = 1|X_1 = x_1, \ldots, X_k = x_k)$.

We will now show a small evaluation of our example with different values of r. The values of r are chosen with steps 0.1. As 0.5 and 1.0 would not be reasonable values, we chose the values 0.6, 0.7, 0.8, 0.9, and 0.99. We will use the same criteria as above, namely insert observation sequences that represent similar observations, switching observations, and for some false observations. The result is shown in Figure 10. For the first observation sequence (Figure 10a), we would like to result in a high probability, so we discard $r = 0.6$ and $r = 0.7$. For the second one (Figure 10b), we would like to have a probability, that switches, but not rapidly, so we discard $p = 0.99$. For the third sequence (Figure 10c), we would like to have a probability that is not too sensitive to false observations, so we finally choose $p = 0.9$. We can now state the final rule for setting the parameters:

- If the situation is on a higher level, the weighted CPT-construction is applied. The weights of the influences can be adapted due to the semantics, and it is suggested to set the stretch value to $r = 0.9$.

VIII. Application Example in the Maritime Domain

Assume for example a security officer who is using a maritime surveillance system located in a port on an island and he is interested in detecting vessels, which are suspicious smuggling vessels. There is also a suspicious zone next to the island, in which a lot of smuggling activities recently happened. The officer is able to formulate several characteristics that lead to a higher probability of a smuggling vessel, either if the vessel is incoming or outgoing. Note that these characteristics are defined as situations itself.

Based on the different situations, the expert is able to model an SDN, as depicted in Figure 11. A sketch of such situations is visualized in Figure 12. Let us give some examples for our modeling approach. In Figure 11, the situation `Has unknown ID` has the necessary characteristic situation `MMSI-Number is empty`. The situation `Sends no AIS signal` has the necessary characteristic situations `MMSI-Number is empty` and `Is inside AIS receiver area`. And the situation `Was in suspicious area` is a sufficient characteristic situation for the situation `Suspicious incoming smuggling vessel`. Thus, the existence of a sufficient situation should lead to a higher probability of the existence of the situation of interest, whereas the existence of the necessary situations has to be fulfilled for inferring the existence of the situation of interest.

Note that the arrows in the SDN are always pointing from a situation that describes some characteristics of the situation that is pointed to, either by a necessary or sufficient condition. The arrows of a necessary condition are always pointing from a higher level of abstraction to a lower level of abstraction and the arrows of a necessary condition vice versa. The level of abstraction of a single situation of interest is determined by the structure of the whole SDN. At the lowest level of abstraction, we only have our elementary situations that can be inferred as true or false directly from attribute values of objects or from geometric computations. An example is the situation

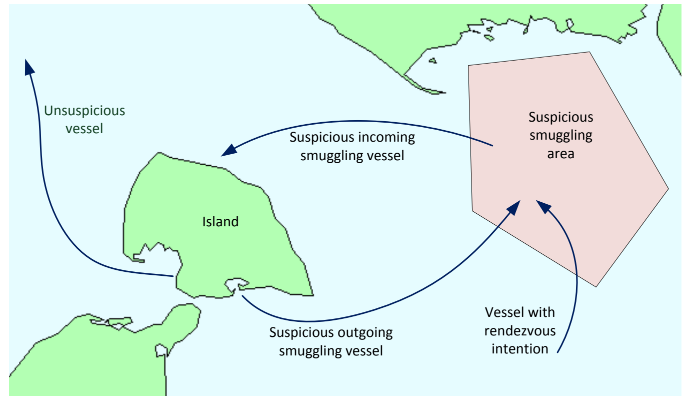


Fig. 12. A sketch of the scenario for incoming and outgoing smuggling vessels.

`Past in polygon check` that checks, if any point of the vessel's path is inside the polygon of the suspicious area. Thus, at the lowest level, and only there, the observations are fed into network. Every situation with a direct connection, either incoming or outgoing arrows, is moved to the next higher level of abstraction. Based on these connections, we result in the SDN, as shown in Figure 11.

We will now evaluate the behavior of the whole network. We construct the DBN with the approach described above and set the parameters with the rules we established. We evaluate two different scenarios. In the first scenario the observations should lead to the decision of a suspicious outgoing smuggling vessel, the second one should indicate a suspicious incoming smuggling vessel.

