A Probabilistic Logic-based Model for Fusing Attribute Information of Objects Under Surveillance*

Steffen Michels Marina Velikova Arjen Hommersom Peter J.F. Lucas

Institute for Computing and Information Sciences
Radboud University Nijmegen, The Netherlands

1 Introduction

The goal of *surveillance* is to detect or predict certain *events*, like accidents or illegal activities, by continuously monitoring the position and behaviour of objects of interest. Those objects can, for instance, be people, cars or ships. The monitoring requires the collection of information from various and heterogeneous *sources*.

Typical sources employed in surveillance tasks, with respect to the detection of the object's position and movements (so-called kinematic data), are radars, transponders or optical sensors. Measurement errors of such sensors can be characterised by statistical methods and fusing multiple independent sensors can correct for those errors. A lot of work has already been done for automating this [8, 9, 13].

In addition to the kinematic analysis, objects need to be *classified* and *identified*. The goal is finally to get to know whether an object is or will be involved in some relevant event. Evidence for this can be given by an object's behaviour, physical properties like size and type or involvement in past events. The identity, which means objects' names or identification numbers, is very valuable information as well, to be able to link information from various sources to objects.

Gathering and handling those kind of non-kinematic information is a complex task, characterised by a number of challenges:

- Information comes from different **heterogeneous sources** with possibly unknown characteristics. Examples range from personal observations or communication of a human operator, performing a surveillance task, to databases or websites. Information is provided in different formats, terminology and might be contradicting. It is hard to judge how much one can trust a source. For instance, social network websites might give very recent information one can not find somewhere else, but it is hard to judge whether provided information is true or not.

- The amount of gathered **information is unknown in advance**. Whether and what information is provided by the sources differs per object. Also the range of possible values one gets is unpredictable, since many are strings like names.

- Wrong information is not only the result of random errors, but can also be due to **intentional misinformation**. This makes it more difficult to characterise errors by statistical methods, since the probability that information is correct depends on intentions of information providers. In the current context, an intention is defined as a course of action that one has planned on and is committed to follow.

- It is uncertain whether a piece of information provides **information about an object of interest**. Information can not straightforwardly been put together, since it might not always be obvious whether a piece of information tells something about the object one is interested in. This is especially true because the identity of objects is virtually always uncertain. For instance, a source might give

*This publication was supported by the Dutch national program COMMIT. The research work was carried out as part of the Metis project under the responsibility of the Embedded Systems Institute with Thales Nederland B.V. as the carrying industrial partner.

information about different objects if queried for either a number or the name, which are about the same object according to another source. A query can even give multiple answers if different objects can have the same name.

The sheer amount of information leads to *information overload* of human operators. Additionally, for humans it is very hard to reason about uncertain information in a consistent, unbiased way. We are therefore seeking for a way to support human surveillance operators by automating fusing information and reasoning about it.

We tackle the problem by developing a probabilistic model to fuse information about objects under surveillance. We found a probabilistic logic to be a suitable choice of modelling language in this research. There are several reasons for choosing a first-order formalism, in contrast to a propositional formalism such as Bayesian networks [12]. A first-order formalism allows to define abstract general rules, for instance about properties of objects or reported evidence, without the need to define similar structures for all possible instantiations. First-order rules also allow to deal with a number of evidence facts which is unbounded and unknown in advance.

Another criterion is that the specific probabilistic logic language we use provides a natural way to represent domain knowledge. This is important since in the domain we often lack sufficient data with known ground truth we could learn models from. It is for instance virtually impossible to get a data set of vessels indicating which one has actually been smuggling. We therefore want to be able to incorporate domain knowledge and estimates for relation for which not sufficient data is available. Specifically uncertainty is expressed as probabilities whose meaning can be interpreted locally and not weights which only have a meaning in combination with weights of other rules (e.g. as in *Markov logic* [6]). Furthermore, it has been shown that the distribution defined by a Markov logic theory depends on the size of the domain [11], which is a major issue in the surveillance domain as the number of relevant objects is not fixed in advance.

Surveillance tasks in the maritime context serve as a motivating example. Examples of information important in this domain are vessel identities and types, smuggling events or whether a ship tries to hide its identity. The contributions presented in this report are twofold.

Research:

- We develop a general framework of a model based on probabilistic logic for fusing non-kinematic information. We take into account possible errors in the information, but also the uncertainty whether information is about a particular object and propose a solution for dealing with attributes having dynamic ranges of values.

- We show that intentions of objects can be handled in a systematic manner within the same framework and that knowledge can guide and improve the fusing process.

Application:

- We apply our framework to build a model for decision support in maritime surveillance. and show that the general rules can be extended with domain specific knowledge. We experimentally evaluate the quality of the model's results and support our claim that reasoning about intentions improves the fusing process. A first prototype of the model has been integrated in a real-world setting, which is the mission management system of our industrial partner[1].

- To our knowledge this is one of the very few real-world applications of probabilistic logics in real-world settings.

Fusing information requires that this information, using different representation formats and terminology, is semantically aligned to a common information model. We do not tackle the issues related to this preprocessing step in this research. For the purpose of this research we assume that all the information is aligned to a common information model, e.g. the Maritime Information Exchange Model (MIEM)[2].

In this report we first give an overview of the example domain, together with a scenario and the basic idea of how human operators could be supported handling such scenarios in Section 2. We then

[1] `http://www.thalesgroup.com`
[2] `https://www.niem.gov/communities/maritime`

in Section 3 introduce the knowledge representation language we use for building the model. Section 4 describes the general model structure together with examples of how maritime specific domain knowledge can be represented. We then experimentally evaluate the model and support our basic claims in Section 5. Finally, we discuss related work and conclude in Sections 6 and 7.

2 Maritime Surveillance

We first give a short introduction to the maritime domain and then give a scenario in which an operator can be supported by our probabilistic model.

2.1 Domain

Vessels are identified by a number of basic properties:

- *MMSI*: a unique 9-digit Maritime Mobile Service Identity number, which may change over time.
- *IMO*: a unique 7-digit International Maritime Organization number, which is assigned only to sea-going merchant ships of 100 gross tons and above, and it is fixed for the entire vessel's lifetime.
- *Name*: an arbitrary string, which does not uniquely identify a vessel.
- *Flag*: the flag of the country where the vessel is registered. It can be derived from the first three digits of the MMSI, the so-called *Maritime identification digits* (MID).
- *Type*: the vessels differ in size, the cargo they carry and waterways on which they navigate. Examples include dry bulk cargo, passenger, tanker.

