Fusing uncertain knowledge and evidence for maritime situational awareness via Markov Logic Networks

Lauro Snidaro[a], Ingrid Visentini[b], Karna Bryan[b]

[a]*University of Udine*
Department of Mathematics and Computer Science, via Margreth 3
33100 Udine, Italy
[b]*NATO STO Centre for Maritime Research and Experimentation*
viale San Bartolomeo 400
19126 La Spezia, Italy

Abstract

The concepts of event and anomaly are important building blocks for developing a situational picture of the observed environment. We here relate these concepts to the JDL fusion model and demonstrate the power of Markov Logic Networks (MLNs) for encoding uncertain knowledge and compute inferences according to observed evidence. MLNs combine the expressive power of first-order logic and the probabilistic uncertainty management of Markov networks. Within this framework, different types of knowledge (e.g. a priori, contextual) with associated uncertainty can be fused together for situation assessment by expressing unobservable complex events as a logical combination of simpler evidences. We also develop a mechanism to evaluate the level of completion of complex events and show how, along with event probability, it could provide additional useful information to the operator. Examples are demonstrated on two maritime scenarios of rules for event and anomaly detection.

Keywords: Context-based fusion, Situational Awareness, Uncertainty management, Markov Logic Networks

1. Introduction

State-of-the-art situation assessment (SA) systems (e.g. an automatic surveillance system [1]) are able to deal with vast amounts of data and information also of a heterogeneous kind. Their goal is to provide a constantly updated situational picture about the observed environment or set of entities to an operator in order to facilitate human decision making. Updating the current system representation of the situation is generally performed by acquiring, through sensors or other sources of information, new observations which provide a possibly incomplete and uncertain view.

Currently, low-level sensory data is the main source of information used to understand the observed evolving scenario and to identify anomalous conditions; in particular, up to now maritime surveillance heavily relies on the Automatic Identification System (AIS), coastal radars, space-based imagery, and other sensors, to form a picture in which the operator can recognize complex patterns and make decisions [2, 3].

Anomaly detectors or event recognition systems for maritime situational awareness are presented in [4, 2, 5, 6, 7, 8, 9, 10]. The common thread that unites these works is the definition of an expert system, that aims at detecting a set of anomalous behaviours or potential threats. Subject matter experts define a knowledge base (KB), which comprises the possible abnormal patterns the target could follow; then, on the top of it, a

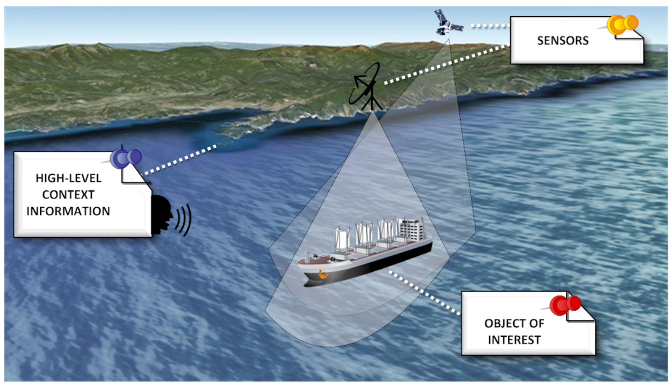


Figure 1: Illustration of an ideal maritime situational awareness situation. The sensory data for an object of interest must be coupled by high-level contextual information.

reasoning engine queries the occurrence of an anomaly for a target object in an arbitrary time instant. For example, in [2] AIS data is used for extracting statistical behaviours of motion patterns, while in [5] situational awareness is achieved fusing knowledge-based detection with data-driven anomaly detection. In [4] a comprehensive literature survey of the anomaly detection process via data analysis is presented; definitions of anomaly and normalcy, explored under the light of decision making systems, are given in order to support the analytical reasoning process.

The main goal of a reasoning engine or probabilistic inference system is to associate a posterior probability distribution to

Email addresses: `lauro.snidaro@uniud.it` (Lauro Snidaro), `visentini@cmre.nato.int` (Ingrid Visentini)

a set of queries [11], given observed evidence. The incorporation of abductive/inductive and deductive inferencing processes is a vital element in an automatic fusion system, and it represents a fundamental step for situational awareness. How this involvement can be obtained, on both theoretical and applicative levels, is a crucial point, and is subject of ongoing research [12].

The reasoner is usually fired by low-level observations provided by sensors, covering in this way the majority of abnormal situations in the domain; however, it is interesting to notice how anomalous behaviours do not always follow standard trends or well-known patterns, especially if related solely to vessels movements, but sometimes they take the form of seemingly unrelated activities on a larger scale [13]. Ship-centric focus should be replaced by a broader vision, where the ideal situational awareness system should then be flexible and adaptive enough to integrate both low-level and high-level information, detecting anomalous or suspicious conditions by reasoning on manifest or uncertain data, but also on (apparently irrelevant) relations among objects, which may reveal unobserved coincidences. The maritime domain is a daunting scenario for testing such systems, because of many factors: its challenging nature where the coverage of wide areas is given by discontinuous and intermittent sensory data, its well-known commercial policies and practices which can suggest normalcy behaviour patterns, the presence of local contextual information, stable in time, which can depict alternative indicators of multi-layered situations, and the urgency for systems capable to provide effective and advanced warning to promote countermeasures to illicit activities.

The integration of contextual knowledge, as demonstrated in [14, 15, 16] where it is exploited for improving tracking accuracy, can greatly enhance the performance of an awareness system. Despite its value, the representation and use of context is often poorly integrated, if not absent, even if the richness and completeness of this information is extremely useful to properly interpret the available stream of raw sensor data from a multitude of points of view (security, safety, economical or environmental situation, etc.). Qualitative high-level knowledge can help to infer about hidden states from low-level data generated by sensors, other fusion processes or human reports. In other words, context is a powerful means to picture a broader and deeper operational situation, as it can reduce uncertainties where normally analysts would need to be consulted.

In this paper, we exploit MLNs to encode uncertain knowledge, fuse data coming from multiple (and possibly heterogeneous) sources, and perform reasoning on incomplete data. One key point of using the MLNs for reasoning is their ability to reason with incomplete or missing evidence, which is a crucial feature hardly found in other approaches, but sought after especially in the maritime domain, where the data is often inaccurate, delayed or simply not available. Another advantage with respect to other systems, is the fact that MLNs support inconsistencies or contradictions in the knowledge base, which is a problem when different experts provide contributes to it. This avoids non-trivial knowledge engineering techniques to be performed in order to guarantee rules consistency. Here we use Markov Logic Networks (MLNs) to detect two possible anomalous conditions in maritime domain, a rendezvous at sea and a hazardous combination of cargo ships in a harbour.

We use exemplary scenarios, the first one derived from experts' suggestions gathered at the NATO STO Centre for Maritime Research and Experimentation and the second one expanded from [17], to highlight how unobserved complex events could be built by logical combination of simpler evidence, and how contextual information is extremely valuable in many conditions. MLNs present advantages suited to our domain as they support reasoning with missing or partial observations (incomplete evidence), they allow to encode expert rules and relational knowledge with an associated degree of uncertainty, they are able to handle contradictions and inconsistencies [18].

Preliminary investigation on MLNs in maritime domain has been initiated in [19], where we leveraged the expressive power of first-order logic (FOL) and the probabilistic uncertainty management of Markov networks in order to detect anomalies via reasoning on uncertain knowledge. Here we aim to expand and refine that work by providing contributions for:

- clarifying the concepts of event (simple and complex) and anomaly in the scope of fusion terminology;
- explicitly explaining how simple and complex events can be encoded in the form of FOL formulas with associated degree of uncertainty in maritime domain;
- demonstrating how MLNs could provide a powerful tool for fusing heterogeneous sources (e.g. a priori, contextual, sensory, etc.) of information for situation assessment by being able to express unobserved complex events by logical combination of simpler evidences;
- developing a mechanism to evaluate the level of completion of complex events as this calculation is not directly solvable within the MLNs framework.

1.1. Terminology

To facilitate human decision making, an updated situational picture of the observed environment assessing the current state of domain entities and their relationships is required. Events and anomalies can be considered fundamental building blocks for developing such a picture of the environment. In this section, we provide the necessary definitions of these concepts and relate them to the JDL fusion model [12]. In the following, the term *level* will be used as per JDL terminology.

While there are many papers in the literature that deal with events and provide various definitions [20], we here breakdown the main concepts in light of the typical functionality and requirements of a SA system. An event modelling framework in maritime domain was recently presented in [21], where a piracy example is presented with the intent of facilitating the decision making process, but no reasoner is associated to the graphical representation of events.

For our purposes an **event** is a "significant occurrence or happening". It can be subdivided in *simple*, when we consider the

Table 1: Examples of events and anomalies at different JDL levels

JDL Level	Event	Type	Anomaly
0	Absence of AIS signal	simple	AIS off
1	Vessel increased speed	simple	Vessel speed over limit
2	Vessel X stopped, Vessel Y stopped, Vessel X and Y are close	complex	Vessel X and Y are having a rendezvous

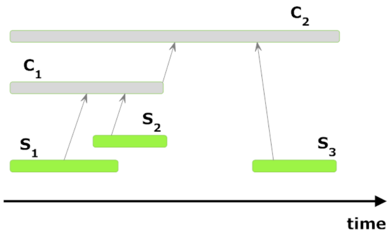


Figure 2: Example of the detection of a complex event by observing the occurrence of its components. C_2 is composed by complex event C_1 and simple event S_3. All the events in this example are non-instantaneous as each of them spans a certain interval of time.

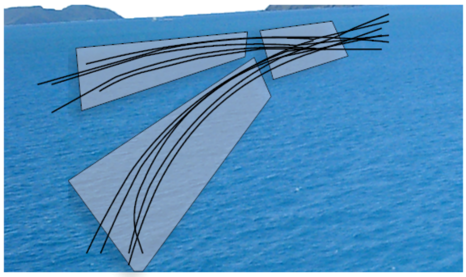


Figure 3: Implicit knowledge modelling. Exemplification of ship trajectories being clustered [23] to model normal movements patterns. A trajectory leaving known clusters would be flagged as anomalous.

variation of a quantity or state, or *complex*, which is a combination of atomic or complex activities [20]. Figure 2 gives an intuitive representation of the idea.