For the first scenario, we will assume some observations that are not noisy, namely

- `past point in polygon check`=0 for all time points, i.e., the vessel was not in a suspicious area,

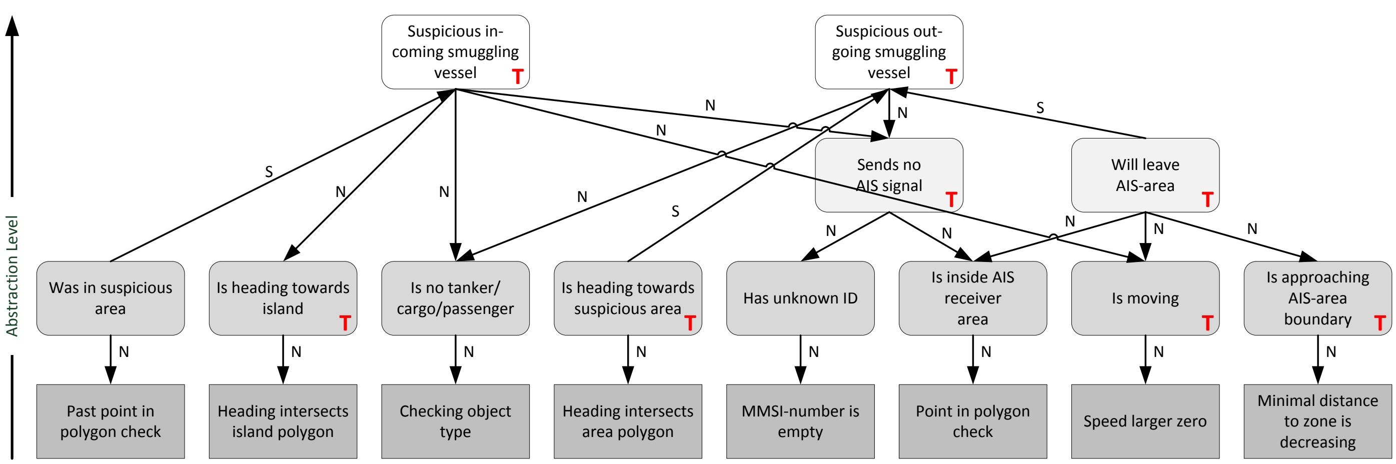


Fig. 11. A SDN with two modeled situations of interest: suspicious incoming and suspicious outgoing smuggling vessel.

- `checking object type`=1 for all time points, i.e., the vessel is no tanker/cargo/passenger vessel,
- `MMSI-Number is empty`=1 for all time points, i.e., vessel has an unknown ID,
- `point in polygon check`=1 for all time points, i.e., vessel is inside AIS receiver area.

We will assume the following noisy observations:

- `heading intersects area polygon`=1 with a certain probability for all time points, i.e., the vessel is heading towards suspicious area,
- `heading intersects island polygon`=0 with a certain probability for all time points, i.e., the vessel is not heading towards the island,
- `speed larger than zero`=1 with a certain probability for all time points, i.e., the vessel is moving,
- `distance to zone is decreasing`=1 with a certain probability for all time points, i.e., the vessel is approaching AIS area boundary.

We evaluated the DBN with different observation probabilities, i.e., different amount of noise in the observation data. Figure 13a), b), and c) shows the result where the probability of wrong observations is 0.1, 0.3, and 0.5, respectively.In the second scenario, where the vessel is an incoming smuggling vessel, we assume the following deterministic observations:

- `past point in polygon check`=1 for all time points, i.e., the vessel was in a suspicious area,
- `checking object type`=1 for all time points, i.e., the vessel is no tanker/cargo/passenger vessel,
- `MMSI-Number is empty`=1 for all time points, i.e., vessel has an unknown ID,
- `point in polygon check`=0 for all time points, i.e., vessel is not inside AIS receiver area.

We assume the following noisy observations:

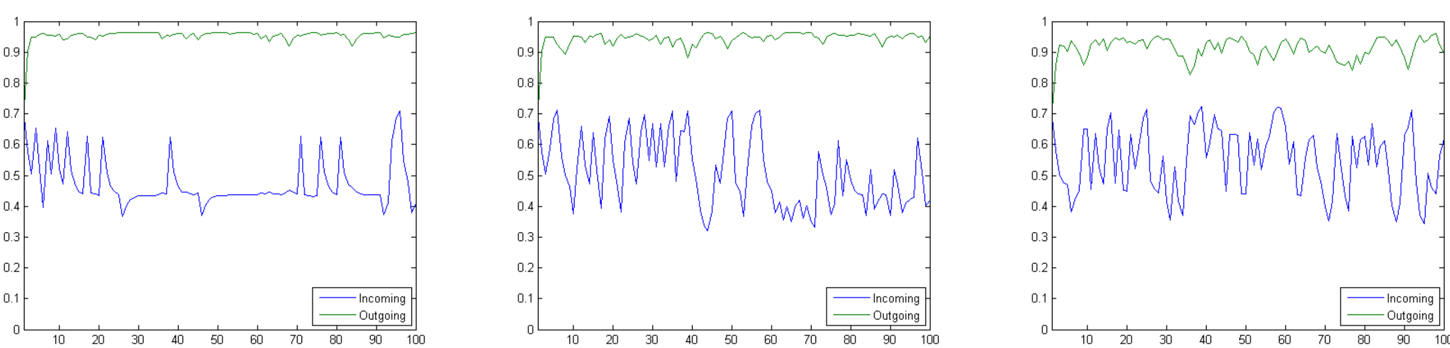
- `heading intersects area polygon`=0 with a certain probability for all time points, i.e., the vessel is not heading towards suspicious area,
- `heading intersects island polygon`=1 with a certain probability for all time points, i.e., the vessel is heading towards the island,
- `speed larger than zero`=1 with a certain probability for all time points, i.e., the vessel is moving,
- `distance to zone is decreasing`=1 with a certain probability for all time points, i.e., the vessel is approaching the boundary of the AIS-area.

Like in the first scenario, we present the results with different observation noise in Figure 14. In both scenarios, we see that the two situations of `suspicious incoming smuggling vessel` and `suspicious outgoing smuggling vessel` can be clearly distinguished from each other, even if we add noisy observations with a probability of 0.5. In both cases, the probability of the underlying situation is around 0.9 or higher, whereas the probability of the other situation is much lower. Of course, if we add more noise, the probability of situation that is not true is more unstable over time. The values range from around 0.4 to 0.7. The reason that the probability is not close to zero is that we have observations that support both situations. Thus, we have contradictory observations for the situation that is not true, which results in probability values between 0.4 and 0.7.

IX. Conclusion and Future Work

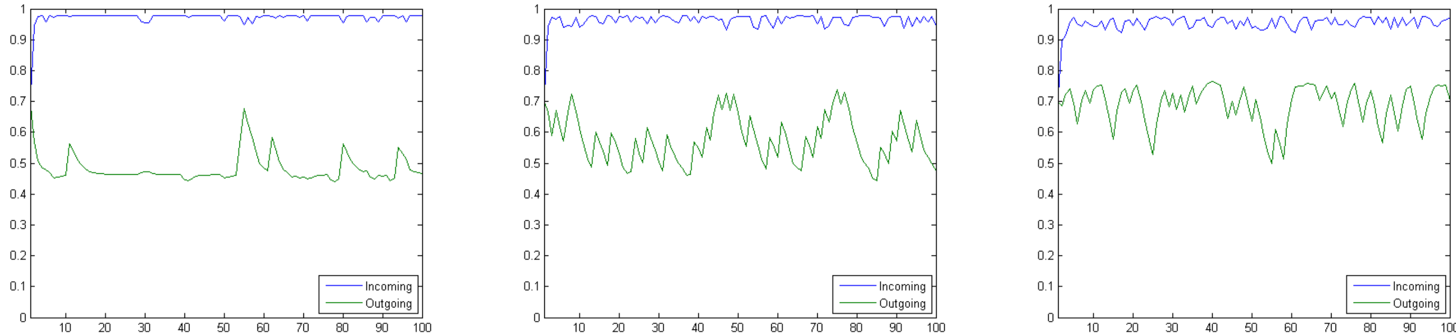
In this article, the information flow in an intelligent surveillance system was highlighted. We described the process of situation assessment in detail and showed how it can be included into the information flow.

The main contribution of this work was the establishment of systematic approach to characterize and recognize situations of interest in a probabilistic way. The focus hereby was on a top-down approach, i.e., that a maritime expert is able to model the situations of interest by necessary and sufficient conditions regarding other situations. The result of the characterization



(a) Probability of wrong observation: 0.1 (b) Probability of wrong observation: 0.3 (c) Probability of wrong observation: 0.5

Fig. 13. Outgoing smuggling vessel scenario: probabilities of incoming and outgoing smuggling vessel over 100 time steps with different observation noise.



(a) Probability of wrong observation: 0.1

(b) Probability of wrong observation: 0.3

(c) Probability of wrong observation: 0.5

Fig. 14. Incoming smuggling vessel scenario: probabilities of incoming and outgoing smuggling vessel over 100 time steps with different observation noise.

process is a Situational Dependency Network (SDN), of which a Dynamic Bayesian Network (DBN) can be generated automatically. For the generation of the DBN, we presented several rules for determining the parameters. During the process of recognizing situations, the DBN uses observation values of so-called elementary situations and is able to determine the probability of more abstract situations over time by using well-known efficient inference methods.