Vessels provide such identification information along with their position using the *Automatic Identification System* (AIS). Since the information send by AIS can be manipulated arbitrarily, in maritime surveillance one cannot trust it, especially if the vessel has a reason to hide its identity. It can, however, be evaluated using additional information from sources such as databases and websites. Examples include *IHS Fairplay*[3]—a commercial database containing detailed vessel information, and *marinetraffic.com*—a free-to-use website that provides real-time ship tracking information. Additional information might come for instance from other operators via tactical chats or intelligence authorities.

As in surveillance in general, in the maritime domain intentions and behaviour of the objects of interest play crucial role for detecting abnormal events. It is known, for example, that vessels being involved in illegal activities, such as smuggling or hijacking, may try to hide their identity via vessel repainting on sea. Evidence about those intentions influence how the information about vessel's properties is judged. A vessel that tries to hide its identity is not likely to transmit correct AIS information.

There is always uncertainty about the actual properties or intentions of the vessel, which complicates the decision-making in the maritime domain. This is clearly illustrated in the next section, where we introduce a scenario from the maritime domain, which is used as an example throughout this report.

2.2 Scenario

A coast guard operator has got an intelligence report that within two days a vessel named "Black Pearl" is about to enter the zone under surveillance with smuggling goods on board. Therefore, the operator starts examining carefully the vessels one by one within the area of interest. The problem is that there can be multiple vessels with this name and the smuggling vessel might hide its identity by transmitting a wrong name.

The operator examines the smuggling vessel, of course, not knowing that it is the smuggling one. The vessel has the following true identity information, which is also unknown to the operator:

$$MMSI = 123456789, Name = \text{“}Black\,Pearl\text{”}$$

The vessel transmits an AIS message with the following information:

$$ais1 : MMSI = 123456789, Name = \text{“}Dutchman\text{”}$$

[3] `http://www.ihs.com`

To verify this, the operator retrieves additional information from *IHS Fairplay* and *marinetraffic.com*:

$$fairplay1 : MMSI = 123456789, Name = \text{“}Black\ Pearl\text{”}$$
$$marinetraffic1 : MMSI = 987654321, Name = \text{“}Dutchman\text{”}$$
$$marinetraffic2 : MMSI = 123456789, Name = \text{“}Black\ Pearl\text{”}$$

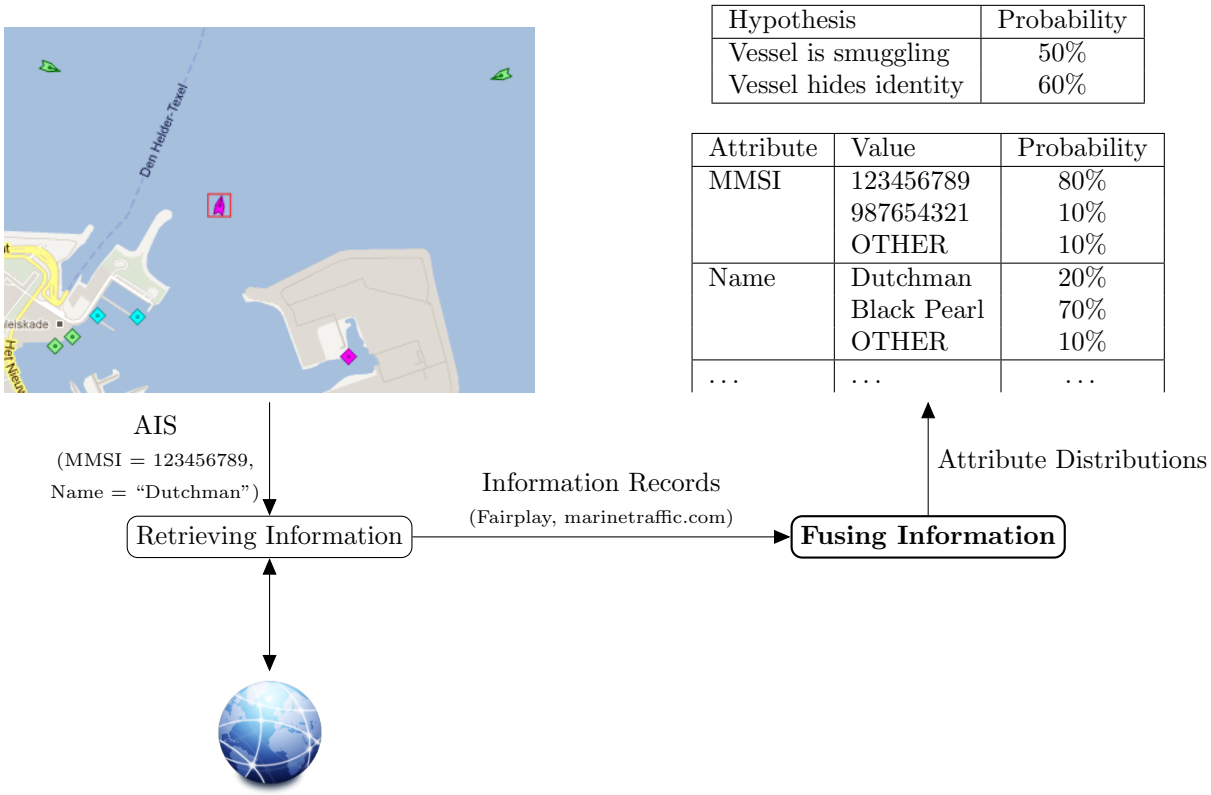
There is obviously a contradiction between the names reported by AIS and by *Fairplay*, and there might be several possible interpretations, e.g., (i) the vessel might send out a wrong name due to input error or intentionally, (ii) the name in the *Fairplay* record is wrong or (iii) the MMSI send by the vessel is wrong and the *Fairplay* record is about a different ship. The first *marinetraffic.com* record is likely not to be about the vessel under examination, while the second one confirms the *Fairplay* information.

Given this information, the operator suspects that the vessel tries to hide its identity by sending a wrong name, and thus it might be the vessel involved in smuggling. However, to verify this hypothesis with certainty the operator requires a patrolling boat to visually observe the vessel at distance. The information reported back is that the vessel has been repainted on sea, which gives the operator further support to the hypotheses for hiding identity and smuggling of the vessel.

Although this scenario is a simplified version of the reality, it clearly illustrates the complexity of maritime surveillance tasks, including retrieving and reasoning about information from heterogeneous sources. This requires automated approaches to support human operators in their daily operational work, where hundreds of ships and dozens of properties are to be examined. In this report we propose a probabilistic model that makes a first step towards such automated support and in the next section we illustrate the model's working principles.