A **simple event**, is any significant variation of input data, at any level, discernible by the system. Also called *atomic* in the literature, we here use the term *simple* to avoid confusion with *ground atoms* defined in Section 2. They can be directly observable or not, and can be either instantaneous or last for an arbitrarily long period of time. As the name implies, this is the most basic type of occurrence and cannot be further decomposed into simpler constituting events.

More in general, variations of input signals (Level 0), of a target's state (e.g. speed, direction, etc. that can be included in Level 1), of a target's relation with other entities (Level 2), are all examples of simple events.

Complex events are a combination of two or more component events (simple or complex) that can be arbitrarily combined through logical operators ($\wedge$, $\vee$, $\neg$) to encode articulated expert and domain knowledge. Complex events can be either triggered by a specific time-ordered sequence of component events, or be just an unordered collection of them. In addition, a complex event can be composed by a heterogeneous combination of events generated by data at different levels. Generally, complex events span a certain interval of time. They could have a fixed time-frame, that is constituting events have to occur within a given time window, or not as they wait indefinitely for all the component events to happen.

Simple and complex events will be represented in Section 2 by predicates, with the difference being that a complex event will appear only as the consequent of an implication (i.e. on the right side of an implication) as it cannot be directly observable but only inferable from the detection of its components.

An **anomaly** can be considered a critical event to which the system is generally called to react to. Usually, a *threshold* establishes if input data can be considered unexpected or anomalous, thus raising an exception. Thresholds, provided by domain experts or learned automatically by the system from data, are therefore used to immediately spot an anomalous condition. However, anomalies provide no notion whatsoever on the meaning of the exceptional input. Following this definition, an anomalous event is an occurrence of some type that deviates from expected values or behaviour. In a Situation Assessment system, the knowledge base is consulted to infer a possible conclusion from the anomalous condition.

An exhaustive description of anomalies taxonomy in maritime domain can be found in [22], where the author provides a clear and comprehensive classification of possible events of interest, grouping them by kinematic and non-kinematic patterns and providing an ontology of possible anomaly causes.

Events and anomalies can be defined by **explicit or implicit models** in the system. In the former case the model encodes, usually exploiting expert and contextual knowledge, the complete description of what an (anomalous) event is. On the contrary, implicit modelling means that samples of activities are unsupervisedly learned by the system in order to detect a deviation from common patterns. Trajectory clustering (as shown in Figure 3) is an example of technique for extracting and maintaining (low-level) knowledge about normal movement patterns [23].

"High (low)-level event" or "High (low)-level anomaly" is something, to the best of our knowledge, never properly formalized. Following the description given in the above sections and taking into account JDL levels, our position here is that whenever the system detects any appreciable variation of input data of any level, a corresponding event is generated. Table 1 shows some examples of events generated from data at different levels. For instance, the detection of presence or absence of

AIS signal is something that can be considered at the bottom of the JDL hierarchy, while the speed of a vessel is a feature of its state and belongs to Level 1. Two stopped vessels very close out at sea is a relation between two entities and helps defining the current situation (JDL level 2).

It is not true then, that, to flag a situation as anomalous, data and information have to bubble up through the levels following increasing processing and refinement steps. Anomalies can be generated from data of every kind and level as shown in Table 1. For example, the absence of AIS signal can be directly considered something anomalous, as well as a speeding boat or a rendezvous out at sea. Also, anomalies could be generated both from simple and complex events as Table 1 exemplifies.

1.2. Related work on reasoning systems

Expert systems, also known as rule-based systems, are often used to represent high-level contextual information and to describe the events to be detected [9]. Simple if-then-else rules have the advantage to be easy to code and extremely effective to flag a suspicious event or anomaly. Unfortunately, in most of the cases the reasoning engine is not refined enough, but simply the result of an binary process that may lead to drastic decisions with no degrees of incertitude. In a step forward, uncertainty can be coupled to the rules in the knowledge base as in the case of MYCIN [24], or can be interpreted as probabilities when Bayes' rule is used as the basis of inference, as in Prospector [25]. A major drawback of these systems is that rule-based systems act as a monolithic chain that triggers the rules only when complete evidence is available. Whereas a more natural behaviour would be to infer (abduct or deduct), with different degrees of information quality or reliability [26], the missing pieces of information from a priori knowledge to draw a general picture even in absence of direct observations.

An improvement is provided by Description Logics, which represent a formalization of Semantic Networks as an extension of classical logic and are very useful for intuitively representing knowledge. Although sound, they do not offer support for uncertainty, even if the structure graph enables the support of multiple hypotheses. From the need for supporting uncertainty and vagueness for reasoning under probabilistic uncertainty in ontologies, probabilistic and fuzzy description logics have stemmed, extending classical DLs to deal with numerical probabilities or fuzzy truth values [27].

Dealing with uncertainty is one of the most desirable characteristics for a fusion system [28], as uncertain data affects decisions and the quality of estimates. Uncertainty is defined as the lack of exact knowledge, which would allow us to formulate a reliable conclusion. Uncertainty is generated when logic fails, according to Russell and Norvig [11], because laziness or theoretical or practical ignorance are introduced in the modellization of the problem. In particular, laziness refers to the will to model the domain with less rules than necessary, while ignorance occurs when the theory is lacking in some respect or part of the data is missing. Probability theory provides a way to overcome and represent the uncertainty that derives from ignorance; on this side, Bayesian networks provide a tractable solution. Already extensively used in surveillance domain (see [29] for a recent survey), in [6] they have been used for assessing the threat probability obtained by the combination of five types of anomalies or abnormal behaviours, that are deviation from standard routes, unexpected AIS activity, unexpected port arrival, close approach and zone entry. Despite being so largely used for probabilistic representations of uncertain knowledge, Bayesian networks have strong limitations, including the fact that they allow reasoning about the same fixed number of attributes, as their nature is essentially propositional: the set of random variables is fixed and finite, and each has a limited domain [11]. As result, their application to complex problems is often impeded, as they require to define in advance with confidence how many entities will be involved, and what type of relationships intercur among them. Even in Hidden Markov Models, which have been used in case of temporally and spatially distributed observations for event recognition [30, 31], the number and type of states must be specified in advance. This last condition largely impacts on performances when scaling up to a larger size scenario, reducing their applicability in a flexible and uncertain domain as the maritime.

Ontologies are another popular means to encode knowledge and represent relationship among entities [31, 32, 7, 33]. In [10], a rule-based system is coupled with ontologies for automatic discovery of anomalies in maritime domains. An attempt to integrate taxonomical knowledge through OWL ontologies and rule-based knowledge through SWRL rules is described in [20] for video surveillance. While the paper presents an actionable solution by exploiting freely available software tools and libraries, on one hand it lacks the full expressiveness of FOL as SWRL provides only a limited support for zeroth-order logic. On the other hand it does not couch uncertainty in principled way if not through an extension for fuzzy reasoning [34]. The same issues are present in [32] and [7]. In the first case a hybrid approach is presented to fuse ontology-based context representation, and deductive and abductive reasoning for detection under uncertainty of abnormal objects from their characteristics and behaviour. In the latter work, PR-OWL is coupled with a Multi-Entity Bayesian Network for vessel of interest identification.

A much more powerful tool is FOL, which, with respect to propositional logic, is enough expressive to represent complex environments in a concise way. A difference relies in the *ontological commitment* [11], that is a concept that involves the reality and its representation by means of a model: propositional logic assumes that a set of mutually exclusive (true or false) facts hold or not in a certain world, linking symbols to values, while FOL considers that *objects and their relations* (predicates) do or do not hold, specifying much richer environment semantics. While being only semi-decidable, this problem is generally mitigated by attempting to convert the KB into Horn clausal form which is commonly used by inference engines [18].

Other possible decision theory tools include Dempster-Shafer theory (DS), from which the Dezert-Smarandache formalization has been derived [35], and fuzzy logic. Dempster-Shafer approaches, also known as evidence theory or theory of belief functions, allows to represent the evidence of different levels of abstraction, with the possibility of distinguishing between uncertainty and ignorance. With this respect, it is more flexible than classical Bayesian theory when dealing with incomplete knowledge and has explicit mechanisms for decision support. However, Dempster-Shafer theory cannot be used directly to encode expert knowledge in terms of complex sequences of events.

As it concerns fuzzy logic, in the maritime domain Balmat et alt. applied this technique to perform a risk assessment in a ship-centric system [36]. Fuzzy logic has been successfully used for event detection and recognition in the past [37], as it enables to take into account insufficient information, dealing thus with imprecise data, and the evolution of available knowledge.

Statistical Relational Learning (SRL) is an emerging research area that aims to represent, reason and learn in domains with complex relational and rich probabilistic structure [38]. SRL techniques have a very strong potential of application to SA systems for their ability to model dependencies between related instances (e.g. any combination of relations among observed entities or between a set of targets and the environment).

Markov Logic Networks is a recent SRL technique that attempts to unify the world of logic and probability [18]. MLNs are able to encode expressive domain knowledge through FOL formulas, and handle typically uncertain sensory data in a probabilistic framework that takes into account relations and dependencies through a graphical model (Markov Networks). MLNs have been applied to video surveillance systems for event detection in [39], where they are shown as a powerful fusion tool to combine observations coming from multiple heterogeneous sources of information.

Described in Section 2, MLNs will be here applied to the maritime domain where the dynamics and relations of many targets have to be captured through a multiplicity and heterogeneity of sensors and sources of information (including a priori knowledge, contextual information, and human-generated reports) in a vast and complex scenario. The overall goal is to provide a comprehensive situational picture to the operator. In our case, uncertainty is generated by the intrinsic ambiguity of subjective opinions of experts (soft data[40]), as no systematic and universal method exists to perfectly formalize the scenario: conflicts, approximations, imprecisions and contradictions can generate inexact, incomplete or unmeasurable information. Thus, attaching a weight to each formula in the knowledge base is a way to recognize the quality of its source and incorporate it into the reasoning process.

2. Markov Logic Networks

We here provide essential background notions of Markov Logic Networks, but the reader is advised to refer to [18] for further details. MLNs are a powerful tool for combining logical and probabilistic reasoning. While a knowledge base of logic formulas is satisfiable only by those worlds (truth values of atomic formulas) in which it is true, a MLN relaxes this hard constraint by associating a probability value to the worlds that do not fully satisfy the KB. Therefore, the fewer formulas a given world violates the more probable it is.