Finally, an application example of a SDN in the maritime domain was given. We generated a DBN by using the established rules and evaluated the DBN by using noisy observations. Especially, we showed that the network behaves as a user would expect. By using this approach, the operator would be able to define situations of interest by himself and to perform a probabilistic situation assessment without the use of training data. Future work includes a refinement of the parameter settings and an evaluation with real data.

ACKNOWLEDGMENT

The work was conducted during a research visit at Ningbo University (China), which was supported by Karlsruhe House of Young Scientists (KHYS).

REFERENCES

[1] Y. Fischer and J. Beyerer, "A top-down-view on intelligent surveillance systems," in *Proc. of the 7th International Conference on Systems (ICONS)*. IARIA, 2012, pp. 43–48.

[2] M. R. Endsley, "Towards a theory of situation awareness in dynamic systems," *Human Factors*, vol. 37, no. 11, pp. 32–64, 1995.

[3] Automatic Identification System (AIS) provided by the International Maritime Organization (IMO), last access: 08.12.2013. [Online]. Available: http://www.imo.org/OurWork/Safety/Navigation/Pages/AIS.aspx

[4] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., 2006.

[5] K. P. Murphy, *Machine Learning: A Probabilistic Perspective (Adaptive Computation and Machine Learning series)*. The MIT Press, 2012.

[6] M. Andersson and R. Johansson, "Multiple sensor fusion for effective abnormal behaviour detection in counter-piracy operations," in *Proc. of the 2nd International Conference on Waterside Security*, 2010, pp. 1–7.

[7] D. Meyer-Delius, C. Plageman, and W. Burgard, "Probabilistic situation recognition for vehicular traffic scenarios," in *Proc. of the IEEE Int. Conference on Robotics and Automation*, 2009, pp. 459–464.

[8] D. L. Hall and S. A. H. McMullen, *Mathematical Techniques in Multisensor Data Fusion*. Artech House, Inc., 2004.

[9] M. Baum, I. Gheta, A. Belkin, J. Beyerer, and U. D. Hanebeck, "Data association in a world model for autonomous systems," in *Proc. of the 2010 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2010, pp. 187–192.

[10] A. Dore, M. Soto, and C. S. Regazzoni, "Bayesian tracking for video analytics: An overview," *IEEE Signal Processing Magazine*, vol. 27, no. 5, pp. 46–55, 2010.

[11] A. Bauer, T. Emter, H. Vagts, and J. Beyerer, "Object oriented world model for surveillance systems," in *Future Security: 4th Security Research Conference*. Fraunhofer Press, 2009, pp. 339–345.

[12] A. N. Steinberg, C. L. Bowman, and F. E. White, "Revisions to the JDL data fusion model," in *Sensor Fusion: Architectures, Algorithms, and Applications, Proceedings of the SPIE Vol. 3719*, 1999, pp. 430–441.

[13] J. Moßgraber, F. Reinert, and H. Vagts, "An architecture for a task-oriented surveillance system: A service- and event-based approach," in *Fifth International Conference on Systems (ICONS)*, 2010, pp. 146–151.

[14] Y. Fischer and A. Bauer, "Object-oriented sensor data fusion for wide maritime surveillance," in *Proc. of the 2nd International Conference on Waterside Security (WSS)*, 2010, pp. 1–6.

[15] J. Ye, S. Dobson, and S. McKeever, "Situation identification techniques in pervasive computing: A review," *Pervasive and Mobile Computing*, vol. 8, no. 1, pp. 36 – 66, 2012.

[16] A. Dahlbom, "Petri nets for situation recognition," Ph.D. dissertation, Örebro University, Sweden, 2011, last access: 08.12.2013. [Online]. Available: http://oru.diva-portal.org/smash/record.jsf?pid=diva2:384609

[17] J. IJsselmuiden, A.-K. Grosselfinger, D. Münch, M. Arens, and R. Stiefelhagen, "Automatic behavior understanding in crisis response control rooms," in *Proc. of International Joint Conference on Ambient Intelligence*, 2012, pp. 97–112.

[18] M. Kokar and M. Endsley, "Situation awareness and cognitive modeling," *Intelligent Systems, IEEE*, vol. 27, no. 3, pp. 91 –96, 2012.