2.3 A Probabilistic Decision Support Model for Maritime Surveillance

Figure 1 presents an example scheme for the operational work of a maritime surveillance system with an embedded decision-support model.



Hypothesis	Probability
Vessel is smuggling	50%
Vessel hides identity	60%

Attribute	Value	Probability
MMSI	123456789	80%
	987654321	10%
	OTHER	10%
Name	Dutchman	20%
	Black Pearl	70%
	OTHER	10%
...	...	...

Figure 1: Example Scenario

Initially, the operator selects a single vessel of interest, whose AIS identity information, as given in the scenario, is used to retrieve data from additional sources. The retrieved information serves as an input to a probabilistic model for fusing the information. The result of the model is a probability distribution of the vessel's properties and its intentions given the obtained information. It is always possible that the vessel's true properties have values we do not observe or cannot retrieve. So the distribution of all properties contain the special value `OTHER`, representing that case. The tables in Figure 1 present a possible outcome from the model for the situation described in the scenario.

As described in the scenario, additional information like that the ship has been repainted gives evidence for smuggling and lowers the trust in AIS. Adding this information therefore results in changes of all probabilities shown in the tables.

3 Preliminaries

We use normal logic programs as representation language and add random variables to handle uncertainty.

3.1 Logic Programs

Our language is based on logic programs. We obey *Prolog* conventions: constants are starting with lower case (e.g. $tanker$, ais, ...) and variables with upper case letters ($Attr$, Src, ...). The placeholder _ is used for variables without name which are not referred to elsewhere. Terms consist of a functor and a number of arguments. A logic program LP consists of rules, given by a head and body separated by $\leftarrow$. Conjunction ($\wedge$) and disjunction ($\vee$) are used as composition for building bodies. We additionally use equality ($=$) and inequality ($\neq$) operators for constants.

A query q succeeds in case it can be derived from the program. We formalise this with the following indicator function:

$$success(q) = \begin{cases} 1 & \text{if } LP \models q \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

EXAMPLE 1

Examples of rules are:

$$suspicious(Vessel) \leftarrow in\_fishing\_area(Vessel) \wedge type(Vessel, Type) \wedge Type \neq fishing$$
$$criminal(Person) \leftarrow pirate(Person) \vee smuggler(Person)$$

Under the assumption that $pirate(person1)$ the query $criminal(person1)$ is positively answered.

To make it more convenient to specify rules for special cases we introduce the notation $\leftarrowtail$ to separate head and body. This means that in case the rule matches, all rules defined after are not used. Note that this does not destroy the declarative character of our language, it merely makes it unnecessary to explicitly define the cases in which a general rules matches, which becomes impractical in case there are a large number of special rules which should be used instead of the general one.

EXAMPLE 2

Assume there is a general rule for predicate $p(Vessel)$ and one wants to add a special rule for the "Black Pearl". This can be done like this:

$$p(black\_pearl) \leftarrowtail \ldots$$
$$p(Vessel) \leftarrow \ldots$$

The same could be achieved without the $\leftarrowtail$ notation:

$$p(black\_pearl) \leftarrow \ldots$$
$$p(Vessel) \leftarrow Vessel \neq black\_pearl \wedge \ldots$$

3.2 Random Variables

We add random variables to our language to handle uncertainty. Random variables have a fixed number of possible values with attached probabilities. Initially all random variables are independent, dependencies are expressed by the structure of the logic program. In each *possible world* one single value is chosen for each random variable. The current value of a random variables can be used within the logic program.

The syntax we use is similar to the one of distributional clauses (DC) [10], but we restrict it to discrete, finite random variables in this research. Random variables are denoted starting with lower case letters and can additionally have arguments, e.g. $\mathbf{r}$ or $\mathbf{r}(A_1, \ldots, A_N)$. A distributional clause defining such random variables has the form $\mathbf{r}(A_1, \ldots, A_N) \sim \{p_1 : c_1, \ldots, p_N : c_N\} \leftarrow \ldots$. This defines a single random variable with given distribution for each distinct grounding of $A_1, \ldots, A_N$. The distribution may depend on the body and therefore a different distribution can be computed for each grounding. We assume a finite set of possible grounding, which means a program defines a finite set of random variables we denote with the set $\mathcal{V}$.

Finally, Random variables are used in clauses by mapping them to their value using the operator $\simeq$. For instance, a predicate $p(X, Y)$ that is true if random variable $\mathbf{v}(X)$ has value Y, can be defined as:

$$p(X, Y) \leftarrow \simeq \mathbf{v}(X) = Y$$

EXAMPLE 3

Given that $p(a, 0.6)$ and $p(b, 0.2)$ are true and we define the following DC:

$$\mathbf{r}(X) \sim \{P : v1, 1 - P : v2\} \leftarrow p(X, P)$$

This DC then defines the following two random variables in case q is $\simeq r(\_) = v1$:

$$\mathbf{r}(a) \sim \{0.6 : v1, 0.4 : v2\}$$
$$\mathbf{r}(b) \sim \{0.2 : v1, 0.8 : v2\}$$

In case the query is $\simeq r(a) = v1$ only the first random variable $\mathbf{r}(a)$ is in the set of random variables $\mathcal{V}_q$.

Such random variables can naturally represent exclusive states, like the properties we consider which can have only one value at a time. For instance, a vessel has only one identification number. In that sense they are similar to *logic programs with annotated disjunctions* [16].

In this research we enforce the additional constraints to make sure that the probabilities of all possible values a random variable can take sum up to one. The same constraint is enforced for instance in Bayesian networks. To achieve this, first the probabilities $p_1, \ldots, p_N$ in each definition $\{p_1 : c_1, \ldots, p_N : c_N\}$ must sum up to one. Second, in case another random variable is used in the bodies of rules defining that random variable, the rules must be exclusive and exhaustively cover all possible values of that other random variable.

EXAMPLE 4

Given this definition of q:

$$\mathbf{q} \sim \{0.5 : a, 0.5 : b\}$$

A proper definition of p could be:

$$\mathbf{p} \sim \{0.9 : a, 0.1 : b\} \leftarrow \simeq \mathbf{q} = a$$
$$\mathbf{p} \sim \{0.2 : a, 0.8 : b\} \leftarrow \simeq \mathbf{q} = b$$

We do not use definitions with one rule missing in this research.

3.3 Query Success Probability

We define the probability $P(q)$ that a query q succeeds. We start defining the probability of a single complete choice of all random variables. Such a choice c is a function which selects a probability-value pair $(p : v)$ for each element of $\mathcal{V}$.