An MLN is then a set L of pairs (F_i, w_i) where F_i is a FOL formula and w_i its corresponding real-valued weight. The set of all formulas F_i in L constitutes the KB while the weight w_i associated to each F_i reflects how strongly the constraint imposed by the formula is to be respected. This impacts directly the probability assignment: worlds which satisfy a high weight formula are going to be much more probable than those that do not.

A Markov Logic Network L together with a finite set of constants C defines a Markov network $M_{L,C}$ that models the joint distribution of the set of random (binary) variables $X = (X_1, X_2, ..., X_n) \in \mathcal{X}$. Each variable of X is a a ground atom (predicate whose arguments contain no variables) and $\mathcal{X}$ is the set of all possible *worlds*, that is the set of all possible truth value assignments of n binary variables. The network is built as follows:

- $M_{L,C}$ contains one (binary) node for each possible ground atom given L and C
- An edge between two nodes indicates that the corresponding ground atoms appear together in at least one grounding of one formula in L. Ground atoms belonging to the same formula are connected to each other thus forming cliques.
- A feature f_i is associated for each possible grounding of a formula F_i in L. Each f_i assumes value 1 if the corresponding ground formula is true and 0 otherwise.

The probability distribution over X taking values $x \in \mathcal{X}$ specified by $M_{L,C}$ is given by:

$$P(X = x) = \frac{1}{Z} \exp\left(\sum_{i=1}^{|L|} w_i n_i(x)\right) \tag{1}$$

where $|L|$ indicates the cardinality of L, thus counting the number of formulas of the knowledge base, and $n_i(x)$ is the number of true groundings of F_i in the world x.

$$Z = \sum_{x' \in \mathcal{X}} \exp\left(\sum_{i=1}^{|L|} w_i n_i(x')\right) \tag{2}$$

is a normalizing factor often call *partition function*.

Given the joint distribution function in (1), it is possible to calculate the probability that a given formula F_i holds given the Markov Network $M_{L,C}$ as follows:

$$\begin{aligned} P(F_i|M_{L,C}) &= \sum_{x \in \mathcal{X}_{F_i}} P(X = x|M_{L,C}) \\ &= \frac{1}{Z} \exp\left(\sum_{x \in \mathcal{X}_{F_i}} w_i n_i(x)\right) \end{aligned} \tag{3}$$

where $\mathcal{X}_{F_i}$ is the set of worlds where F_i holds.

While (1) provides the probability of configuration x of truth values for the ground atoms in the Markov Network, (3) can be used instead to evaluate the probability that a formula F_i (e.g. a predicate representing an event) holds given $M_{L,C}$ where C is composed by observed entities and other constants provided by contextual knowledge. This gives a glimpse of the power of the framework, an arbitrary formula that can be grounded in $M_{L,C}$ can be queried to get the probability of being true. Thus not only the formulas in L but also any logical combination of them that can be grounded in the Markov network can be queried as well. This is extremely important for a SA system where the operator might want to evaluate the truth degree of a new (complex) event or condition as the combination of existing evidence in the KB.The framework can also provide the probability that a formula F_2 holds given that formula F_1 does or provide an answer to weather the KB entails a given formula.

According to the definitions given in Section 1.1, an MLN provides an explicit way of encoding knowledge. However, both rule weights and the rules themselves can be be learned from data. These capabilities make MLNs a powerful tool that combines the benefits of both implicit and explicit modelling.

One point that should be highlighted is that the Markov network of ground atoms is comprised, as already mentioned, by a set of binary random variables each of which constitutes a node in the graph. In our application, the grounding of a predicate can happen due to a priori or contextual knowledge (e.g. $harbour(H_1)$ where H_1 is a known location such as "La Spezia") or accrued sensory evidence (e.g. cargo(V_1) where V_1 corresponds to an observed vessel). An example of Markov Network is shown in Figure 4. The truth value of grounded predicates can be provided by external knowledge and observations or inferred by the reasoning engine. For example we might know that H_1 is a harbour while H_2 is not. This would allow to say that harbour(H_1) is true and that harbour(H_2) is false thus providing binary input to the system. However, one would like to add an additional level of uncertainty, that is observation uncertainty, to encode the naturally imprecise or uncertain nature of data coming from sensors and human observers. For example, the ground predicate $proximity(V_1, V_2)$ asserts the vicinity of vessels V_1 and V_2 according to a certain threshold. The measured distance could be affected by error leaving thus a margin of uncertainty on the truth value of the predicate. To encode such observation uncertainty, a possible solution, proposed in [39], involves adding a single atom rule to the rule base with associated weight proportional to the detection probability of the observed evidence. Observation uncertainty will not be employed in the experiments of Section 4 and is instead left to further investigation and future work.

3. Completion of complex events

As described in Section 1.1, a complex event is the logical combination of two or more events and therefore cannot be directly observed but has to be inferred by observing the occurrence of component events. As suggested in Section 1.1, it is often the case for SA systems that anomalous or critical situations have to be deduced from a number of indicators derived from apparently normal behaviour of observed entities. Indicators take generally the form of simple events but this is not a rule and a complex event can be composed by an arbitrary combination of simple and other complex events.

In order to prepare a timely response, adopt adequate counter-measures, or simply plan in advance future actions, it would be of course useful to detect a critical condition when it is about to happen. It would be interesting then to detect complex events that are "almost completed" or, in our logic setting, "almost true". In other words, it would useful to provide to the operator a continuously updated indication of how complex events are building up. Take for example the rule

$$cargo(v) \wedge isHeadingTo(v,h) \wedge harbour(h) \wedge risk(h, \text{“high”}) \\ \Rightarrow alarm(v)$$

where the unary predicate *alarm* marks the occurrence of a complex (critical) event involving vessel v. Since the antecedent is composed by the conjunction of four predicates (events), it evaluates to *true* only when all four predicates do. It would be interesting to know how much the current world is satisfying the implication. For example, if the first, third and fourth predicates of the antecedent were true then we would say that the rule is at 75% of its completion and could poentially trigger if the remaining predicate evaluated also true in the future. A condition like the one exemplified could attract the attention of the operator or of the system itself that might decide to conduct further investigation on vessel v directing sensing capabilities or information providers to acquire additional data (see JDL level 4 [12]).

The formulation of a priori (e.g. contextual) knowledge and observed evidence within the MLN framework allows the evaluation of complex events by querying the truth value of the associated (consequent) predicate. In the example above, the complex event encoding a critical condition is represented by the *alarm* predicate on the right side of the implication. However, the probability of such predicates does not necessarily reflect their completion condition. While it could work nearly as expected for double implications ($\Leftrightarrow$), in the case of implications, as in the example above, the antecedent is only a sufficient condition for the consequent to be true but is not a necessary one. Therefore, when the antecedent is false, is says nothing about the truth value of the consequent. In addition, the consequent could be involved in other formulas and be subject to their effects.

Evaluating the probability of the antecedent being true does not provide a solution either as said probability does not match the completion concept. If for example the weight the formula expresses a hard constraint, then regular FOL holds and one false predicate brings the truth value of a conjunction to zero (as in the example above).

Therefore, to evaluate the level of completion of complex events we use the following procedure for each formula F in the KB representing a complex event:

1. Convert F into Conjunctive Normal Form[1] (CNF) where grounding of variables is done according to observations and contextual knowledge;

[1] conjunction of disjunctions of literals

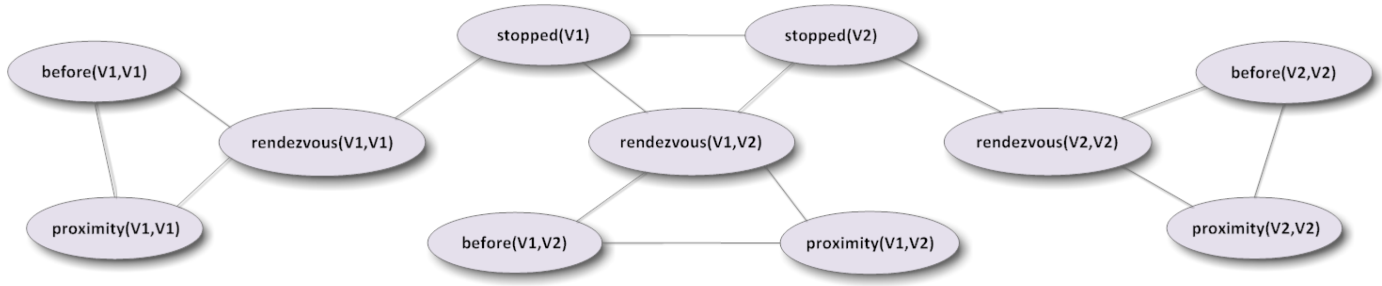


Figure 4: Example of Markov network. The network is obtained by grounding formulas #18 and #19 in Table 3 with constants $V1$ and $V2$ referring to two observed vessels.

2. Evaluate the truth value of ground predicates according to available evidence. In the case of atoms whose truth value is unknown, the Closed World Assumption[2] (CWA) holds;
3. Let E_i be FOL literals representing simple events, then recursively compute the completion of F by using the following two rules:
   - if $(E_1 \wedge \cdots \wedge E_K)$ is a conjunction of K events (predicates) then $completion(E_1 \wedge \cdots \wedge E_K) = \sum_{k=1}^{K} (P(E_k))$
   - if $(E_1 \vee \cdots \vee E_N)$ is a disjunction of N events then $completion(E_1 \vee \cdots \vee E_N) = \max_n (P(E_n))$

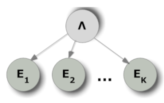
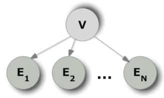
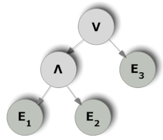
In other words, the completion of a conjunction of events is given by the average of their probability. If the events are of complex type they must be recursively evaluated by applying the same rules. The probability of each atom is obtained querying the Markov Network as per (3). The probability of literals whose truth value is unknown is set to zero, the same is done for not yet grounded predicates. Please note that each atom could be either a positive or negative literal and that the probability being evaluated for each atom is the probability of it being true.

