How not to drown in a sea of information: An event recognition approach

Elias Alevizos[1], Alexander Artikis[2,1], Kostas Patroumpas[3,4],
Marios Vodas[3], Yannis Theodoridis[3], Nikos Pelekis[5]
[1]*Institute of Informatics & Telecommunications, NCSR Demokritos, Athens, Greece*
[2]*Department of Maritime Studies, University of Piraeus, Greece*
[3]*Department of Informatics, University of Piraeus, Greece*
[4]*School of Electrical & Computer Engineering, National Technical University of Athens, Greece*
[5]*Department of Statistics & Insurance Science, University of Piraeus, Greece*
alevizos.elias@iit.demokritos.gr, {a.artikis, kpatro, mvodas, ytheod, npelekis}@unipi.gr

***Abstract*—Maritime monitoring is a typical Big Data problem where hundreds of thousands of vessels across the globe transmit messages about their location, speed and other information. We have developed a system for online vessel tracking that performs, as a first step, a high-rate but accurate trajectory compression. Subsequently, the compressed trajectories are analyzed by a complex event recognition engine, promptly reporting alerts to maritime authorities. To deal with realistic maritime event patterns, we seamlessly integrated spatial and temporal reasoning for online event recognition. The system is evaluated on real data from the Greek seas.**

***Keywords*-event processing; data streams; event calculus**

I. Introduction

Global trade is carried mainly by sea. Projections estimate that the current 8 billion tones of cargo transported at sea will have grown to 23 billion tones by 2060 [1]. The current and growing economic importance of maritime transport brings with it a number of challenges, in terms of ecological responsibility, financial costs, safety and security. As a first step towards addressing such challenges, the Automatic Identification System (AIS)[1] was introduced, in order to provide real-time information about the behavior of vessels. Today, there are more than 580,000 AIS-equipped vessels worldwide[2]. The main purpose of AIS is to help in avoiding dangerous situations, such as collisions, and in providing maritime authorities with the necessary information to monitor the activities of vessels. Both the volume of AIS messages, transmitted by hundreds of thousands of distributed sources, and their uncertainty/ambiguity, make the task of maritime monitoring a typical Big Data problem.

In previous work [2], we presented an AIS stream processing system for maritime surveillance. The focus of that work was a trajectory detection component that achieves a 95% compression of the AIS messages, keeping only the 'critical points' of each trajectory, i.e. those few points carrying the necessary information to reconstruct the initial trajectory. These points were fed to a complex event recognition engine that detected some simple forms of suspicious maritime activity. In order to meet the real-time requirements of data stream processing, this in-memory process made use of a *sliding window* [3], which abstracts the time period of interest and keeps up with the evolving movement. A window covers trajectory segments over a recent *range* (e.g. only positions received during the past hour). The window gets refreshed at a specific *slide step*, e.g. each minute, in order to keep in pace with newly arrived positions.

In the present paper, we present novel patterns of a set of realistic, complex maritime events, such as vessel rendezvous and package picking. Moreover, to deal with the necessary spatial tasks efficiently, such as determining whether a vessel/point is located within a protected area/polygon, we introduce a grid partitioning of the surveillance area, and integrate seamlessly spatial reasoning with temporal reasoning for online complex event recognition. Finally, to evaluate our system, we combine real AIS messages with real geographical data about ports and areas of interest, such as NATURA areas[3]. The surveillance area—the Greek seas—includes approximately 4K polygons representing the areas of interest, with a total of 78K edges. We show that the system performs in real-time on data ranging from June to August 2009 including 168M AIS messages that are compressed to 16M critical positions.

II. Critical Points along Vessel Trajectories

Vessels usually move along straight courses at open sea with stready speed over long intervals. Thus, it suffices to detect *"critical points"* such as stops, accelaration or turning points along their trajectory. Such *movement events* (ME) can readily offer a concise, approximate, yet quite reliable representation of vessel traces. In this Section, we overview such a tracking operation as introduced in [2], performed *online* against numerous streaming positions without harming the quality of the resulting trajectory representations.

[1]http://www.imo.org/OurWork/Safety/Navigation/Pages/AIS.aspx
[2]https://www.vesselfinder.com
[3]http://ec.europa.eu/environment/nature/natura2000/

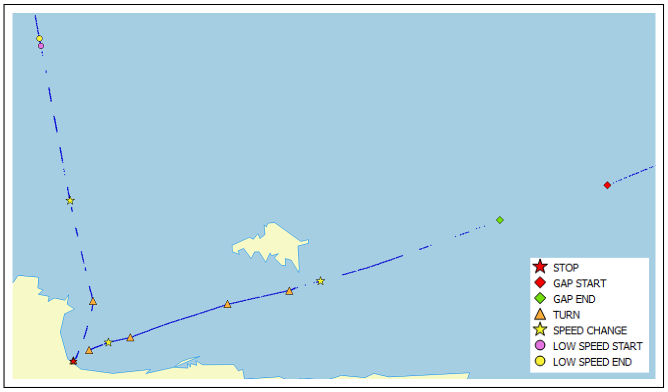


Figure 1: Critical points identified along a vessel trajectory.