A logic program PL_c can be assigned to each choice, in which all DCs are replaced by a rule. The head of such a rule assigns a value to the random variable. The DC $\mathbf{r}(A_1, \ldots, A_N) \sim \{p_1 : c_1, \ldots, p_N : c_N\} \leftarrow \ldots$ is replaced by $\simeq\mathbf{r}(A_1, \ldots, A_N) = v \leftarrow \ldots$ where $(p : v) = c(\mathbf{r})$.

The probability of a choice P_c is the product of all probabilities in the choice, since we assume them to be independent:

$$P_c = \prod_{\mathbf{r} \in \mathcal{V},\, c(\mathbf{r}) = (p:v)} p \tag{2}$$

The probability of a query $P(q)$ is finally defined by the sum of the probabilities of all possible choices for which q can be derived from PL_c. We denote the set of all possible choices with $\mathcal{C}$ and use the indicator function $csuccess(q, c)$, meaning that query q can be derived under the choice c:

$$csuccess(q, c) = \begin{cases} 1 & \text{if } LP_c \models q \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The success probability of a query is finally defined as:

$$P(q) = \sum_{c \in \mathcal{C}} P_c \cdot csuccess(q, c) \tag{4}$$

3.4 Conditional Success Probabilities

One usually does not only want to know the success probability of a query for the general case, but wants to use evidence about a particular case to update that probability. For instance, one wants the probability that an object has a certain property, given the available information about it.

We therefore want to compute the probability of the query q given evidence e, denoted by $P(q|e)$. The probability can be defined using basic probability theory:

$$P(q|e) = \frac{P(q \wedge e)}{P(e)} \tag{5}$$

The two separate probabilities are defined as done before.

3.5 Syntactic Sugar

We use some syntactic sugar to denote special finite distributions for readability:

$$\begin{aligned} constant(C) &\equiv \{1.0 : C\} \\ uniform(\{V_1, \ldots, V_N\}) &\equiv \{\frac{1}{N} : V_1, \ldots, \frac{1}{N} : V_N\} \\ uniform\_other(\{V_1, \ldots, V_N\}, M) &\equiv \{\frac{1}{M} : V_1, \ldots, \frac{1}{M} : V_N, \frac{M-N}{M} : other\} \\ flip(P) &\equiv \{P : true, 1 - P : false\} \\ combination(\{w_1 : Dist_1, \ldots, w_N : Dist_N\}) &\equiv w_1 \cdot Dist_1 \cup \ldots \cup w_N \cdot Dist_N \end{aligned}$$

The special distribution $uniform\_other$ represents a uniform distribution over M possible states of which only $V_1, \ldots, V_N$ are known. All other possible, unknown values are represented by *other*. The distribution *combination* is used to combine different distributions and weight them according to the given weights. Multiplication of a weight with a distribution means multiplying all probabilities in the distribution with the weight.

4 Probabilistic Model

The goal of our probabilistic model is to reason about intrinsic properties and intentions of objects given information records as evidence. We first describe how we represent objects and their properties and intentions and information records. Then we model the dependency between them, making it possible to reason about what given information records tell about the object's actual properties and intentions.

4.1 Objects and Attributes

We assume there is a set of uniform objects in the real world we are interested in. In the maritime surveillance domain, such objects are typically vessels. We refer to the single object we are currently interested in as object of interest (OoI) and denote it with the special constant *ooi*. This allows us to model properties and intentions, and the information about them in the same uniform way.

As discussed, in surveillance tasks we are interested in both the intrinsic properties and intentions of objects. For vessels these can be the name, MMSI, type or intention for smuggling. Although properties and intentions of objects are semantically different, from a modelling point of view in this research we do not make a distinction between them and refer to both as *attributes*. For instance, a record from a vessel database and an intelligence report that a ship with a certain name is smuggling can be expressed using the same representation.

Attributes behave like functions, which means each object attribute can only have one value at a time. We therefore represent the value of a single attribute *Attr* or object *Obj* as a single random variable:

$$\mathbf{attr}(Obj, Attr) \tag{6}$$

EXAMPLE 5

The attributes of the OoI from the running example (Section 2.2) are represented as:

$$\simeq\mathbf{attr}(ooi, mmsi) = 123456789$$
$$\simeq\mathbf{attr}(ooi, name) = \text{“}Black\ Pearl\text{”}$$

Each attribute has a domain, which means the possible values it can take. We distinguish between fixed and dynamic domain. The first case applies to attributes with a domain which can be enumerated straightforwardly. Examples are a vessel's type or whether the vessel is smuggling. That the fixed domain of attribute *Attr* consists of values $v_1, \ldots, v_N$ is represented as:

$$domain(Attr, fixed(\{v_1, \ldots, v_N\}))$$

There are however attributes for which a fixed domain makes no sense or would result in a model for which inference is infeasible. This is either because all values can theoretically be enumerated but this enumeration would be too large, like for identification numbers of fixed size, or because the possible values are not all known, like for names. Such dynamic domains are represented as follows:

$$domain(Attr, dynamic(\{known_1, \ldots, known_K\}, N))$$

The first part is a set of values $known_1, \ldots, known_K$ for which it is known that the attribute can take them for a particular case. They are determined dynamically based on the information reported. Additionally, there is the estimated number of distinct values in the actual domain. The possible number of distinct identification numbers of fixed size could in principle be derived exactly, but not all numbers may be in use in the real world. So a better estimation can be given.

EXAMPLE 6

The domains of the type attribute and the attribute telling whether a ship is repainted are fixed and represented as:

$$domain(type, \quad fixed(\{cargo, tanker, passenger, \ldots\}))$$
$$domain(smuggling, fixed([true, false]))$$

Given the reported information from the running Example (Section 2.2) the dynamically computed domains would be:

$$domain(mmsi, dynamic(\{123456789, 987654321\}, 1000000))$$
$$domain(name, dynamic(\{\text{“}BlackPearl\text{”}, \text{“}Dutchman\text{”}\}, 500000))$$

For each attribute finally a prior distribution has to be defined. Those distributions represent the prior distribution of attribute values independent of the object or any further knowledge about it. We

distinguish between unconditional prior distributions and conditional distributions depending on another attribute's value.