The completion of a disjunction of events is instead obtained by taking the maximum probability value among the events in the disjunction. Completion calculation is further exemplified in Table 2, where the first two rows show how again the calculation for conjunctions and disjunctions and the third row exemplifies the case of complex event whose event tree is shown at the bottom left.

The application of the CWA sets to zero the probability of any predicate that has no supporting evidence for being true. This means that before invoking the CWA both external evidence (i.e. actual observations) and (possible) inference effects should be checked for ground atoms with unknown truth value every time their completion level is to be evaluated.

Since the completion of events is calculated only after the evaluation of the probability values of the predicates through the MLNs framework, it does not affect inference and it is used only as an indicator for the operator or the system. The choice of the CWA provides a prudent look by setting to zero the probability of unobserved events being true. The open world case could be practically dealt with in several ways. One example

Table 2: Completion of complex events. Being E_i FOL literals representing simple events, the first two rows show the how the completion is calculated in the case of conjunction ($\wedge$) and disjunction ($\vee$) of component simple events respectively. The third row (bottom) shows an example of a complex event whose event tree comprises a combination of both logical operators.

Event tree	Logic formula	Completion
	$(E_1 \wedge \cdots \wedge E_K)$	$\frac{1}{K} \sum_{k=1}^{K} (P(E_k))$
	$(E_1 \vee \cdots \vee E_N)$	$\max_n (P(E_n))$ $n = \{1, \ldots, N\}$
	$(E_1 \wedge E_2) \vee E_3$	$\max\left(\frac{P(E_1)+P(E_2)}{2}, P(E_3)\right)$

could be to set the probability of unknown predicates to 0.5. A better option could be to compute the completion as in the CWA case (probability of unknown literals set to zero) but to explicitly highlight to the operator the number of false and unknown events explicitly.

4. Knowledge representation and experiments

The creation of a knowledge base implies the use of a representation formalism to code Subject Matter Experts (SME) knowledge into formulas. The domain, that is a part of the world about which we want to express sentences, is represented by a set of assertions, also said formulas, in FOL, which guarantees a precise semantic characterization. In our experiments we set manually the MLN weights, which represent the uncertainty of each rule, as sample data was not available to unsupervisedly

[2] according to the Closed World Assumption, atoms with unknown truth value are considered false [18].

learn the anomaly model. Weights and rule learning from data will be investigated in future work. When a large amount of data is available, it is preferable to train the MLN as described in Section 2.

We start defining a knowledge base that will model our domain by describing entities and their relationship. We use a maximum weight ω that is proportional to the number of the ground atoms [18] to define the highest certainty about a rule, that is a hard constraint; fractions of ω can be assigned to less confident formulas [39].

Contextual information is provided by a human operator, and observed evidence necessary to ground the MLN is simulated as already processed sensory data. It is important to distinguish the role of these two sources of information. A static entity and the associated resources or characteristics can be described a priori by a human operator; this knowledge can be updated in time when some of these features vary, but the entity is (almost) permanent in the domain. On the contrary, evidence about moving or non-static objects is created on-the-fly when needed, it is not permanent and it can vary in time. For this reason, we must distinguish between *contextual evidence*, that is contextual (static) information, and *observed evidence* that refers to sensory data regarding a specific vessel of interest in a certain instant of time.

Two main scenarios will be examined, discussing the impact of contextual information injection. In the first case, a possible rendezvous (stow 'n' go) between ships is depicted; maritime experts have been consulted to highlight common patterns and to create the knowledge base after the situation definition. In the second scenario, extended from [17], the threat is represented by a dangerous combination of material carried by cargo ships that arrive at adjacent berths simultaneously.

To run experiments with the Markov Logic Networks we used Alchemy[3], a library for statistical relational learning and probabilistic logic inference based on the Markov logic representation, integrated in a graphical interface provided by probCog[4], which allows to easily code rules and set Alchemy's parameters.

Alchemy offers the possibility to learn the weights associated to the KB formulas from a set of sample patterns, in the form of true/false atoms. In our experiments we set manually the MLN weights (Table 12 and Table 3), which represent the uncertainty of each rule, as sample data was not available to unsupervisedly learn the anomaly model. Weights and rule learning from data will be investigated in future work.

MLN inference hinges on one side on Markov Network inference (#P-complete) and on FOL inference (NP-complete) on the other. However, properly exploiting the structure of the network, MLN inference is in some cases more efficient than in standard FOL [18]. In our experiments, we run approximate inference using the MonteCarlo-SAT (MC-SAT) algorithm with 5000 maximum steps.

[3]http://alchemy.cs.washington.edu/

[4]http://wwwbeetz.informatik.tu-muenchen.de/probcog-wiki/index.php

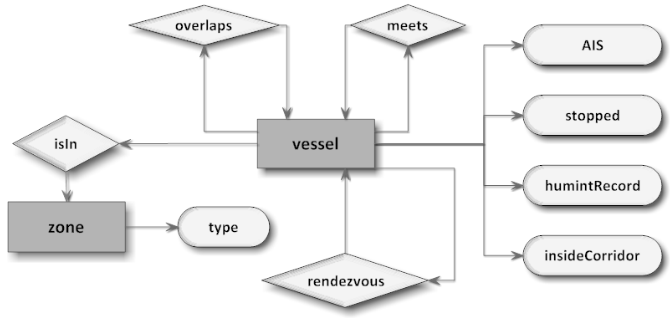


Figure 5: Entities and relations of the proposed "rendezvous" example.

4.1. Rendezvous scenario

The first scenario aims to detect a *rendezvous* act, that is a meeting of two vessels for trafficking or smuggling of people or goods (drugs, food, oil, etc.). The two vessels usually have no transponder system activated, to be undetected, and commonly they meet offshore, far away from the coast. A less frequent case is when a small boat would rendezvous with the smugglers' mother ships, usually cargo or large vessels.

The *rendezvous* represents a complex event, and is a binary relation as it involves two objects of the same type (vessel). The complete diagram of entities involved in this scenario is shown in Figure 5. The focus is a vessels-centric structure, with the main entity having attributes as *stopped*, *insideCorridor* and *AIS* to indicate that the ship has stopped, navigates inside a virtual traffic corridor (allowed area), and has AIS transponder on. The *stopped* predicate implies a low-level action where the speed over ground (SOG) or the position of the ship is monitored, as well as *AIS* requires a translation of a low-level signal into an event. The *insideCorridor* predicate requires a prior definition of navigable zones and normalcy traffic corridors (as in Figure 3), but their modelization is out of the scope of this work (see for example [23]).

To describe complex events, which often imply a temporal sequence of facts, we employ Allen's temporal logic [41]. Allen's Interval Algebra provides a composition table for reasoning about the relations that occur between temporal intervals. Other predicates as *overlaps* and *meets* are unary relations and indicate a temporal link between two vessels.

Another attribute represents HUMan INTelligence (HUMINT) reports, which consist in additional information about suspicious vessels. The other entity in the scenario is the *zone* where the vessel is located, and it can take values $harbour$, $nearCoast$, $openSea$ or $intWaters$.

4.1.1. Knowledge base

The knowledge base, whose rules are defined by a human operator, describes the rendezvous procedure and how it can be discovered and represented, and the anomalies that can be derived from data. A subject matter expert can help hand coding the knowledge rule that are accepted for each domain, or the models can be supervisedly learned from data.

The formulas set consists of nineteen rules, that describe normal and abnormal behaviours for ships. The first rules (#1-#4)

Table 3: Knowledge base for the rendezvous scenario in FOL with associate weights

#	Rule	Weight
1	$overlaps(v, y) \Leftrightarrow overlaps(y, v)$	ω
2	$meets(v, y) \Leftrightarrow meets(y, v)$	ω
3	$proximity(v, y) \Leftrightarrow proximity(y, v)$	ω
4	$rendezvous(v, y) \Leftrightarrow rendezvous(y, v)$	ω
5	$stopped(v) \wedge (isIn(v, openSea) \vee isIn(v, intWaters)) \Rightarrow suspicious(v)$	$4/5\ \omega$
6	$stopped(v) \wedge (isIn(v, harbour) \vee isIn(v, nearCost)) \Rightarrow \neg suspicious(v)$	$2/5\ \omega$
7	$\neg AIS(v) \Rightarrow alarm(v)$	ω
8	$\neg insideCorridor(v) \Rightarrow suspicious(v)$	$4/5\ \omega$
9	$humint(v, smuggling) \Rightarrow suspicious(v)$	$3/5\ \omega$
10	$humint(v, clear) \Rightarrow \neg suspicious(v)$	$1/5\ \omega$
11	$suspicious(v) \Rightarrow alarm(v)$	ω
12	$\neg suspicious(v) \Rightarrow \neg alarm(v)$	$1/5\ \omega$
13	$isIn(v, z) \Rightarrow (z \neq zp) \wedge \neg isIn(v, zp)$	ω
14	$isIn(v, z) \wedge isIn(y, zp) \wedge (z \neq zp) \Rightarrow \neg proximity(v, y)$	ω
15	$\neg proximity(v, y) \Rightarrow \neg rendezvous(v, y)$	ω
16	$suspicious(v) \wedge suspicious(y) \wedge (overlaps(v, y) \vee meets(v, y)) \wedge proximity(v, y) \Rightarrow rendezvous(v, y)$	ω
17	$(overlaps(v, y) \vee meets(v, y)) \wedge proximity(v, y) \Rightarrow rendezvous(v, y)$	$1/5\ \omega$
18	$\neg stopped(v) \vee \neg stopped(y) \Rightarrow \neg rendezvous(v, y)$	$3/5\ \omega$
19	$before(v, y) \wedge proximity(v, y) \Rightarrow \neg rendezvous(v, y)$	$4/5\ \omega$

describe the symmetry of some relationship as temporal ones, proximity and rendezvous. Then we build the anomaly identifiers: a vessel is defined suspicious if stopped in international waters or open sea (rule #5), if it sails outside traffic corridors (#8), or if there is a HUMINT report on it (#9), while the permanence in a harbour or near the coast is not considered an anomaly (#6). If the vessel has the AIS transceiver system turned off, the alarm flag is raised (#7) independently from where the ship is located or what is doing. A suspicious vessel triggers an alarm (#11 and #12).