Monitoring the continuously incoming AIS messages from a large fleet of N vessels, effectively yields an *append-only stream* of timestamped locations $\langle MMSI, Lon, Lat, \tau \rangle$. Each input tuple includes the $MMSI$ (Maritime Mobile Service Identity) of the reporting vessel and its 2-d geographical coordinates (Lon, Lat) at a discrete timestamp τ (e.g., at the granularity of seconds). The evolving sequence of such positional samples represents the *trajectory* of each vessel.

This online tracking process can detect various types of trajectory events, either *instantaneous* (e.g., a sudden turn) or of *longer duration* (e.g., a smooth turn), and accordingly emit critical points at each window slide:

- *Long-term stop* occurs when at least m (e.g., m=10) consecutive positions are found within a small radius r (e.g., r=200 meters), and thus indicate immobility. So, these locations are collectively approximated by a single critical point marked as *stopped* with its duration.
- *Communication gaps* occur when a vessel has not emitted a message over a period ΔT, e.g., the past 10 minutes. A pair of critical points signify when contact was lost (*gapStart*) and when it was restored (*gapEnd*).
- An instantaneous turn occurs when the heading of a vessel has changed by more than a given angle $\Delta\theta$, e.g., 15^o. Yet, this may be coincidental, because a series of such nearby events may also signify a stop (e.g., due to sea drift). If this is not the case, then a critical point (*turn*) is emitted. But normally, vessels take smooth turns due to their large size and maritime regulations. By checking whether the cumulative change in heading over the sequence of previous positions exceeds $\Delta\theta$, a series of such *turn* points may be emitted.
- A critical point for *speedChange* is issued once the rate of change for speed exceeds a given threshold a (e.g., a = 25%). This is typical when a ship slows down approaching to a port or speeds up departing from it.
- *Slow motion* means that the vessel moves along a route at low speed (e.g., $<$ 1 knot) over at least m (e.g., m=10) of its most recent messages. The first and the last of these positions are both reported as critical, respectively marked as *lowSpeedStart* and *lowSpeedEnd*.

As empirically verified in [2], this online filtering achieves data compression close to 95%, incurring negligible loss in approximation accuracy. With such dramatic reduction in system load, subsequent stages of analysis can be greatly improved, e.g., reducing latency of online collision detection or similarity search among recent vessel paths. The example trajectory in Figure 1 illustrates the data compression gains achieved when retaining critical points only.

III. Complex Event Recognition

The derived critical points, or Movement Events (ME), are transmitted to the complex event (CE) recognition module, which combines the ME stream with static geographical data, such as the locations of ports and protected areas. The objective of this process is to recognize potentially suspicious or dangerous maritime situations. When recognized, such CE are forwarded to the marine authorities for real-time decision making.

A. Event Calculus for Run-Time reasoning

Our CE recognition component is based on the 'Event Calculus for Run-Time reasoning' (RTEC) [4]. The Event Calculus [5] is a logic programming language for representing and reasoning about events and their effects. We chose RTEC for maritime monitoring because it has a formal, declarative semantics, supports atemporal reasoning and reasoning over background knowledge, has built-in axioms for complex temporal phenomena, explicitly represents CE intervals and thus avoids the related logical problems [6], and supports out-of-order data streams. In this Section, we present RTEC following [4] and illustrate its use for maritime surveillance.

The time model of RTEC is linear and includes integer time-points. Variables start with an upper-case letter, while predicates and constants start with a lower-case letter. Where F is a *fluent*—a property that is allowed to have different values at different points in time—the term $F=V$ denotes that fluent F has value V. $\mathsf{holdsAt}(F=V, T)$ represents that fluent F has value V at a particular time-point T. $\mathsf{holdsFor}(F=V, I)$ represents that I is the list of the maximal intervals for which $F=V$ holds continuously. holdsAt and holdsFor are defined in such a way that, for any fluent F, $\mathsf{holdsAt}(F=V, T)$ if and only if T belongs to one of the maximal intervals of I for which $\mathsf{holdsFor}(F=V, I)$.

The happensAt predicate represents an instance of an event type. E.g. $\mathsf{happensAt}(speedChange(vessel_1), 5)$ represents the occurrence of event type $speedChange(vessel_1)$ at time 5. An *event description* in RTEC includes rules that define the event instances with the use of the happensAt predicate, the effects of events with the use of the initiatedAt and terminatedAt predicates, and the values of the fluents with the use of the holdsAt and holdsFor predicates, as well as other, possibly

Table I: RTEC Predicates.