Unconditional prior distributions are simple prior estimates of the distribution of values objects in the domain take for a particular attribute. This can straightforwardly be expressed for attributes with fixed domain:

$$\mathbf{attr}(\_, Attr) \sim \{p_1 : v_1, \ldots, p_N : V_N\}$$

The prior distribution for attributes with dynamic domain can be derived from that domain. In the common case we assume the probability is uniformly distributed over all values and we use the distribution $uniform\_other$ as described in Section 3.5. We only need a single default rule for this case:

$$\mathbf{attr}(\_, Attr) \sim uniform\_other(Values, N) \leftarrow domain(Attr, dynamic(Values, N))$$

EXAMPLE 7

The prior probability for smuggling can simply be expressed as:

$$\mathbf{attr}(\_, smuggling) \sim flip(0.01)$$

A special case is the IMO, since it is not assigned to all vessels. This can be expressed with a combination of the dynamic distribution with a constant distribution of value $noIMO$:

$$\mathbf{attr}(\_, imo) \sim combination(\{0.3 : uniform\_other(Values, N), 0.7 : constant(noIMO)\}) \\ \leftarrow\!\!\shortmid domain(Attr, dynamic(Values, N))$$

Conditional prior distributions are used to express dependencies between attributes. The formalism allows to define directed causal relationships without circles. Those distributions are defined with rules like:

$$\mathbf{attr}(Obj, Attr) \sim Dist \leftarrow \mathbf{Body}$$

Here **Body** can be any clause defining a prior distribution $Dist$. The distribution can dynamically be defined conditioned on values of other attributes.

EXAMPLE 8

The probability that is vessel is hiding its identity is much higher in case it is smuggling. This can be expressed by:

$$\mathbf{attr}(Obj, hides\_identity) \sim flip(0.7) \quad \leftarrow\!\!\simeq \mathbf{attr}(Obj, smuggling) = true$$
$$\mathbf{attr}(Obj, hides\_identity) \sim flip(0.01) \leftarrow\!\!\shortmid\!\!\simeq \mathbf{attr}(Obj, smuggling) = false$$

Again a $\leftarrow\!\!\shortmid$ is used to prevent the general rules, which has to be defined after, to match.

An example of a conditional distribution for an attribute with dynamic domain is a vessel's flag. It can be derived from the MMSI. To reflect the the prior distribution can be defined as:

$$\mathbf{attr}(Obj, flag) \sim constant(Flag) \leftarrow\!\!\shortmid\!\!\simeq \mathbf{attr}(Obj, mmsi) = MMSI \wedge MMSI \neq other \wedge mmsi\_flag(MMSI, Flag)$$

In case the MMSI is a concrete number the flag can be derived from it. We assume we have the predicate $mmsi\_flag(MMSI, Flag)$ to do this. In case the MMSI is *other* the default rules is used.

4.2 Information Records

We refer to information reported by sources as *information records*, or *records* for short. Formally, a record Rec reported by a source Src is represented as follows:

$$source(Rec, Src)$$

We use the convention that the label of each record contains the source name with an attached number ID, e.g. a record with a label $intel1$ means that it is provided by an intelligence report with an ID of 1. In the maritime scenario such records are for instance $ais1$, $fairplay1$ and $marinetraffic1(2)$.

Each record contains values for a number of attributes. We introduce a random variable to represent the value of an attribute $Attr$ reported in a record Rec:

$$\mathbf{rec\_attr}(Rec, Attr)$$

The fact that values are missing (not reported) can itself be useful input information for the probabilistic model. How to represent that is itself a complex modelling task where different types of missingness need to be considered, e.g., whether a value is not known or it is known but just not reported. While this is a valuable research direction, for the purpose of this research, we focus only the values available (reported) in a record.

EXAMPLE 9

Some representations of the information reported about the vessel from the example in Section 2.2 are:

$$\simeq\textbf{rec\_attr}(intel1, smuggling) = true$$
$$\simeq\textbf{rec\_attr}(ais1, name) = \text{“}Dutchman\text{”}$$
$$\simeq\textbf{rec\_attr}(fairplay1, mmsi) = 123456789$$
$$\simeq\textbf{rec\_attr}(visualsign1, repainted) = true$$

4.3 Establishing A Relation Between Objects & Information Records

The goal of the model is finally to predict the OoI's true attributes using information records. This is done by establishing probabilistic relationships between the object and record attributes based on domain knowledge. A schematic representation of this relation is given in Figure 2.

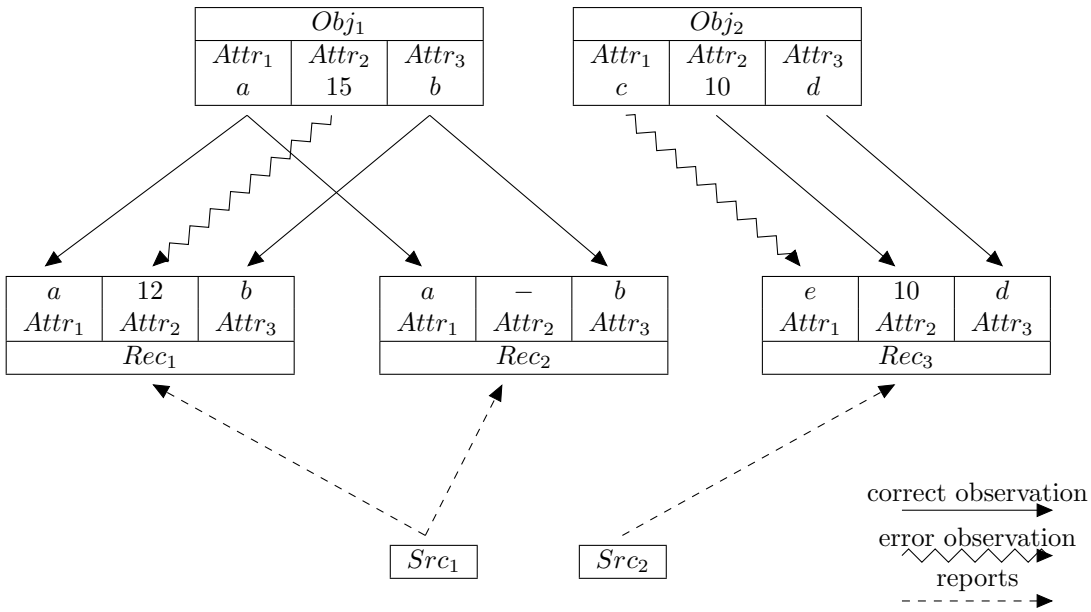


Figure 2: Relation between object & records attributes

As mentioned earlier, a source Src reports one or more records Rec_i, and we assume that each record is always related to a single object Obj. For example, in the figure Rec_1 and Rec_2 are about Obj_1 and Rec_3 is about Obj_2. Attribute values in records are observations of the object's actual attribute values, but records do not have to provide values for all attributes. Rec_2 for instance does not provide a value for attribute $Attr_2$. Records can report erroneous attribute values, which means that the value is not equal to the observed object's actual one. For instance, Rec_1 is about Obj_1 and reports 12 for $Attr_2$, although the attribute's actual value for that object is 10.