[19] M. Kokar, C. Matheus, and K. Baclawski, "Ontology-based situation awareness," *Information Fusion*, vol. 10, no. 1, pp. 83–98, 2009.

[20] J. Garcia, J. Gomez-Romero, M. Patricio, J. Molina, and G. Rogova, "On the representation and exploitation of context knowledge in a harbor surveillance scenario," in *Proceedings of the 14th International Conference on Information Fusion (FUSION)*, 2011, pp. 1–8.

[21] A. C. Van den Broek, R. M. Neef, P. Hanckmann, S. P. Van Gosliga, and D. Van Halsema, "Improving maritime situational awareness by fusing sensor information and intelligence," in *Proceedings of the 14th International Conference on Information Fusion (FUSION)*, 2011, pp. 1–8.

[22] A. Vandecasteele and A. Napoli, "An enhanced spatial reasoning ontology for maritime anomaly detection," in *7th International Conference on System of Systems Engineering (SoSE)*, 2012, pp. 1–6.

[23] J. Roy and M. Davenport, "Exploitation of maritime domain ontologies for anomaly detection and threat analysis," in *International Waterside Security Conference (WSS)*, 2010, pp. 1–8.

[24] G. Pilato, A. Augello, M. Missikoff, and F. Taglino, "Integration of

ontologies and bayesian networks for maritime situation awareness," in *IEEE Sixth International Conference on Semantic Computing (ICSC)*, 2012, pp. 170–177.

[25] M. Kruger, L. Ziegler, and K. Heller, "A generic bayesian network for identification and assessment of objects in maritime surveillance," in *15th International Conference on Information Fusion (FUSION)*, 2012, pp. 2309–2316.

[26] W3C Web Ontology Language, last access: 08.12.2013. [Online]. Available: http://www.w3.org/2004/OWL

[27] P. C. G. Costa, "Bayesian semantics for the semantic web," Ph.D. dissertation, School of Information Technology and Engineering, George Mason University. Fairfax, VA, USA, 2005, last access: 08.12.2013. [Online]. Available: http://digilib.gmu.edu/dspace/handle/1920/455

[28] K. B. Laskey, "MEBN: A language for first-order bayesian knowledge bases," *Artificial Intelligence*, vol. 172, no. 2-3, pp. 140–178, 2008.

[29] S. C. Dinkel, W. Hafner, P. C. G. Costa, and S. Mukherjee, "Service-based situational awareness on the semantic web," in *Proc. of the IEEE 3rd Int. Multi-Disciplinary Conf. on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*, 2013, pp. 249–256.

[30] R. Carvalho, R. Haberlin, P. Costa, K. Laskey, and K. Chang, "Modeling a probabilistic ontology for maritime domain awareness," in *Proc. of the 14th Int. Conference on Information Fusion*, 2011, pp. 1 –8.

[31] K. B. Laskey, R. Haberlin, R. N. Carvalho, and P. C. G. Costa, "PR-OWL 2 case study: A maritime domain probabilistic ontology," in *Proceedings of the 6th International Conference on Semantic Technologies for Intelligence, Defense, and Security (STIDS 2011).*, 2011, pp. 76–83.

[32] Y. Fischer, A. Bauer, and J. Beyerer, "A conceptual framework for automatic situation assessment," in *Proc. of the IEEE 1st International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*, 2011, pp. 234–239.

[33] Y. Fischer and J. Beyerer, "Acceptance of automatic situation assessment in surveillance systems," in *Proc. of the IEEE 2nd International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*, 2012, pp. 324–331.

[34] Y. Fischer and J. Beyerer, "Defining dynamic bayesian networks for probabilistic situation assessment," in *Proc. of the 15th International Conference on Information Fusion*, 2012, pp. 888–895.

[35] Y. Fischer and J. Beyerer, "Ontologies for probabilistic situation assessment in the maritime domain," in *Proc. of the IEEE 3rd International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*, San Diego, USA, feb 2013, pp. 105–108.

[36] K. P. Murphy, "Dynamic bayesian networks: Representation, inference and learning," Ph.D. dissertation, University of California, Berkeley, 2002, last access: 08.12.2013. [Online]. Available: http://www.cs.ubc.ca/˜murphyk/Thesis/thesis.html

[37] GeNIe & SMILE, last access: 08.12.2013. [Online]. Available: http://genie.sis.pitt.edu/

[38] Bayes Net Toolbox by Kevin Murphy, last access: 08.12.2013. [Online]. Available: https://code.google.com/p/bnt/

[39] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.