Predicate	Meaning
holdsAt$(F = V,\ T)$	The value of fluent F is V at time T
holdsFor$(F = V,\ I)$	I is the list of the maximal intervals for which $F = V$ holds continuously
happensAt$(E,\ T)$	Event E occurs at time T
initiatedAt$(F = V,\ T)$	At time T a period of time for which $F = V$ is initiated
terminatedAt$(F = V,\ T)$	At time T a period of time for which $F = V$ is terminated
intersect_all(L, I)	I is the list of maximal intervals produced by the intersection of the lists of maximal intervals of list L

atemporal, constraints. Table I presents a fragment of the predicates available to the event description developer.

For a fluent F, $F = V$ holds at a particular time-point T if $F = V$ has been *initiated* by an event that has occurred at some time-point earlier than T, and has not been *terminated* at some other time-point in the meantime. This is an implementation of the *law of inertia*. To compute the *intervals* I for which $F = V$, that is, holdsFor$(F = V, I)$, we find all time-points T_s at which $F = V$ is initiated, and then, for each T_s, we compute the first time-point T_f after T_s at which $F = V$ is terminated. Consider the following example:

$$\begin{aligned}&\textsf{initiatedAt}(gap(Vessel) = \textsf{true},\ T) \leftarrow\\&\quad \textsf{happensAt}(gapStart(Vessel),\ T),\\&\quad \textsf{holdsAt}(coord(Vessel) = (Lon, Lat),\ T),\\&\quad \textsf{not}\ nearPorts(Lon, Lat)\end{aligned} \tag{1}$$

$$\begin{aligned}&\textsf{terminatedAt}(gap(Vessel) = \textsf{true},\ T) \leftarrow\\&\quad \textsf{happensAt}(gapEnd(Vessel),\ T)\end{aligned} \tag{2}$$

$gap(Vessel)$ is a Boolean fluent denoting a communication gap concerning $Vessel$, i.e. the $Vessel$ stops sending AIS messages. Under certain circumstances, the absence of AIS messages may be considered suspicious and therefore we need to record it. $gapStart(Vessel)$ and $gapEnd(Vessel)$ are instantaneous MEs indicating, respectively, the time-points in which a $Vessel$ stops and resumes sending AIS messages. MEs are detected by the trajectory detection module (see Section II). $coord$ is a fluent reporting the coordinates of a vessel. This type of information is also provided by the trajectory detection module. $nearPorts(Lon, Lat)$ is an atemporal predicate that becomes true when the point (Lon, Lat) is close to a port. 'not' is negation-by-failure. Rule (1) states that $gap(Vessel) = \textsf{true}$ is initiated if the trajectory detection module reports a $gapStart$ ME for the $Vessel$, and the $Vessel$ is far from the ports of the area under surveillance. In other words, we discard communication gaps when vessels are in ports. Rule (2) states $gap(Vessel) = \textsf{true}$ is terminated when the $Vessel$ resumes communications. Given rules (1) and (2), RTEC computes the list of maximal intervals during which $gap(Vessel) = \textsf{true}$ holds continuously.

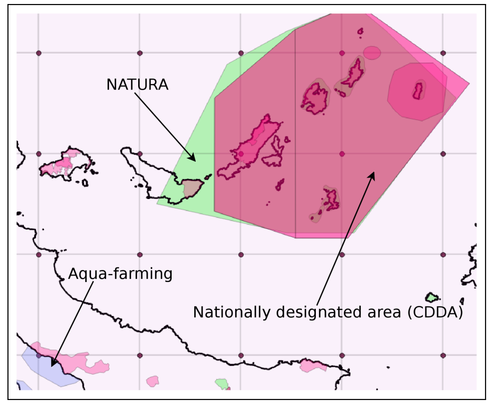


Figure 2: Grid view of the map: different types of polygon indicate different types of (overlapping) area of interest.

B. Spatial Indexing

CE recognition for maritime surveillance requires various types of spatial task. E.g. we need to determine whether a point—a ship's location—lies inside a polygon indicating an area of interest, such as a NATURA area, and whether it is near another point, such as a port. Moreover, we need to detect the vessels that are in close proximity (heading towards each other). For efficient spatial reasoning, we employed a grid partitioning scheme which divides the surveillance area into equal sized cells (see Figure 2). Each area of interest and port is assigned only to those cells with which it overlaps. This assignment is performed off-line and provided as background knowledge to the CE recognition module. The use of a grid allows us to quickly determine, through a simple calculation on the coordinates, the cell inside which a vessel is located (the task of determining each vessel's cell is performed before each CE recognition step). This way, we can efficiently calculate the number of vessels in close proximity and check whether a vessel is inside an area of interest, by performing calculations (e.g. using the ray crossings algorithm [7] for determining whether a point lies inside a polygon) *only* for those vessels/areas in the same or immediately surrounding cells.

C. Maritime Surveillance Scenarios

Given the critical ME stream produced by the trajectory detection module, and a set of areas of interest, RTEC recognizes a set of CEs for the benefit of maritime authorities. The choice of CEs and their patterns were specified in collaboration with the domain experts of the AMINESS project. Below we present a fragment of our CE patterns.