We define a binary random variable to indicate whether or not a record attribute is erroneous:

$$\textbf{error}(Rec, Attr)$$

We define a default probability distribution p_d for the error, which is independent of the record or attribute:

$$\mathbf{error}(\_, \_) \sim flip(p_d).$$

Again special rules are possible, to for instance express that we judge the error rate of records from certain sources different. Also more complex dependencies can be expressed.

EXAMPLE 10

Assume we judge the error rate of all records from the *Fairplay* database to be less than the default. This can be expressed by a special case for **error**:

$$\mathbf{error}(Rec, \_)\ flip(0.02) \leftarrow\!\shortmid source(Rec, fairplay)$$

AIS messages are handled in a special way. They are certainly about the OoI sends they are sent out by them. In case the OoI is hiding its identity it will certainly send out a wrong name, since the name is always mentioned in news articles and other reports. There is also a high chance that a wrong MMSI and IMO is sent. There are not many reasons to hide other attributes like the type, since it cannot be used to identity a vessel:

$$\begin{aligned}
\mathbf{error}(ais, name) \sim constant(true) &\leftarrow\!\shortmid \mathbf{attr}(ooi, hides_identity, true)\\
\mathbf{error}(ais, mmsi) \sim flip(0.6) &\leftarrow\!\shortmid \mathbf{attr}(ooi, hides_identity, true)\\
\mathbf{error}(ais, imo) \sim flip(0.7) &\leftarrow\!\shortmid \mathbf{attr}(ooi, hides_identity, true)\\
\mathbf{error}(ais, _) \sim flip(0.1) &\leftarrow\!\shortmid \mathbf{attr}(ooi, hides_identity, true)\\
\mathbf{error}(ais, _) \sim flip(0.05) &\leftarrow\!\shortmid \mathbf{attr}(ooi, hides_identity, false)
\end{aligned}$$

The object a record contains observations about is formalised by the random variable which is defined for each record *Rec*:

$$\mathbf{about}(Rec)$$

We make the assumption that each record is either about the *ooi* or about one of the set of other objects. For each record we introduce an additional object the record is potentially about. The probability p_{ooi} that a record is about the OoI is one divided by the estimated total number of objects. That the record is about the other object has a very high probability consequently, since it represents the choice of the proper object from the set of all other ones. We define the following default rule for **about**. To introduce the additional object we use the record label as object label:

$$\mathbf{about}(Rec) \sim \{p_{ooi} : ooi, 1 - p_{ooi} : Rec\}$$

As for other rules also special cases can be expressed.

EXAMPLE 11

The AIS message and visual observations are certainly about the OoI. This can be expressed by the special rule:

$$\mathbf{about}(Rec) \sim constant(ooi) \leftarrow\!\shortmid source(Rec, ais) \vee source(Rec, visualsign)$$

We can finally define rules for the relation between object and record attributes. There are two cases: the records correctly reports the attribute's value or not. In the first case the reported value is deterministically defined and it equals the true attribute value. It is formally represented as:

$$\begin{aligned}
\mathbf{rec_attr}(Rec, Attr) \sim constant(\simeq\!\mathbf{attr}(Obj, Attr)) &\leftarrow\\
\simeq\!\mathbf{error}(Rec, Attr) = false \wedge\ \simeq\!\mathbf{about}(Rec) &= Obj
\end{aligned}$$

For the second case when the value is erroneously reported the error distribution has to be determined dynamically based on the attribute's domain:

$$\begin{aligned}
\mathbf{rec_attr}(Rec, Attr) \sim Dist &\leftarrow\\
\simeq\!\mathbf{error}(Rec, Attr) = true \wedge\ \simeq\!\mathbf{about}(Rec) &= Obj\\
\wedge\ error_dist(Attr, \simeq\!\mathbf{attr}(Obj, Attr), Dist)&
\end{aligned}$$

The predicate $error\_dist(Attr, Value, Dist)$ computes a distribution for erroneous information, given that $Value$ is the attribute's real value. We define the error distribution of an attribute with fixed domain as uniform distribution above all possible values except the correct one:

$$error\_dist(Attr, Value, uniform(Values \setminus \{Value\})) \leftarrow domain(Attr, fixed(Values))$$

The false information distribution for attribute with dynamic domain is defined analogously, by filtering out the correct value and then using the already discussed distribution $uniform\_other$:

$$error\_dist(Attr, Value, uniform\_other(Values \setminus \{Value\}, N)) \leftarrow domain(Attr, dynamic(Values, N))$$

4.4 Application to the Scenario

We apply our model to the scenario described in Section 2.2. The evidence e we have is formalised as:

$$\begin{aligned}
\simeq\mathbf{rec_attr}(ais, mmsi) = 123456789 \wedge \simeq\mathbf{rec_attr}(ais, name) &= \text{“}Dutchman\text{”} \wedge \\
\simeq\mathbf{rec_attr}(fairplay1, mmsi) = 123456789 \wedge \simeq\mathbf{rec_attr}(fairplay1, name) &= \text{“}BlackPearl\text{”} \wedge \\
\simeq\mathbf{rec_attr}(mtraffic1, mmsi) = 987654321 \wedge \simeq\mathbf{rec_attr}(mtraffic1, name) &= \text{“}Dutchman\text{”} \wedge \\
\simeq\mathbf{rec_attr}(mtraffic2, mmsi) = 123456789 \wedge \simeq\mathbf{rec_attr}(mtraffic2, name) &= \text{“}BlackPearl\text{”}
\end{aligned}$$

We compute probabilities for the relevant queries with this evidence e and with the additional visual observation that the vessel has been repainted $e' = e \wedge \simeq\mathbf{rec\_attr}(visualsign1, repainted) = true$. The following table gives an overview of the rounded probabilities. Probabilities of 0.00000 do not indicate an impossibility, but are rounded very small probabilities:

Query	$P(q\|e)$	$P(q\|e')$
$\simeq\mathbf{attr}(ooi, mmsi) = 123456789$	0.99996	0.99995
$\simeq\mathbf{attr}(ooi, mmsi) = 987654321$	0.00000	0.00000
$\simeq\mathbf{attr}(ooi, mmsi) = other$	0.00004	0.00005
$\simeq\mathbf{attr}(ooi, name) = \text{“}Dutchman\text{”}$	0.00068	0.00031
$\simeq\mathbf{attr}(ooi, name) = \text{“}BlackPearl\text{”}$	0.99927	0.99962
$\simeq\mathbf{attr}(ooi, name) = other$	0.00005	0.00007
$\simeq\mathbf{attr}(ooi, smuggling) = true$	0.05501	0.24912
$\simeq\mathbf{attr}(ooi, hides\_identity) = true$	0.12638	0.59849

The results quantitatively confirms the intuitive line of reasoning sketched in the scenario. The vessel seems to send the wrong name “Dutchman” in order to hide its identity. The model gives a low probability for that name, which decreases even more with additional visual evidence that the ship has been repainted. As expected, the probabilities for smuggling and that the vessel tries to hide its identity increase with the added evidence.