The concept of proximity is shaped by rule #13, #14 and #15: a vessel can be located in one zone per time (#13), two vessels that are not located in the same area can not be close in space (#14), and in this last case there can be no rendezvous (#15). On the contrary, two vessels that are in the same area in overlapping time intervals define a rendezvous anomaly (#17), rule that is stronger if they are flagged as suspicious (#16). If one of the two has not stopped, the rendezvous is not possible (#18), like when the two vessels are in the same area in not overlapping time intervals (#19).

4.1.2. *Contextual information*

Generally, context does not directly provide information on the object of interest, but in this case the evidence assumes meaning only when matched with contextual information. For instance, the *isIn* predicate fuses observed evidence (the vessel position) with a priori knowledge (the zones).

Table 4: Contextual information for the rendezvous scenario.

$isIn(V1, openSea)$	$\neg insideCorridor(V1)$
$isIn(V2, openSea)$	$\neg insideCorridor(V2)$
$isIn(V3, nearCoast)$	$insideCorridor(V3)$
$isIn(V4, intWaters)$	$insideCorridor(V4)$
$isIn(V5, intWaters)$	$insideCorridor(V5)$

Thus in our case, the zone subdivision and the traffic corridor are additional information that can be exploited to better refine the complex events. We know that rendezvous are mostly happening in open sea. Without zones definition, thus no context, we would not be able to formalize this rule.

Moreover, context is used here to strengthen the posterior probability of a predicate; for instance, *suspicious* can be grounded either if the "vessel stopped *at zone X*", or HUMINT provided a *smuggling* report, or vessel is not transiting into *traffic corridors*. Two of these situations are generated only when sensory data is matched with context, and context actively contributes to reinforce the predicate. Contextual information is presented in Table 4.

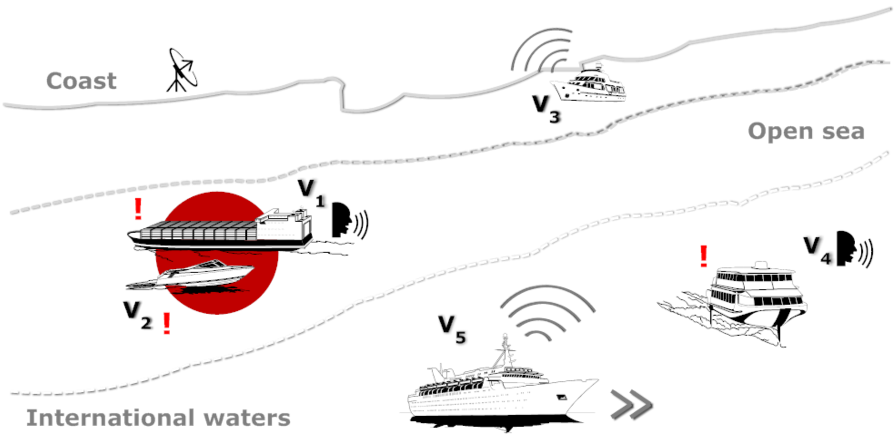


Figure 6: Illustration of the rendezvous example: several vessels navigate at different distance from the coast. Some of them have the AIS transponder off (symbolized by the red "!"), and for two of them intelligence reports have been provided (represented by the speaking head).

Table 5: Observed evidence in the rendezvous scenario.

$stopped(V1)$	$humint(V1, smuggling)$
$stopped(V2)$	$humint(V4, smuggling)$
$stopped(V3)$	$proximity(V1, V2)$
$stopped(V4)$	$\neg proximity(V1, V3)$
$\neg stopped(V5)$	$\neg proximity(V1, V4)$
$overlaps(V1, V2)$	$proximity(V1, V5)$
$overlaps(V2, V3)$	$\neg proximity(V2, V3)$
$overlaps(V3, V4)$	$\neg proximity(V2, V4)$
$overlaps(V4, V5)$	$\neg proximity(V2, V5)$
$\neg AIS(V1)$	$\neg proximity(V3, V4)$
$\neg AIS(V2)$	$\neg proximity(V3, V5)$
$AIS(V3)$	$proximity(V4, V5)$
$\neg AIS(V4)$	
$AIS(V5)$	

Table 6: Anomalies in the rendezvous scenario; the rendezvous anomalous event is taking place only between V_1 and V_2.

	V_1	V_2	V_3	V_4	V_5
V_1		Y	N	N	N
V_2	Y		N	N	N
V_3	N	N		N	N
V_4	N	N	N		N
V_5	N	N	N	N	

Table 7: Results for the rendezvous alarm in the first scenario.

	V_1	V_2	V_3	V_4	V_5
V_1		1.00	0	0	0
V_2	1.00		0	0	0
V_3	0	0		0.02	0
V_4	0	0	0.02		0.01
V_5	0	0	0	0.01	

Table 8: Results for the rendezvous alarm in the first scenario with no contextual information provided.

	V_1	V_2	V_3	V_4	V_5
V_1		0.96	0	0	0
V_2	0.96		0	0	0
V_3	0	0		0	0
V_4	0	0	0		0.2
V_5	0	0	0	0.2	

4.1.3. Results

As concrete example, we imagine a situation in which V_1 and V_2 are having a rendezvous at open sea; they have the AIS transponder switched off and are close in space ($proximity(V_1, V_2)$). On V_1, intelligence sources provided a *smuggling* report, indicating that historical data suggests the vessel may be suspicious. In the meantime, a cruise ship V_5 is transiting in international waters, a big leisure craft V_3 is moored near the coast. Another vessel V_4, for which smuggling reports have been provided as well, is still in international waters with the AIS transponder not activated. The scenario is illustrated in Figure 6. From sensory data (Table 5), we observe that V_1, V_2, V_3 and V_4 are still, thus the predicate *stopped* is true. Three of them do not have AIS transponder sending messages, and (V_1, V_2) and (V_4, V_5) are close to each other.

Table 9: Low-level anomalies for a single vessels, as, for instance, AIS transceiver switched off can raise an alarm flag.

	V_1	V_2	V_3	V_4	V_5
Alarm	Y	Y	N	Y	N

The system is asked to answer two queries, $P(alarm(V_n)|M_{L,C})$ and $P(rendezvous(V_n, V_m)|M_{L,C})$, which represent the probability for the predicates *alarm* to be true for a given vessel V_n, and the probability that a *rendezvous* is happening between vessels V_n and V_m. $M_{L,C}$ represents

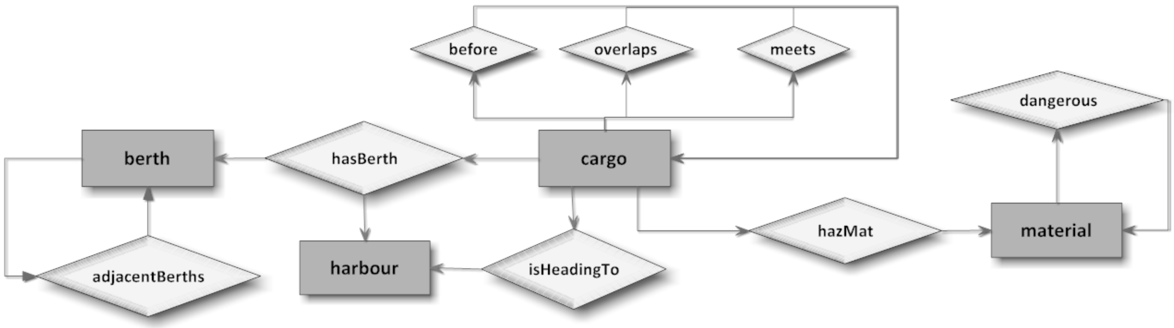


Figure 7: Entities and relations of the proposed “hazmat” maritime example.

Table 10: Results for the single vessel alarm in the rendezvous scenario.

	V_1	V_2	V_3	V_4	V_5
Alarm	0.99	0.98	0.08	1.00	0.01

Table 11: Results for the single vessel alarm in the rendezvous scenario with no contextual information provided.

	V_1	V_2	V_3	V_4	V_5
Alarm	1.00	0.97	0.56	0.99	0.09

the Markov Network created groundings the set formulas L as per Table 3, while C is the set of constants as in Section 4.1.1. Contextual and sensory evidences are specified in Tables 4 and 5 respectively. We tested the MLN with the same evidence set, but two different contexts: in the first case, all the available information is provided, while in the second case the contextual information is removed and only the sensory data is maintained.

Ground truth values are shown in Table 6 and 9 for $P(rendezvous(V_n, V_m)|M_{L,C})$ and $P(alarm(V_n)|M_{L,C})$ respectively.

The results presented in Table 10 indicate that, in presence of contextual information, the system correctly classifies the anomaly of having the AIS turned off or being suspicious enough to alert the operator, and, as suggested by Table 7, it also recognizes the rendezvous between V_1 and V_2 with a high probability assigned. No other rendezvous is detected. When contextual information is missing, the AIS anomaly, which derives from sensory data, is detected with high confidence (Table 11). Contrarily, the rendezvous alert, that depends on traffic corridors and sea zones defined by context (Table 4), is classified as an abnormal pattern with lower confidence (Table 8).

Uncertainty rises for (V_4, V_5) and two single (V_3 and V_5) unsuspicious vessels when no high-level information is provided. This can be noticed in the detected anomalies for ships couples (Table 7), where the missing context suggests the possibility of a open sea rendezvous, and in the single-vessel alarm values (Table 10), which are no longer zero (or close to zero). This is due to the fact that the reasoner, which entirely relies on sensory data, assigns higher importance to the *proximity* predicate. As we can see, context here does not have significant discriminative influence on results and rendezvous detection; however, it helps by reducing uncertainty and refining the event detection probability.

4.2. Hazmat scenario

In this scenario several cargo ships head toward a harbour. Some of the ships carry chemical or generic hazardous materials (hazmat), as, for instance, bleach and ammonia, that when combined may cause a severe threat [17]. The ships are assigned berths in a row, and will be in the harbour before others or at the same time.