Illegal Shipping. A common feature of illegal shipping is communication gap; certain vessels, such as those passing through protected areas in order to minimize trip length and fuel consumption, switch off their transmitters and stop sending position signals. In such cases, it is often claimed that the transmitter temporarily broke down. To capture this type of activity, we defined the CE below:

$$\begin{aligned}&\mathsf{happensAt}(\mathit{illegalShipping}(\mathit{Vessel}),\ T) \leftarrow\\&\quad \mathsf{happensAt}(\mathsf{start}(\mathit{gap}(\mathit{Vessel})=\mathsf{true}),\ T),\\&\quad \mathsf{holdsAt}(\mathit{coord}(\mathit{Vessel})=(\mathit{Lon},\mathit{Lat}),\ T),\\&\quad \mathit{inArea}(\mathit{Lon},\mathit{Lat})\end{aligned} \tag{3}$$

illegalShipping is an instantaneous CE. $\mathsf{start}(F=V)$ (respectively $\mathsf{end}(F=V)$) is a built-in RTEC event taking place at each starting (ending) point of each maximal interval for which $F=V$ holds continuously. Thus $\mathsf{start}(\mathit{gap}(\mathit{Vessel})=\mathsf{true})$ takes place at the starting point of each maximal interval for which there is a communication gap concerning the *Vessel* at open sea (see rules (1) and (2) for the *gap* fluent). Recall that $\mathit{coord}(\mathit{Vessel})$ is a fluent indicating the location of a *Vessel*. $\mathit{inArea}(\mathit{Lon},\mathit{Lat})$ is an atemporal predicate that succeeds when the point $(\mathit{Lon},\mathit{Lat})$ lies inside at least one of the protected areas (such areas may be overlapping). Rule (3) states that 'illegal shipping' is recognized when a vessel stops sending AIS messages while sailing through a protected area.

Rule (3) is but one possible formulation of 'illegal shipping'. In some applications, we may want to fire rule (3) when a vessel is *close* to a protected area, as opposed to sailing *inside* the area. Moreover, we may want to extend the formulation of 'illegal shipping' in order to take into consideration the possibility of sailing through a protected area without necessarily switching off the AIS transmitter. To deal with this case, we can replace $\mathsf{start}(gap(Vessel)=\mathsf{true})$ in the first condition of rule (3) with an auxiliary event aux_e defined e.g. as follows:

$$\begin{aligned}&\mathsf{happensAt}(\mathit{aux\_e}(\mathit{Vessel}),\ T) \leftarrow\\&\quad \mathsf{happensAt}(\mathit{turn}(\mathit{Vessel}),\ T)\\&\mathsf{happensAt}(\mathit{aux\_e}(\mathit{Vessel}),\ T) \leftarrow\\&\quad \mathsf{happensAt}(\mathsf{start}(\mathit{lowSpeed}(\mathit{Vessel})=\mathsf{true}),\ T)\\&\mathsf{happensAt}(\mathit{aux\_e}(\mathit{Vessel}),\ T) \leftarrow\\&\quad \mathsf{happensAt}(\mathsf{start}(\mathit{gap}(\mathit{Vessel})=\mathsf{true}),\ T)\end{aligned} \tag{4}$$

turn is an instantaneous ME. *lowSpeed* is a fluent defined by the *lowSpeedStart* and *lowSpeedEnd* MEs—the definition of *lowSpeed* is similar to the definition of *gap* and therefore omitted for brevity. According to rule-set (4), an $\mathit{aux\_e}(\mathit{Vessel})$ takes place when the *Vessel* turns, starts moving slowly or stops sending AIS signals at open sea.

Instead of defining 'illegal shipping' as an instantaneous CE, we could have specified a durative CE in terms of $\mathsf{initiatedAt}$ and $\mathsf{terminatedAt}$ rules. In this case, the conditions of rule (3) would be the conditions of an $\mathsf{initiatedAt}$ rule. 'Illegal shipping' would then be terminated when an ME—*gapEnd*, *turn*, etc—is detected outside a protected area. However, since MEs are *critical* points, it is likely that an ME may be detected well after the vessel has passed through the protected area, thus leading to a (much) delayed termination of 'illegal shipping'. In other words, in the absence of *all* trajectory points, we sometimes have to refrain from defining durative CE in order to avoid false positives.