Quantitatively the probabilities might seems somewhat extreme. The reason for that is that we assume all information records to be independent observations of the object. Two independent observations of an event with a very low probability, like that a ship has a certain identification number, gives very high probability that it is the actual number. In other words, the probability that two observations are erroneous and accidentally report the same wrong identification number is very small.

In practice observations may not be that independent, since source may not provide information based on direct observations of the real object. They may get and aggregate information from other sources. How to model and discover such dependencies remains future work.

5 Experiments

We experimentally evaluated our approach using simulated vessel data. We simulated a dataset with 150 vessels, according to the prior distributions defined in the model. We assume that in the model we exactly know the characteristics of the data, e.g. the probability that a record is about the OoI is 1/150 and the error rate of all sources is known. In this first experiment we restrict to the attributes MMSI, IMO and name and do not deal with intentions. Note that only 30% of the vessels have an IMO, the rest have value $noIMO$ for that attribute.

The aim is to check whether our model can correct for errors in AIS data. We used the model to predict a value for each attribute by selecting the value with the highest probability which is not *other*. We then determined the error rate per attribute by comparing to the actual value. For each vessel we simulated AIS messages with error rate varying from 0 to 1. We used the same error rate for all attributes.

We further simulated for each vessel records for the Fairplay database and a website with fixed error rate of 0.02 and 0.03 respectively. All records having the same MMSI, IMO or name as the AIS message were used as evidence. Since the attributes can be reported erroneously in all information records it is possible that multiple records include the same MMSI, IMO or name. The number of records used was therefore dynamic.

The result is illustrated in Figure 3. The predictions made by the model are significantly less erroneous than the AIS information alone.

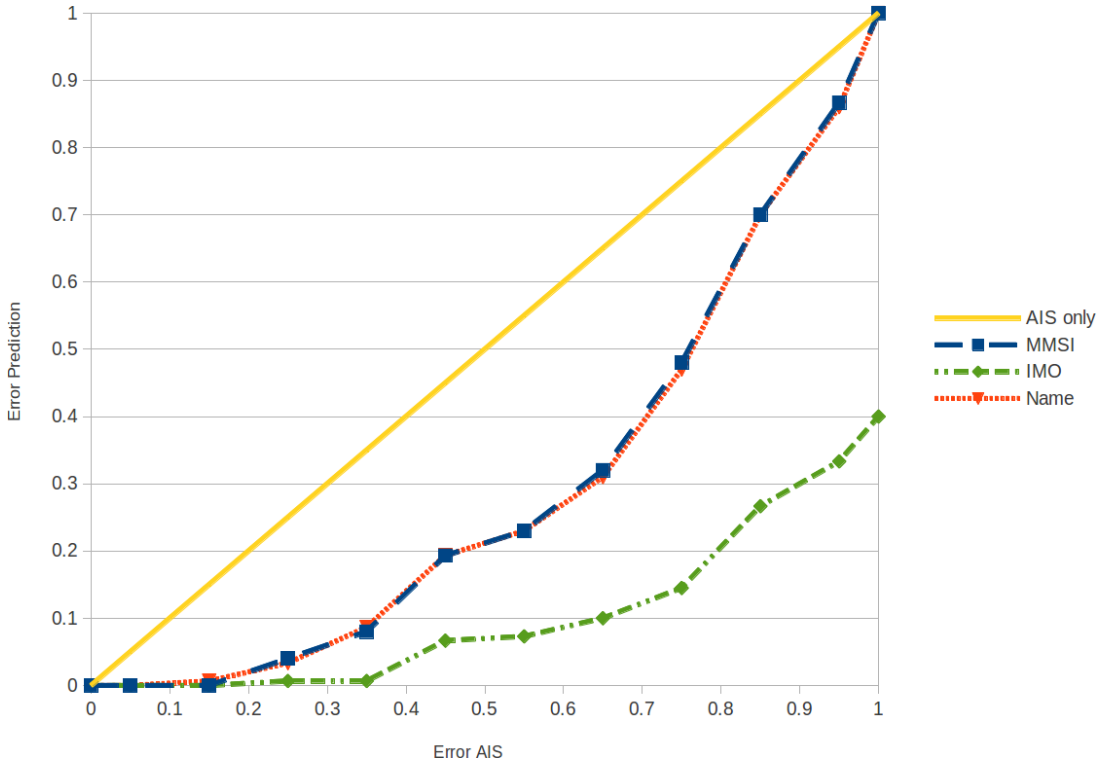


Figure 3: Experiment result

The performance for the MMSI and the name is virtually the same. The fact that the MMSI uniquely identifies a vessel and the name not, plays no role in the data simulated for this small number of vessels. In case the error rate gets too high the model cannot correct errors for the MMSI and name any more.

For the IMO the model performs much better, which is not surprising given that the model contains the knowledge that only 30% of the vessels have an IMO. Simply always predicting $noIMO$ would result in a error rate of only 0.3.

The model could in the experiment correct for all errors in case the error rate is around 0.05, which is an error rate not expected to be higher in practice. As in practice we do not know the true distribution of the data, the results cannot be interpreted as the performance the approach would have in a realistic setting. Still, it is a promising first result showing the potential of this technique.

6 Related Work

Waltz and Llinas call make the distinction between "low-level processing" and "high-level processing" [17] also referred to as 'low-level information fusion" (LLIF) and "high-level information fusion" (HLIF) respectively. While LLIF is about fusing information from sensors, HLIF deals with behaviour and intents.

For movement data probabilistic methods have successfully been applied to fuse data from sensors, e.g. [8]. It has also been shown that anomalies and possible behaviour can be recognised [9, 13] which supports HLIF. Despite the encouraging results obtained, this research is highly restricted to movement data reported from sensors, which limits their capability to detect or predict relevant events.