The entities in our examples are, as shown in Figure 7, *cargo*, *harbour*, *material* and *berth*, which are linked together by the fact that the cargo ship, carrying some hazardous material (*hazMat*, which can be *dangerous* if combined with other sensitive material) is heading (*isHeadingTo*) toward a certain harbour, in which has a berth. The predicate *hasBerth* takes a triplet of harbour, vessel and berth as argument to bound the three classes. The berth has a predicate *adjBerth*, which is important to indicate that two vessels are moored in adjacent berths, and thus are *neighbours*.

Instead of the seven original predicates ($before(v_1, v_2)$, $meets(v_1, v_2)$, $overlaps(v_1, v_2)$, $starts(v_1, v_2)$, $during(v_1, v_2)$, $finishes(v_1, v_2)$ and $isEqualTo(v_1, v_2)$), which define time of permanence at berths of the two ships v_1 and v_2, we shorten the list to $before(v_1, v_2)$, $meets(v_1, v_2)$ and $overlaps(v_1, v_2)$, as these are the most frequent time relations between ships permanence times. In fact, a ship can leave a harbour *before* another comes in, thus the two vessels do not meet. Alternatively, it can stay moored for a long time, which *overlaps* with other vessels permanence. One more case is represented by the *meeting* event, that happens if a vessel leaves just after another one arrives; this situation is relevant as the cargo content may not be fully processed, and still placed on the berth, thus allowing interactions with other ships contents. Other temporal definitions in our domain can be considered special cases of the *overlap* relation. These predicates, that are unary relations, are important as they allow us to properly model the scenario time line and the causality between successive events.

Table 12: Knowledge base for the hazmat scenario in FOL with associated weights

#	Rule	Weight
1	$overlaps(v, y) \Leftrightarrow overlaps(y, v)$	ω
2	$meets(v, y) \Leftrightarrow meets(y, v)$	ω
3	$neighbours(v, y) \Leftrightarrow neighbours(y, v)$	ω
4	$concurrent(v, y) \Leftrightarrow concurrent(y, v)$	ω
5	$dangerous(m1, m2) \Leftrightarrow dangerous(m2, m1)$	ω
6	$alarm(v, y) \Leftrightarrow alarm(y, v)$	ω
7	$meets(v, y) \vee overlaps(v, y) \Leftrightarrow concurrent(v, y)$	ω
8	$\neg meets(v, y) \wedge \neg overlaps(v, y) \Leftrightarrow \neg concurrent(v, y)$	$4/5\ \omega$
9	$before(v, y) \Rightarrow \neg concurrent(v, y)$	ω
10	$\neg concurrent(v, y) \Rightarrow \neg alarm(v, y)$	ω
11	$cargo(v) \wedge isHeadingTo(v, h) \wedge harbour(h) \Leftrightarrow hasBerth(v, x, h) \wedge berth(x)$	ω
12	$cargo(v) \wedge cargo(y) \wedge hasBerth(v, x, h) \wedge hasBerth(y, z, h) \wedge adjBerth(x, z) \Leftrightarrow neighbours(v, y)$	ω
13	$\neg neighbours(v, y) \Rightarrow \neg alarm(v, y)$	$4/5\ \omega$
14	$cargo(v) \wedge cargo(y) \wedge hazMat(v, m1) \wedge hazMat(y, m2) \wedge \neg dangerous(m1, m2) \Rightarrow \neg alarm(v, y)$	$3/5\ \omega$
15	$cargo(v) \wedge cargo(y) \wedge hazMat(v, m1) \wedge hazMat(y, m2) \wedge neighbours(v, y) \wedge dangerous(m1, m2) \wedge concurrent(v, y) \Rightarrow alarm(v, y)$	ω

4.2.1. Knowledge base

The domain knowledge can be formalized with FOL formulas, described in Table 12, where the higher the weight the more confident the statement.

The first six rules (#1-#6) codify the symmetry among elements, and are useful to avoid sorting items; in this way relations between ships V_X and V_Y hold also viceversa.

Rule #7 states that two vessels which *meet* or *overlap* are *concurrent* in time, simplifying the concept of "simultaneous" or "operative/moored at the same time". The opposite condition (#8) or the case when one vessel arrives or leaves *before* (#9) others define having no interaction with other ships in the scenario (being not concurrent). If two vessels are not concurrent, they do not represent a threat (#10).

Referring to spacial relationship, a cargo that is heading toward a harbour will have a berth assigned (#11), and two vessels in the same harbour will be *neighbours* only if they will share adjacent berths (#12). If two vessels are not neighbours, they can not generate an alarm (#13), as well as if they transport cargo materials that are not dangerous when combined (#14).

We can then define the main threat by the rule for which two neighbour cargo ships carry hazmats which are potentially dangerous if combined (#15). In this case the cargo ships share adjacent berths and are moored in the harbour at the same time.

4.2.2. Contextual information

Probabilistic knowledge must be integrated with explicit contextual knowledge, as sensory data may be not enough to represent and identify complex situations. A simple low-level anomaly detector would not detect the aforementioned threat, as two cargo ships which enter in a harbour, even carrying hazmats, for commercial reasons raise no alarm. However, additional information provided by context can help to identify a suspicious event.

Context, described in detail in Table 13, can be represented in our scenario by scenario-dependent information, which is:

- A harbour H_1 has four berths $B_1, \ldots, B_4$, and some of the berths are adjacent. The exact map of adjacent berths can be provided by a human operator. In our case, we codify the proximity with a set of symmetric rules. We suppose, as shown in figure 8, that berths B_1 and B_2 are adjacent, as well as B_3 and B_4, and B_4 and B_5.
- Some materials defined by M, if combined together, are dangerous or potentially lethal. This information must necessary be provided by a SME, as it can not directly be inferred from materials only. In our example, we suppose that (M_1, M_2), (M_2, M_3) and (M_2, M_4) are dangerous combinations.

As we will see in the experiments, it is important that this information be the most complete as possible, to depict accurately the scenario with its entities and relationships.

4.2.3. Results

We aim to demonstrate how contextual information is a crucial key element to build an exhaustive and accurate situational picture, which allows to timely detect an anomaly.

We imagine a situation as the one described in Figure 8 and Table 14. V_1 leaves the harbour prior to the arrival of V_2 and V_4. After a while, V_3 reaches berth B_3, and it remains there when V_5 arrives and moors at B_2. The fact that a ship is carrying hazardous material and the type of material can be classified

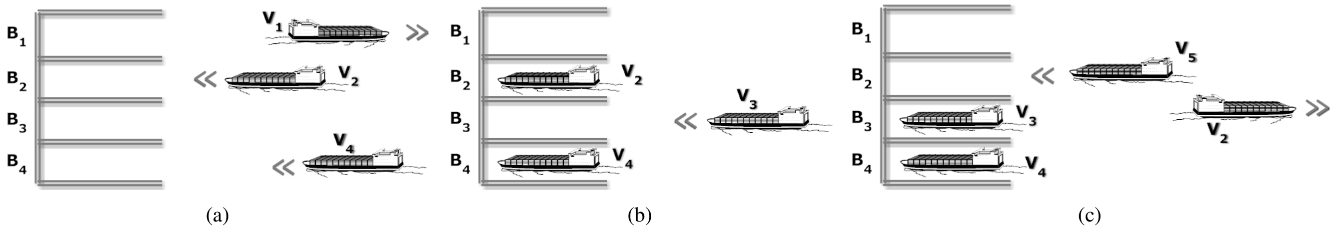


Figure 8: Illustration of the evolution of the hazmat scenario. Cargo ship V_1 leaves much earlier than the arrival of V_2 and V_4 (a), and before V_3 reaches its berth (b). As V_2 leaves, V_5 arrives (c).

Table 13: Contextual information provided a priori for the "hazmat" scenario. Apart from harbour and its facilities, the description of dangerous combinations of materials is provided.

$harbour(H_1)$
$berth(B_1, H_1)$
$berth(B_2, H_1)$
$berth(B_3, H_1)$
$berth(B_4, H_1)$
$adjBerth(B_1, B_2)$
$adjBerth(B_2, B_3)$
$adjBerth(B_3, B_4)$
$\neg adjBerth(B_1, B_3)$
$\neg adjBerth(B_1, B_4)$
$\neg adjBerth(B_2, B_4)$
$dangerous(M_1, M_2)$
$dangerous(M_2, M_3)$
$dangerous(M_2, M_4)$
$\neg dangerous(M_1, M_4)$
$\neg dangerous(M_3, M_4)$

Table 14: Observed facts (evidence) in the "hazmat" scenario. The time of permanence of a cargo at its berth is calculated only with respect to neighbour cargo ships.

$cargo(V_1)$	$hasBerth(V_1, B_1, H_1)$
$cargo(V_2)$	$hasBerth(V_2, B_2, H_1)$
$cargo(V_3)$	$hasBerth(V_3, B_3, H_1)$
$cargo(V_4)$	$hasBerth(V_4, B_4, H_1)$
$cargo(V_5)$	$hasBerth(V_5, B_2, H_1)$
$hazMat(V_1, M_1)$	$before(V_1, V_2)$
$hazMat(V_2, M_2)$	$before(V_1, V_5)$
$hazMat(V_3, M_3)$	$overlaps(V_2, V_3)$
$hazMat(V_4, M_4)$	$overlaps(V_2, V_4)$
$hazMat(V_5, M_2)$	$overlaps(V_4, V_3)$
$isHeadingTo(V_1, H_1)$	$overlaps(V_3, V_5)$
$isHeadingTo(V_2, H_1)$	$overlaps(V_4, V_5)$
$isHeadingTo(V_3, H_1)$	
$isHeadingTo(V_4, H_1)$	
$isHeadingTo(V_5, H_1)$	

Table 15: Dangerous combinations of hazardous materials carried by cargo ships that share adjacent berths are highlighted in red and marked with "Y".

	V_1	V_2	V_3	V_4	V_5
V_1		N	N	N	N
V_2	N		Y	N	N
V_3	N	Y		N	Y
V_4	N	N	N		N
V_5	N	N	Y	N	

as sensory data, as this information can be fetched on-the-fly when the ship becomes a vessel-of-interest or when the system registers the vessel. Also the time predicates can be calculated at runtime, comparing the ETA (Estimated Time of Arrival) and a minimum time of permanence to handle the ship content.