Suspicious Vessel Delay. Sailing through a protected area is not the only reason for switching off an AIS transmitter. To investigate further the behavior of vessels during a communication gap, we formulated the CE below:

$$\begin{aligned}&\mathsf{holdsFor}(\mathit{suspiciousDelay}(\mathit{Vessel})=\mathsf{true},\ I) \leftarrow\\&\quad \mathsf{holdsFor}(\mathit{gap}(\mathit{Vessel})=\mathsf{true},\ I_{gap}),\\&\quad \mathit{extendedDelays}(\mathit{Vessel},\ I_{gap},\ I)\end{aligned} \tag{5}$$

Recall that I in $\mathsf{holdsFor}(F=V,I)$ is the list of the maximal intervals for which $F=V$ holds continuously (see Table I). I_{gap} in $\mathsf{holdsFor}(\mathit{gap}(\mathit{Vessel})=\mathsf{true},\ I_{gap})$, therefore, is the list of maximal intervals during which a *Vessel* stops sending AIS signals while at open sea. $\mathit{extendedDelays}(\mathit{Vessel},I',I)$ is a predicate which returns the maximal intervals I of the list I' for which the highest possible speed of the *Vessel* is below a threshold. We estimate the highest possible speed of a vessel in a simplified way: we assume that the vessel moved along a straight line from the point of *gapStart* to that of *gapEnd*. Under this assumption, its speed cannot have been greater than the one determined by dividing this shortest path by the time spent to travel it. Rule (5) thus states that a very low vessel speed combined with a communication gap occurring at open sea is to be treated as a suspicious delay.

A more refined implementation would estimate the highest possible speed of a vessel during a communication gap even when it is impossible (due to e.g. terrestrial areas) or unlikely (due to weather conditions) to sail along a straight line.

Vessel Rendezvous. *suspiciousDelay* allows us to define further types of activity—consider the rule below:

$$\begin{aligned}&\mathsf{holdsFor}(\mathit{possibleRendezvous}(\mathit{Vessel_1},\mathit{Vessel_2})=\mathsf{true},\ I) \leftarrow\\&\quad \mathsf{holdsFor}(\mathit{in}(\mathit{Vessel_1})=\mathit{Cell},\ I_1),\\&\quad \mathsf{holdsFor}(\mathit{in}(\mathit{Vessel_2})=\mathit{Cell},\ I_2),\\&\quad \mathsf{holdsFor}(\mathit{suspiciousDelay}(\mathit{Vessel_1})=\mathsf{true},\ I_3),\\&\quad \mathsf{holdsFor}(\mathit{suspiciousDelay}(\mathit{Vessel_2})=\mathsf{true},\ I_4),\\&\quad \mathsf{intersect\_all}([I_1,I_2,I_3,I_4],\ I)\end{aligned} \tag{6}$$

$\mathit{in}(\mathit{Vessel})=\mathit{Cell}$ indicates the *Cell* of the grid in which the *Vessel* is located. The value of this fluent is set prior to each CE recognition step, as described in Section III-B. $\mathsf{intersect\_all}$ is a built-in RTEC predicate which calculates the intersection of a list of lists of maximal intervals (see Table I). According to rule (6), if two vessels simultaneously exhibit a *suspiciousDelay* and are located in the same cell, then this could indicate that they had arranged for a rendezvous. Note that, since we do not have information about the vessels' locations during gaps, the above rule cannot capture the precise time of the rendezvous, if any.

Suspicious Area. Under normal circumstances, vessels stop in ports or other designated areas, typically located close to ports. The CE defined below deals with the case where vessels stop in non-designated areas:

$$\begin{aligned}&\mathsf{happensAt}(suspicious(Cell),\ T) \leftarrow\\&\quad \mathsf{happensAt}(\mathsf{start}(stopped(Vessel)=\mathsf{true}),\ T),\\&\quad \mathsf{holdsAt}(in(Vessel)=Cell,\ T),\\&\quad \mathsf{holdsAt}(coord(Vessel)=(Lon,Lat),\ T),\\&\quad \mathsf{holdsAt}(nearbyStoppedVessels(Lon,Lat)=N,\ T),\\&\quad N \geq 2\end{aligned} \tag{7}$$

$suspicious(Cell)$ is an instantaneous CE indicating that an area overlapping a $Cell$ of the grid might be suspicious. $stopped(Vessel)$ is a Boolean fluent indicating that a $Vessel$ has stopped at open sea. The definition of this fluent is based on the information provided by the trajectory detection module. This module reports the list of maximal intervals during which a vessel has stopped. From this list, we keep only those intervals where the vessel is not in any port. Recall that $\mathsf{start}(F=V)$ is a built-in RTEC event taking place at each starting point of each maximal interval for which $F=V$ holds continuously. $nearbyStoppedVessels(Lon,Lat)=N$ calculates the number of stopped vessels N that are close to (Lon,Lat). According to rule (7), the area of a $Cell$ is said to be suspicious when a $Vessel$ stops in the $Cell$ and there are at least two other stopped vessels in close proximity. (The threshold of two stopped vessels was chosen in collaboration with domain experts.) The recognition of $suspicious(Cell)$ may indicate a vessel located in the $Cell$, or neighboring cells, is facing problems and other vessels have stopped to help.