We further focus on work for non-movement attributes as discussed in this report. There are some results already showing the potential of applying first-order probabilistic formalisms to support surveillance tasks. While the way how those approaches tackle the problem of representing uncertainty in domain knowledge is very similar to out work, to our knowledge all work in this area is restricted to mostly HLIF tasks.

A system for situational awareness that explicitly represents uncertainty in a probabilistic manner is proposed in [1]. The reasoning mechanism is based on Multy-Entity Bayesian Networks (MEBN)–a first-order Bayesian logic formalism. A realistic scenario shows the flexible, distributed and probabilistic nature of the proposed approach. There is further work based on MEBNs or Markov logic to represent maritime domain knowledge and reason about vessel's intents [2, 4, 14, 6]. An interesting approach of how to reason about intentions, taking into account the behaviour of vessels over a period of time is presented in [7]. An example of similar work outside of the maritime domain is [15].

In contrast to this work, we also deal with LLIF of information about intrinsic properties, like names, and show that the synergy between LLIF and HLIF can improve the results of both. Furthermore, our work provides a general, domain-independent core model we believe to be applicable to a wide range of tasks and use the maritime domain only as example.

Finally, research has been done about how to represent and align information in complex and uncertain domains, by means of so called *ontologies*. Research in this field aims at extending existing ontology formalisms with uncertainty [5, 3]. Also the work already discussed above partially deals with that issue (e.g. [1, 2]). While we abstract from that this problem in our research and assume all information used is semantically aligned using a fixed set of attributes, ontology research lays the basis for being able to automatically reason about information.

7 Conclusions

We developed a framework to fuse information about intrinsic properties and intentions of objects under surveillance. The framework is based on a probabilistic logic. This representation allows us to deal with the dynamic amount of information in the domain and dynamic ranges of attributes which are unknown in advance. The formalism furthermore allows to represent knowledge in a way that makes relationships between entities and also the attached probabilistic knowledge locally interpretable.

We further show how our approach can be applied to build a model for maritime surveillance by adding domain specific knowledge to the general framework. To our knowledge this is one of the very few real-world applications of probabilistic logics in real-world settings. We finally experimentally show that our model can correct for errors in information transmitted by simulated vessels by fusing this information with information from other sources.

Acknowledgements

We thank the colleagues in the Metis project, and in particular Jan Laarhuis, for the fruitful discussions and valuable ideas on this research.

References

[1] R.N. Carvalho, P.C.G. Costa, K.B. Laskey, and K.C. Chang. PROGNOS: Predictive situational awareness with probabilistic ontologies. In *Proc. of the 13th Conference on Information Fusion*, pages 1–8, 2010.

[2] R.N. Carvalho, R. Haberlin, P.C.G. Costa, K.B. Laskey, and K.C. Chang. Modeling a probabilistic ontology for maritime domain awareness. In *Proceedings of the 14th International Conference on Information Fusion*, pages 1285–1292, 2011.

[3] Paulo Cesar, G. Costa, Kathryn B. Laskey, and Kenneth J. Laskey. Pr-owl: A bayesian ontology language for the semantic web. In *Center for Technology-Enhanced Learning, University of Missouri-Rolla*, 2003.

[4] Paulo C. G. Costa, Kathryn B. Laskey, Kuo-Chuand Chang, Weiand Sun, Cheol Y. Park, and Shou Matsumoto. High-level information fusion with bayesian semantics. In *Proceedings of the Nineth Bayesian Modelling Applications Workshop, held at the Conference of Uncertainty in Artificial Intelligence (BMAW UAI 2012)*, 2012.

[5] Zhongli Ding and Yun Peng. A probabilistic extension to ontology language owl. In *In Proceedings of the 37th Hawaii International Conference On System Sciences (HICSS-37), Big Island*, 2004.

[6] Pedro Domingos, Stanley Kok, Daniel Lowd, Hoifung Poon, Matthew Richardson, and Parag Singla. Markov logic. In *Probabilistic Inductive Logic Programming*, pages 92–117, 2008.

[7] Yvonne Fischer and Jrgen Beyerer. A top-down-view on intelligent surveillance systems. In *Proceedings of the Seventh International Conference on Systems*, pages 43–48, Saint Gilles, Reunion, February 2012.

[8] J. García, J.L. Guerrero, A. Luis, and J.M. Molina. Robust sensor fusion in real maritime surveillance scenarios. In *Proc. of the 13th Conference on Information Fusion*, 2010.

[9] M. Guerriero, P. Willett, S. Coraluppi, and C. Carthel. Radar/AIS data fusion and SAR tasking for maritime surveillance. In *Proc. of the 11th Conference on Information Fusion*, pages 1–5, 2008.

[10] Bernd Gutmann, Ingo Thon, Angelika Kimmig, Maurice Bruynooghe, and Luc De Raedt. The magic of logical inference in probabilistic programming. *CoRR*, abs/1107.5152, 2011.

[11] Dominik Jain, Bernhard Kirchlechner, and Michael Beetz. Extending Markov Logic to Model Probability Distributions in Relational Domains. In *KI 2007: Advances in Artificial Intelligence, 30th Annual German Conference on AI*, volume 4667 of *Lecture Notes in Computer Science*, pages 129–143. Springer, 2007.

[12] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

[13] M. Seibert, B.J. Rhodes, N.A. Bomberger, P.O. Beane, J.J. Sroka, W. Kogel, W. Kreamer, C. Stauffer, L. Kirschner, E. Chalom, M. Bosse, and R. Tillson. SeeCoast port surveillance. In *Proc. of SPIE*, volume 6204, 2006.

[14] L. Snidaro, I. Visentini, K. Bryan, and G.L. Foresti. Markov logic networks for context integration and situation assessment in maritime domain. In *Proceedings of the 14th International Conference on Information Fusion*, pages 1534–1539, 2012.

[15] Son D. Tran and Larry S. Davis. Event modeling and recognition using markov logic networks. In *Proceedings of the 10th European Conference on Computer Vision: Part II*, ECCV '08, pages 610–623, Berlin, Heidelberg, 2008. Springer-Verlag.

[16] Joost Vennekens, Sofie Verbaeten, and Maurice Bruynooghe. Logic programs with annotated disjunctions. In *In Proc. Intl Conf. on Logic Programming*, pages 431–445. Springer, 2004.

[17] Edward L. Waltz and James Llinas. *Multisensor Data Fusion*. Artech House, Inc., Norwood, MA, USA, 1990.