All the cargo ships in our scenario transport hazardous material, but from a priori information (Table 13) we know that the dangerous combinations are constituted by (M_1, M_2), (M_2, M_3) and (M_2, M_4).

The query $P(alarm(V_n, V_m)|M_{L,C})$ represents the probability for predicate *alarm* to be true for a given vessel couple (V_n, V_m), where $M_{L,C}$ is the Markov Network created by grounding the set formulas L shown in Table 12, C is the set of constants as defined in Section 4.2.1, and contextual and sensory evidences are provided according to Tables 13 and 14 respectively. For sake of clarity and completeness here we consider all the possible combinations of vessels (V_n, V_m); however, to speed up the application of the proposed techniques in a real-world environment, only vessels that will share adjacent berths could be selected for a check.

In Table 15 are shown the possible risky combinations of hazardous materials carried by cargo ships that share adjacent berths and are moored in the harbour at the same time. Threats are highlighted in red and marked with "Y", while a normal situation is white coloured, marked with "N" and should raise no alarm flag. Diagonal terms give no anomaly. Hazardous material $M1$ is considered dangerous when combined with others, but as the cargo which carries it leaves $before$ others, no alarm is raised. Materials that are brought at not adjacent berths do not constitute a dangerous combination, thus the couple (V_4, V_5) does not constitute a threat.

In Table 16 and 17 the results for this scenario are presented.

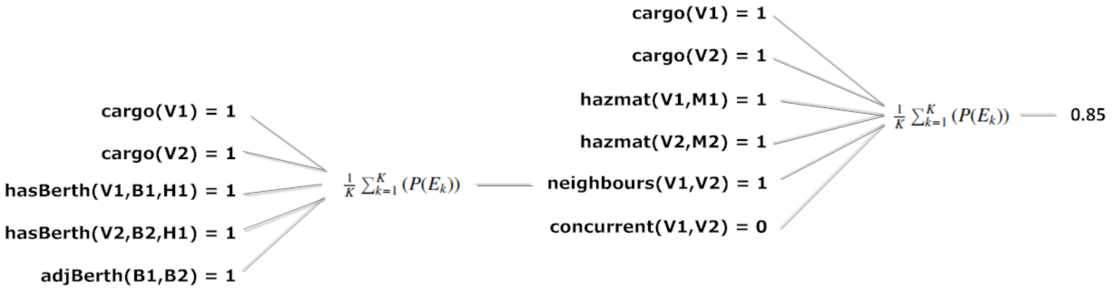


Figure 9: Example of level of completion calculation for the *alarm* complex event in formula #15 in Table 12.

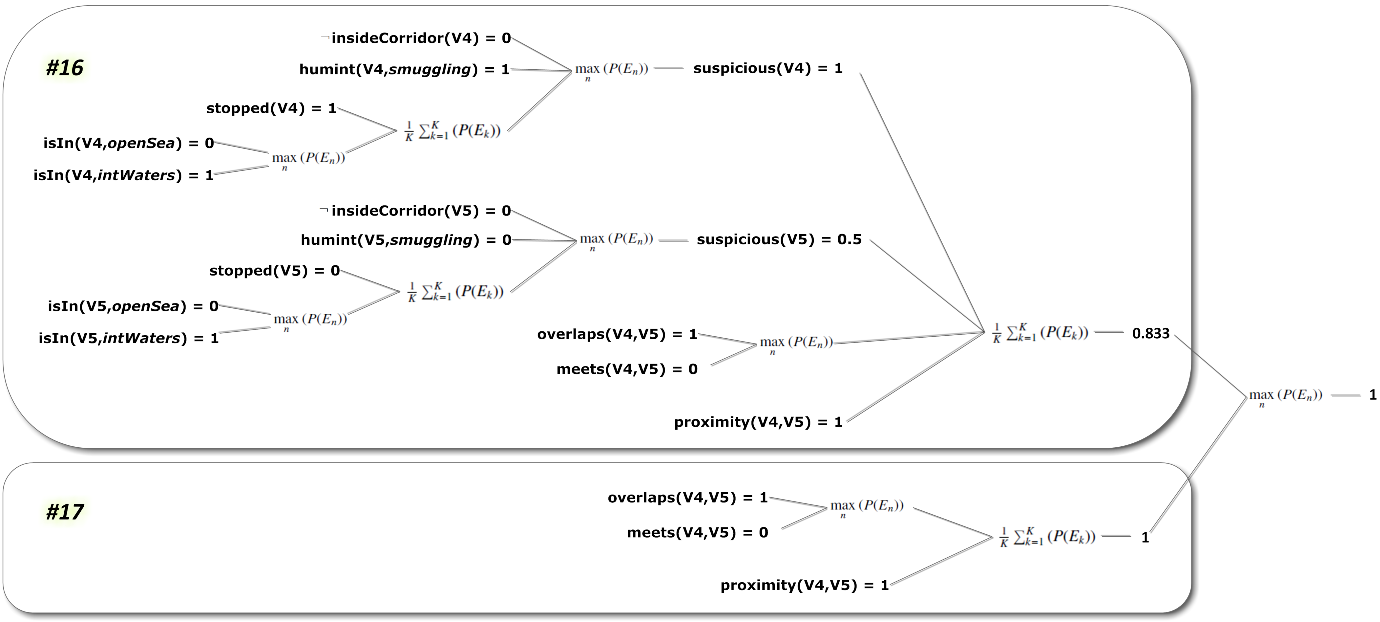


Figure 10: Example of level of completion calculation for the *rendezvous* complex event of formulas #16 and #17 in Table 3.

Table 16: Results for alarm raising in the "hazmat" scenario.

	V_1	V_2	V_3	V_4	V_5
V_1		0.05	0	0	0.01
V_2	0.05		0.95	0.18	0.01
V_3	0	0.95		0.02	0.89
V_4	0	0.18	0.02		0.51
V_5	0.01	0.01	0.89	0.51	

Table 17: Results for the "hazmat" scenario without contextual information.

	V_1	V_2	V_3	V_4	V_5
V_1		0.01	0.23	0.17	0.01
V_2	0.01		0.33	0.37	0
V_3	0.23	0.33		0.34	0.32
V_4	0.17	0.37	0.34		0.32
V_5	0.01	0	0.32	0.32	

We performed two tests with the MLN: in both cases the evidence set is the same, but the contextual information is completely missing in the second experiment. We tested all the possible vessels combinations, and, when contextual information is provided, the reasoner set an alarm in the case of (V_2, V_3) and (V_3, V_5), matching the truth (Table 15). Contrarily, no alarm is risen when context is missing, as the values for suspicious cargo ships are low.

4.3. Completion level of complex event example

Two examples of completion calculation are provided in the following. Figure 9 shows the first case where formula #15 in Table 12 has been grounded according to observed evidence. The antecedent of formula is entirely composed by a conjunction of predicates. The literal $neighbours(V1, V2)$ corresponds to a complex event whose components are expanded on the left. All the component simple events are direct observations or contextual knowledge that, in the example, have resulted being true with complete certainty. The remaining literals in formula #15 are direct evidence or contextual knowledge where the predicate $concurrent(V1, V2)$ has been evaluated as being false. The level of completion of this formula is simply calculated as the average of all probabilities with a resulting 85% that indicates that the *alarm* complex event is close to fully occur. However, as can be seen in Table 16, the probability associated to the event by the Markov network is 0.05. The two results together provide useful information to the operator as the Markov network is correctly not considering $V1$ and $V2$ as being currently a threat thus not rising unnecesary false alarms. However, the level of completion could be useful to monitor how events are evolving in the scenario. In this case, it is highlighting a dangerous condition close to be fully occurring. The level of completion could also provide useful information for a posteriori forensic analysis.

A second example of calculation is shown in Figure 10 illustrating the grounding of formulas #16 and #17 in Table 3. The example is functional in showing the calculation when multiple formulas refer to the same complex event. In this case, formula #17 defines a low weight condition for the *rendezvous* complex event that is defined as two vessels that are in the same area in overlapping time intervals. Formula #16 provides a much stronger condition by considering also HUMINT information on the two vessels. The level of completion of a complex event that appears as consequent of two or more formulas is calculated by applying the rules described above in this section to the disjunction of the antecedents of the formulas. Figure 10 shows how, given accrued evidence on vessels $V4$ and $V5$, a *rendezvous* event has actually occurred. Specifically, formula #17 is complete while #16 is at 83%. The output probability for the *rendezvous* event for vessels is 0.01 as can be seen in Table 6. This means that the event has occurred but is not considered dangerous (i.e. one of the two vessels has not stopped and is not considered suspicious by HUMINT reports).

4.4. Discussion

Analysing the results of the previous experimental section, we can state that MLNs constitute an improvement and a promising addition to the maritime situational awareness literature.

First of all, output probabilities can be associated to simple and (unobserved) complex events, and, more in general, to every predicate in the knowledge base. Together with the level of completion, this feature constitutes an important information, as it allows the operator to assess the level of risk associated to the events, and the percentage of their completion for further analysis.

All FOL based approaches are problematic in the case of an inconsistent knowledge base because a single inconsistency leads to the inconsistency of the entire knowledge base from which anything can follow. Many approaches have been attempted at paraconsistent logics to allow for local inconsistency without global inconsistency. A survey can be found in [42]. However, these techniques generally tackle the problem by weakening classical logic (e.g. negation) and thus loosing expressiveness. Loops, contradictions and inconsistencies between rules are, contrarily to what happens in most frameworks (fuzzy logic, DLs, basic expert systems, etc.), handled autonomously by weighing the evidence supporting the formulas [18] as per (3). This means that the framework can be used even in presence of an inconsistent KB as it would likely be the case when merging the knowledge from multiple sources or collecting it from different experts. Probabilities are involved also in the construction of the knowledge base, as they are associated to the each formula: contradictory or subjective experts formulation of the problem can have associated an "a priori" uncertainty, which represents the degree of confidence they assigned to the relational knowledge among objects and entities in the scenario.