Fast Approach. Another dangerous situation may arise when a vessel is rapidly moving towards some other vessel(s). Such a behavior could indicate a vessel pursuit or even imminent collision. Consider the formalization below:

$$\begin{aligned}&\mathsf{happensAt}(fastApproach(Vessel),\ T) \leftarrow\\&\quad \mathsf{happensAt}(speedChange(Vessel),\ T),\\&\quad \mathsf{holdsAt}(velocity(Vessel)=Speed,\ T),\\&\quad Speed > 20\ knots,\\&\quad \mathsf{holdsAt}(coord(Vessel)=(Lon,Lat),\ T),\\&\quad \mathsf{not}\ nearPorts(Lon,Lat),\\&\quad \mathsf{holdsAt}(headingToVessels(Vessel)=\mathsf{true},\ T)\end{aligned} \tag{8}$$

$fastApproach(Vessel)$ and $speedChange(Vessel)$ are instantaneous CE and ME respectively. $velocity$ is a fluent indicating the speed of a vessel. Similar to the coordinates of a vessel, this information, as well as a vessel's heading, is provided by the trajectory detection module and accompanies every detected ME. $headingToVessels(Vessel)$ is a fluent that becomes true whenever a $Vessel$'s direction of movement is towards at least one other vessel. According to rule (8), a 'fast approach' movement is recognized when a $Vessel$ changes its speed at open sea, the new speed is above 20 knots, and there is at least one other nearby vessel towards which it is heading. The value of 20 knots, like all other numerical thresholds, was chosen by domain experts.

In the case of a gradual speed change, the trajectory detection module would not emit a $speedChange$ ME. Consequently, RTEC would not be able to recognize 'fast approach'. This is the side-effect of the trajectory compression. The benefits of this compression, however, are substantial with respect to the efficiency of the system, as demonstrated by our empirical evaluation (see Section IV).

Package Picking. Another possible interaction between two vessels is when one of them drops a package at some area and another vessel appears later in order to pick the package. One possible way of formulating this type of interaction is the following:

$$\begin{aligned}&\mathsf{happensAt}(possiblePicking(Vessel_1,Vessel_2),\ T_{pick}) \leftarrow\\&\quad \mathsf{happensAt}(\mathsf{end}(stopped(Vessel_1)=\mathsf{true}),T_{drop}),\\&\quad \mathsf{holdsAt}(in(Vessel_1)=Cell,\ T_{drop}),\\&\quad \mathsf{happensAt}(\mathsf{start}(stopped(Vessel_2)=\mathsf{true}),T_{pick}),\\&\quad \mathsf{holdsAt}(in(Vessel_2)=Cell,\ T_{pick}),\\&\quad T_{pick}-T_{drop} < 1\ hour,\\&\quad \mathsf{holdsAt}(coord(Vessel_1)=(Lon_1,Lat_1),\ T_{drop}),\\&\quad \mathsf{holdsAt}(coord(Vessel_2)=(Lon_2,Lat_2),\ T_{pick}),\\&\quad distance((Lon_1,Lat_1),(Lon_2,Lat_2),\ Dist),\\&\quad Dist < 0.5\ miles\end{aligned} \tag{9}$$

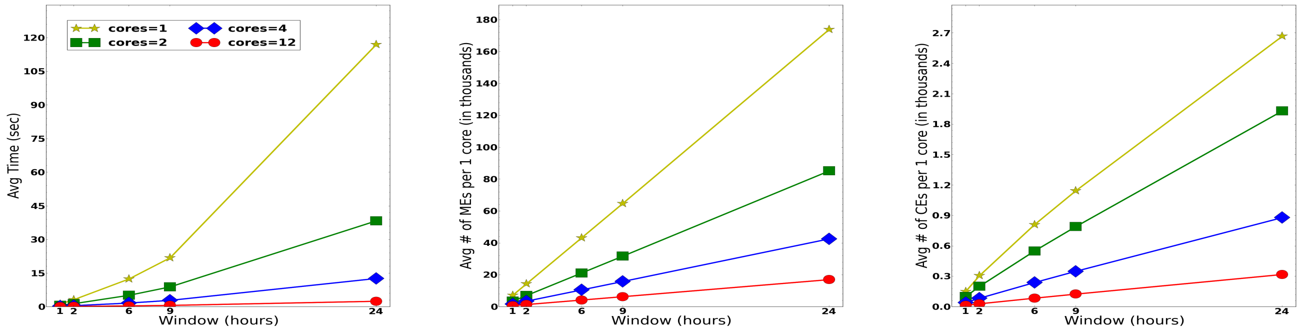
Rule (9) describes a scenario where a vessel had stopped inside a $Cell$ and started moving at time T_{drop}, then, after no more than an hour, a second vessel arrived and stopped inside the same $Cell$, and the Haversine distance between the two stop locations, as calculated by the $distance$ predicate, was no more than half a mile.

IV. Empirical Evaluation

RTEC[4], the CE recognition component, is implemented in Prolog. The experiments presented below were run on a computer with Intel Xeon CPU E5-2630 v2@2.60GHz×12 processors and 256GB RAM, running Ubuntu Linux 14.04 and SWI Prolog 7.2.1. Our empirical evaluation was based on a real AIS dataset obtained from IMIS Hellas[5], our partner in the AMINESS project. Raw data is 23GB in size and spans from 1 June 2009 to 31 August 2009 for 6,425 vessels sailing through the Greek seas: the Aegean, the Ionian, and part of the Mediterranean Sea. When decoded and cleaned from corrupt messages, the dataset yielded 168,240,595 timestamped positions[6].

The trajectory detection module compresses the position stream to a stream of critical movement events (ME)s. The input of RTEC consists of the instantaneous MEs $gapStart$ and $gapEnd$, indicating communication gaps, $lowSpeedStart$, $lowSpeedEnd$, $speedChange$ and $turn$, and the durative $stopped$ ME. Each such event is accompanied by the coordinates and velocity (speed and heading) of the

[4] https://github.com/aartikis/RTEC
[5] http://www.imishellas.gr/
[6] http://chorochronos.datastories.org/?q=content/imis-3months



(a) Average CE recognition times.

(b) Average number of MEs.

(c) Average number of CEs.

Figure 3: CE recognition for approximately 6,5K vessels, 4K areas with 78K edges, and 16M MEs.

corresponding vessel at the time of ME detection. Note that, since the input stream consists of *critical* MEs, most of them activate CE pattern matching. This is in contrast to other empirical analyses found in the literature (including [4]) where the input stream includes several events that do not affect CE recognition. In addition to this event stream, RTEC makes use of real static data consisting of areas of interest, such as NATURA and aqua-farming areas, represented as polygons, and ports, represented as points, across the Greek seas. Given this combination of event stream and static geographical information, RTEC recognizes the CEs described in the previous section: illegal shipping, suspicious vessel delay, vessel rendezvous, suspicious areas, vessel pursuit, and package picking. In total, the dataset has 15,884,253 MEs, 3,966 areas of interest with a total of 78,418 edges, and 64 ports. The size of the grid is 720×900 km^2.

We simulated a streaming behavior by consuming the MEs little by little, i.e. reading small chunks periodically according to window specifications. We examine sliding windows of varying sizes. Next, we report indicative results from these experiments.

Experimental Results

Figure 3 shows our empirical evaluation. First, we used a single processor to perform CE recognition for all 6,425 vessels, 3,966 areas and 64 ports. We subsequently employed multiple processors on which RTEC operated in parallel, by following a data partitioning scheme. We divided the grid covering the surveillance area into multiple sub-grids (groups of adjacent cells) whose number was equal to that of the processors used in parallel. Each processor was responsible for the areas and ports located in, and the vessels passing through its assigned sub-grid. We used three distributed settings: running CE recognition on two, four and twelve processors. We made an attempt to evenly distribute the load of MEs among the different processors, by exhaustively searching for the best configuration. The constraint was that the different sub-grids were required to be aligned parallelograms with right angles. As a result, we did not take into account solutions with sub-grids of arbitrary shapes and the load distribution was thus not the best possible. This is an off-line process that takes place only once.

Figure 3a shows the average CE recognition times in CPU seconds, including the time taken for spatial indexing (see Section III-B). The slide is 1 hour while the window ranges from 1 hour to 24 hours. Figure 3b shows the average number of MEs for each setting. In the case of a single processor, the window ranges from ≈7,200 MEs (1 hour) to 175,000 MEs (24 hours). In the distributed settings—two, four and twelve processors for CE recognition—the input MEs are forwarded to the appropriate processor according to vessel location. E.g. when twelve processors are used in parallel, each one of them processes ≈700 MEs for the 1 hour window, and 17,000 MEs for the 24 hour window. Figure 3c shows the average number of CEs for each setting. This number also depends on the window size. E.g. in the case of a single processor, for 1 hour windows approximately 150 CEs are recognized, while for 24 hour windows RTEC recognizes around 2,900 CEs.

We do not show memory consumption figures because memory usage is stable. E.g. when using a single processor, memory consumption is approximately 2.0 GB for 1 hour windows and around 2.06 GB for 24 hour windows.

Figure 3a shows that we can achieve a significant performance gain by running RTEC in parallel. As the window size increases, the gain becomes more pronounced. Moreover, Figure 3a shows that RTEC is capable of supporting real-time CE recognition. E.g. for a window of 6 hours, RTEC recognizes all CEs requested by end users in 14 sec when a single processor is used, and in 0.4 sec when twelve processors are used in parallel.