Moreover, differently from expert systems or rigid if-then-else rules, which require all the evidence to be provided simultaneously to make a decision, the chosen framework and the definition of completion of events are here developed to satisfy the operator's need for detecting potentially anomalous events while they are still in progress even in presence of incomplete evidence. The maritime field is favourable in stressing how data and information coming from heterogeneous sources can be fused together within the MLN framework; in fact, the domain is characterized by events that typically span a considerable amount of time and that slowly evolve as new information is acquired while they are still happening. This implies that the system should work with partial or missing observations, and not being jeopardized by them. This aspect is really valuable for timely anomaly and threat prediction, as the status of the situation can be foreseen in advance and updated as new information is acquired.

Another advantage over the state-of-the-art systems is the possibility to query an arbitrary formula and get probabilities as output. For instance, the operator may want to build on-the-fly a custom query and test it in real-time. In the rendezvous example, the query for indirect predicates as $suspicious(v)$, with $v = \{V1, \ldots, V5\}$ returns the probability for each vessel to be suspicious, given the evidences, and more complex formulas, as $[(suspicious(v) \vee suspicious(y)) \wedge proximity(v, y)]$, allow to better investigate the situation for couples of vessels.

Finally, the experimental results highlight the importance of context as a key element in a real-world domain for a timely, complete and accurate situation assessment. In absence of context, the anomaly detection performance dramatically decreases; therefore, we can say that context provides the necessary discriminative power to achieve correct inferences.

5. Conclusions

Events and anomalies are fundamental concepts to identify and comprehend critical occurrences in the observed environment. Proper event detection and understanding facilitate situation assessment and human decision making with applications to safety, security, consequence management and recovery among others. In this paper, we examined this concept in the maritime domain in the scope of the JDL fusion model terminology.

Furthermore, we employed Markov Logic Networks as an efficient and robust tool which leverages both the expressive power of FOL and the probabilistic uncertainty management of Markov Networks. This allowed us to encode uncertain a priori and contextual knowledge, fuse data coming from multiple and heterogeneous sources of information and perform reasoning on incomplete data. In particular, observed and contextual evidences are fed into the inference engine and reasoning is thus performed combining evidence from low-level data and high-level information. The latter, represented in our case by logic formulas which refer to domain entities and their relationships, is a key element to reduce uncertainty and achieve a more complete and accurate situational awareness picture. The results confirm this rationale and encourage further developments on more complex scenarios.

We have also provided a mechanism for early event detection by evaluating the level of completion of complex events. This could be useful to provide early warnings before hazardous conditions actually take place.

References

[1] L. Snidaro, I. Visentini, G. Foresti, Studies in Computational Intelligence, Vol. 336/2011, Springer-Verlag, 2011, Ch. Data fusion in modern surveillance, pp. 1–21.

[2] B. Ristic, B. La Scala, M. Morelande, N. Gordon, Statistical analysis of motion patterns in ais data: Anomaly detection and motion prediction, in: Proceedings of the 11th International Conference on Information Fusion, 2008.

[3] C. Carthel, S. Coraluppi, P. Grignan, Multisensor tracking and fusion for maritime surveillance, in: Proceedings of the 10th International Conference on Information Fusion, 2007.

[4] M. Riveiro, G. Falkman, T. Ziemke, T. Kronhamn, Reasoning about anomalies: a study of the analytical process of detecting and identifying anomalous behavior in maritime traffic data, Proceedings of SPIE Defense, Security, and Sensing 2009 SPIE Volume 7346, 73460A.

[5] C. Brax, L. Niklasson, Enhanced situational awareness in the maritime domain : An agent-based approach for situation management, in: Intelligent Sensing, Situation Management, Impact Assessment, and Cyber-Sensing : Proceedings of SPIE Defense, Security, and Sensing 2009, SPIE, 2009.

[6] R. Lane, D. Nevell, S. Hayward, T. Beaney, Maritime anomaly detection and threat assessment, in: Proceedings of the 13th International Conference on Information Fusion, 2010.

[7] R. Carvalho, R. Haberlin, P. Costa, K. Laskey, K. Chang, Modeling a probabilistic ontology for maritime domain awareness, in: Proceedings of the 14th International Conference on Information Fusion, 2011.

[8] J. Roy, Automated reasoning for maritime anomaly detection, in: Proceedings of the NATO Workshop on Data Fusion and Anomaly Detection for Maritime Situational Awareness (MSA 2009), NATO Undersea Research Centre (NURC), La Spezia, Italy, 2009.

[9] J. Roy, Rule-based expert system for maritime anomaly detection, in: Proceedings of the 12th International Conference on Information Fusion, 2009.

[10] J. Roy, M. Davenport, Exploitation of maritime domain ontologies for anomaly detection and threat analysis, in: Waterside Security Conference (WSS), 2010 International, 2010.

[11] S. J. Russell, P. Norvig, Artificial Intelligence - A Modern Approach (3rd ed.), Vol. I-XVIII, Pearson Education, 2010.

[12] J.Llinas, C. L. Bowman, G. L. Rogova, A. N. Steinberg, E. L. Waltz, F. E. White, Revisiting the JDL data fusion model II, in: Proceedings of the Seventh International Conference on Information Fusion, Vol. II, Stockholm, Sweden, 2004, pp. 1218–1230.

[13] L. M. Munson, Looking for anomalies all in the wrong places, USNI Proceedings Magazines 135/7/1,277.

[14] I. Visentini, L. Snidaro, Integration of contextual information for tracking refinement, in: 14th International Conference on Information Fusion, Chicago, Illinois, 2011.

[15] C. Sodtke, J. Llinas, Terrain based tracking using position sensors, in: International Conference on Information Fusion, Vol. 2, 2001, pp. 27–32.

[16] V. Nimier, Supervised multisensor tracking algorithm, in: International Conference on Multisource-Multisensor Information Fusion, Las Vegas, USA, 1998, pp. 149–156.

[17] J. Roy, M. Davenport, Categorization of maritime anomalies for notification and alerting purpose, in: NATO Workshop on Data Fusion and Anomaly Detection for Maritime Situational Awareness, La Spezia, Italy, 2009.

[18] M. Richardson, P. Domingos, Markov logic netwoks, Machine Learning 62 (2006) 107–136.

[19] L. Snidaro, I. Visentini, K. Bryan, G. L. Foresti, Markov logic networks for context integration and situation assessment in maritime domain, in: Proceedings of the 15th International Conference on Information Fusion, Singapore, 2012, pp. 1534–1539.

[20] L. Snidaro, M. Belluz, G. Foresti, Domain knowledge for surveillance applications, in: Proceedings of the Tenth International Conference on Information Fusion, Quebec City, Canada, 2007.

[21] W. R. van Hage, V. Malaise, R. H. Segers, L. Hollink, G. Schreiber, Design and use of the simple event model (sem), Web Semantics: Science, Services and Agents on the World Wide Web 9 (2).

[22] J. Roy, Anomaly detection in the maritime domain, in: Optics and Photonics in Global Homeland Security IV. Proceedings of the SPIE, Vol. 6945, 2008, pp. 69450W–69450W–14.

[23] L. Snidaro, C. Piciarelli, G. L. Foresti, Fusion of trajectory clusters for situation assessment, in: Proceedings of the Ninth International Conference on Information Fusion, Florence, Italy, 2006.

[24] E. H. Shortliffe, B. G. Buchanan, A model of inexact reasoning in medicine, Mathematical Biosciences 23 (3–4) (1975) 351–379.

[25] D. A. Waterman, A guide to expert systems, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1985.

[26] G. Rogova, E. Bossé, Information quality in information fusion, in: Proc. of the 13th. International Conference on Information Fusion, Edinburgh, U.K., 2010.

[27] T. Lukasiewicz, U. Straccia, Managing uncertainty and vagueness in description logics for the semantic web, Journal of Web Semantics 6 (4) (2008) 291–308.

[28] P. Costa, R. Carvalho, K. Laskey, C. Park, Evaluating uncertainty representation and reasoning in HLF systems, in: Proceedings of the Fourteenth International Conference on Information Fusion (Fusion 2011), Chicago, IL, USA, 2011.

[29] A. Patcha, J. Park, An overview of anomaly detection techniques: Existing solutions and latest technological trends, Comput. Netw. 51 (2007) 3448–3470.

[30] X. Li, F. Porikli, A hidden markov model framework for traffic event detection using video features, in: International Conference on Image Processing (ICIP), Vol. 5, 2004, pp. 2901–2904.

[31] G. de Vries, V. Malais, M. van Someren, P. Adriaans, G. Schreiber, Semi-automatic ontology extension in the maritime domain, in: Proceedings of twentieth Belgian-Dutch Conference on Artificial Intelligence (BNAIC 2008), 2008.

[32] J. Garcia, J. Gomez-Romero, M. Patricio, J. Molina, 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, 2011.

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

[34] R. Orchard, Fuzzy Reasoning in JESS: The Fuzzyj Toolkit and Fuzzyjess, in: Proceedings of the Third International Conference on Enterprise Information Systems (ICEIS), Setubal, Portugal, 2001, pp. 533–542.

[35] F. Smarandache, J. Dezert, Advances and Applications of DSmT for Information Fusion: Collected works, no. v. 3, American Research Press, 2009.

[36] J. Balmat, F. Lafont, R. Maifret, N. Pessel, A decision-making system to maritime risk assessment, Ocean Engineering 38 (1) (2011) 171–176.

[37] T. Imkamon, P. Saensom, P. Tangamchit, P. Pongpaibool, Detection of hazardous driving behavior using fuzzy logic, in: Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2008. ECTI-CON 2008. 5th International Conference on, Vol. 2, 2008, pp. 657 –660.

[38] L. Getoor, B. Taskar (Eds.), Statistical Relational Learning, Adaptive Computation and Machine Learning, MIT Press, Cambridge, Massachussets, 2007.

[39] S. D. Tran, L. S. Davis, Event modeling and recognition using markov logic networks, in: Proceedings of the European Conference on Computer Vision (ECCV), Vol. 5303, Springer Berlin / Heidelberg, 2008, pp. 610–623.

[40] D. Hall, M. McNeese, J. Llinas, T. Mullen, A framework for dynamic hard/soft fusion, in: Proceedings of the 11th International Conference on Information Fusion, Cologne, Germany, 2008.

[41] J. F. Allen, Maintaining knowledge about temporal intervals, Communications of the ACM (1983) 832–843.

[42] G. Priest, Handbook of Philosophical Logic, Vol. 6, Kluwer, 2002, Ch. Paraconsistent logic.