V. Discussion

Very few approaches have applied CE recognition techniques to maritime surveillance. In a recent effort [8], maritime patterns were implemented in Esper[7], using its Event Processing Language. All of the trajectory averaging, smoothing and filtering operations are performed by Esper. In contrast, we have opted for a separation of concerns. We delegate trajectory compression to a separate, special-purpose module, and perform spatio-temporal reasoning on the compressed trajectories with the use of the CE recognition engine.

Our empirical evaluation combined real AIS messages with real geographical data about ports and areas of interest, such as NATURA and aqua-farming areas. The surveillance area includes approximately 4K polygons with 78K edges representing the areas of interest. We showed that the system performs in real-time using data including approximately 6,5K vessels and 16M critical positions.

Maritime surveillance requires the representation of complex phenomena [9]. We presented novel patterns of a set of realistic, composite maritime events, such as illegal shipping, suspicious vessel delay, vessel rendezvous, suspicious area, vessel pursuit and package picking. Maritime activities form hierarchies, in the sense that the formulation of one activity is also used to define other, higher-level activities. Recall e.g. the specification of vessel rendezvous in terms of suspicious vessel delay. In contrast to many state-of-the-art CE recognition systems, RTEC can naturally express hierarchical knowledge by means of well-structured specifications. E.g. it has been shown that RTEC outperforms significantly Esper in hierarchical domains [10].

Maritime surveillance also requires a combination of temporal reasoning with spatial reasoning. To deal with the necessary spatial tasks efficiently, such as determining whether a vessel (point) is located within a protected area (polygon), we introduced a grid partitioning of the surveillance area, and integrated seamlessly spatial reasoning with temporal reasoning for online CE recognition. Note that most CE recognition systems offer very limited capabilities for atemporal reasoning [11, 12, 13], restricting their applicability to maritime surveillance.

The construction of CE patterns posed a challenge in the development of the maritime surveillance system. Although the domain experts of the AMINESS project had initially some idea about the CE of interest, the patterns of these CE were not always clear. To facilitate the process of CE pattern construction, we plan to employ a recent framework for incremental structure learning that takes advantage of Big Data in order to construct Event Calculus programs [14]. The learning framework may also be used for code maintenance since it additionally supports pattern refinement.

[7]http://www.espertech.com/esper/

Acknowledgments

This work was carried out in the framework of the project *"AMINESS: Analysis of Marine INformation for Environmentally Safe Shipping"* co-financed by the European Fund for Regional Development and from Greek National funds through the operational programs "Competitiveness and Enterpreneurship" and "Regions in Transition" of the National Strategic Reference Framework - Action: "COOPERATION 2011 – Partnerships of Production and Research Institutions in Focused Research and Technology Sectors".

References

[1] "International shipping facts and figures—information resources on trade, safety, security, environment," International Maritime Organization, 2012.

[2] K. Patroumpas, A. Artikis, N. Katzouris, M. Vodas, Y. Theodoridis, and N. Pelekis, "Event recognition for maritime surveillance," in *EDBT*, 2015, pp. 629–640.

[3] K. Patroumpas and T. Sellis, "Maintaining consistent results of continuous queries under diverse window specifications," *Information Systems*, vol. 36, no. 1, pp. 42–61, 2011.

[4] A. Artikis, M. Sergot, and G. Paliouras, "An event calculus for event recognition," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 4, 2015.

[5] R. Kowalski and M. Sergot, "A logic-based calculus of events," *New Generation Computing*, vol. 4, no. 1, 1986.

[6] A. Paschke and M. Bichler, "Knowledge representation concepts for automated SLA management," *Decision Support Systems*, vol. 46, no. 1, pp. 187–205, 2008.

[7] J. O'Rourke, *Computational Geometry in C*. Cambridge University Press, Oct. 1998.

[8] F. Terroso-Saenz, M. Valdes-Vela, and A. F. Skarmeta-Gomez, "A complex event processing approach to detect abnormal behaviours in the marine environment," *Information Systems Frontiers*, pp. 1–16, 2015.

[9] J. van Laere and M. Nilsson, "Evaluation of a workshop to capture knowledge from subject matter experts in maritime surveillance," in *FUSION*, 2009, pp. 171–178.

[10] E. Alevizos and A. Artikis, "Being logical or going with the flow? A comparison of complex event processing systems," in *SETN*, ser. LNCS, vol. 8445. Springer, 2014, pp. 460–474.

[11] C. Dousson and P. L. Maigat, "Chronicle recognition improvement using temporal focusing and hierarchisation," in *IJCAI*, 2007, pp. 324–329.

[12] G. Cugola and A. Margara, "TESLA: a formally defined event specification language," in *DEBS*, 2010, pp. 50–61.

[13] H. Zhang, Y. Diao, and N. Immerman, "On complexity and optimization of expensive queries in complex event processing," in *SIGMOD*, 2014, pp. 217–228.

[14] N. Katzouris, A. Artikis, and G. Paliouras, "Incremental learning of event definitions with inductive logic programming," *Machine Learning*, vol. 100, no. 2–3, pp. 555–585, 2015.