EUCISE-OWL: An Ontology-based Representation of the Common Information Sharing Environment (CISE) for the Maritime Domain

Marina Riga[a], Efstratios Kontopoulos[a], Konstantinos Ioannidis[a,*], Spyridon Kintzios[b], Stefanos Vrochidis[a] and Ioannis Kompatsiaris[a]

[a] *Information Technologies Institute, Centre for Research and Technology Hellas, Thessaloniki, Greece*

[b] *Hellenic Ministry of National Defence, Athens, Greece*

Editor(s): Stefano Borgo, Consiglio Nazionale delle Ricerche (CNR), Italy

Solicited review(s): Guohui Xiao, Free University of Bozen-Bolzano, Italy; Two anonymous reviewers

Abstract. The timely and efficient cooperation across sectors and borders during maritime crises is paramount for the safety of human lives. Maritime monitoring authorities are now realizing the grave importance of cross-sector and cross-border information sharing. However, this cooperation is compromised by the diversity of existing systems and the vast volumes of heterogeneous data generated and exchanged during maritime operations. In order to address these challenges, the EU has been driving several initiatives, including several EU-funded projects, for facilitating information exchange across sectors and borders. A key outcome from these efforts is the Common Information Sharing Environment (CISE), which constitutes a collaborative initiative for promoting automated information sharing between maritime monitoring authorities. However, the adoption of CISE is substantially limited by its existing serialization as an XML Schema only, which facilitates information sharing and exchange to some extent, but fails to deliver the fundamental additional benefits provided by ontologies, like the richer semantics, enhanced semantic interoperability and semantic reasoning capabilities. Thus, this paper presents EUCISE-OWL, an ontology representation of the CISE data model that capitalizes on the benefits provided by ontologies and aims to encourage the adoption of CISE. EUCISE-OWL is an outcome from close collaboration in an EU-funded project with domain experts with extensive experience in deploying CISE in practice. The paper also presents a representative example for handling information exchange during a maritime crisis as well as performance results for specific querying tasks that can demonstrate and evaluate the use of the proposed ontology in practice.

Keywords: Maritime Monitoring, CISE, EUCISE2020, Data Model, Ontology

1. Introduction

During maritime crises, human lives are constantly at stake, while the time to react to unforeseen events is extremely limited. Therefore, cooperation across sectors, and often across borders, is valuable in order to ensure the safety and efficiency of operations. Relevant studies indicate that authorities are indeed starting to realize the importance of cross-sector and cross-border information sharing [28].

Nevertheless, on a practical level, this cooperation is compromised by the vast diversity of systems that operate simultaneously but are not yet adequately interconnected. On top of that, one should also add the vast volumes of heterogeneous data generated during maritime operations, including sensor measurements, intelligence, and reporting, amongst others.

*Corresponding author. E-mail: kioannid@iti.gr

In order to address the above challenges, the EU launched in 2005 several initiatives for improving the interoperability between national authorities' systems. These efforts included published communications, roadmaps, and green and blue papers, and eventually resulted in EU's Integrated Maritime Policy (IMP) [6]. In parallel, there have been several EU-funded projects aimed at fostering information exchange across sectors and borders [27], with the participation of many EU maritime monitoring authorities.

A key outcome from these EU-wide efforts is the *Common Information Sharing Environment* (*CISE*), representing an open, collaborative process within the EU for promoting automatic information sharing between authorities involved in maritime monitoring, across sectors and borders [5]. In order to support the authorities' continuously increasing needs, in conjunction with the constant decrease of operational personnel (operators), CISE aims to (a) increase the efficiency, quality, and responsiveness of surveillance and operations at sea, and, (b) ensure a safer, more secure, and environmentally protected EU maritime domain. The benefits of deploying a uniform model like CISE for maritime monitoring include: (a) minimizing the risk of human errors; (b) establishing a standard detection threshold, which can be dynamically adapted each time according to the needs and the occurring incidents; (c) expanding the human cognitive area; (d) reducing the need for highly experienced and specialized personnel; (e) reducing the adaptation and familiarization time for the users (operational personnel) with a minimal impact in their performance.

An important milestone in the roadmap for implementing CISE is represented by the EUCISE2020 project [4], which ran from 2014 to 2018 and promised to deliver an operational solution built on a common service-based architecture and open information exchange. In order to facilitate the adoption of CISE by third parties, EUCISE2020 openly published its CISE-based data model as an XML Schema specification accompanied by UML diagrams [3].

However, the adoption of CISE is compromised by the very serialization of the data model as an XML Schema. The latter does promote information sharing and exchange to an extent, but largely fails to deliver the fundamental additional benefits provided by ontologies, most prominently including a syntactically and semantically richer representation, enhanced semantic interoperability and semantic reasoning capabilities [13]. In order to capitalize on the critical benefits provided by ontologies, this paper presents a serialization of the EUCISE2020 data model as an ontology that will substantially encourage the extensive adoption of CISE. The proposed ontology is called *EUCISE-OWL* and aims to serve as a common representation framework for putting CISE in practical use. To the best of our knowledge, EUCISE-OWL is the first ontology attempting to fully implement the EU-wide approach represented by CISE. EUCISE-OWL is an outcome from the ROBORDER EU-funded project [20], after close collaboration with our end users who have extensive experience in deploying CISE in practice [14].

The paper is structured as follows. Section 2 presents the EUCISE2020 data model, while Section 3 describes our adopted process for converting the EUCISE2020 UML diagrams into an ontology. In Section 4, we present our EUCISE-OWL ontology, followed by an example in Section 5 for handling information exchange during a maritime crisis to demonstrate the use of the proposed ontology in practice. Section 6 assesses the EUCISE-OWL ontology, while Section 7 presents relevant approaches and models with similar scope. Finally, Section 8 concludes the paper with a discussion and key directions for future work.

2. The EUCISE2020 data model

The EUCISE2020 data model is based on the CISE Data Model v1.0 [1], which was defined in FP7 project CoopP (Cooperation Project Maritime Surveillance). CISE's key ambition is to serve as a common European format for sharing information across countries and sectors. Towards this direction, and in order to facilitate the adaptation of existing maritime monitoring systems in Europe, the CISE data model considers the corresponding data standards and identifies the most useful aspects for maritime monitoring authorities, as they were identified and validated by experts who participated in the CoopP project and represented all relevant sectors at EU and national level. CISE's main design principles included sector neutrality, flexibility, extensibility, simplicity and understandability.

In a nutshell, the CISE data model identifies seven core data entities (*Agent*, *Object*, *Location*, *Document*, *Event*, *Risk* and *Period*) and eleven auxiliary ones (*Vessel*, *Cargo*, *Operational Asset*, *Person*, *Organization*, *Movement*, *Incident*, *Anomaly*, *Action*, *Unique Identifier* and *Metadata*). Fig. 1 illustrates the core concepts of the CISE v1.0 data model.

Without extending the scope of the CISE data model, EUCISE2020 maintains the original concepts, but also defines some additional attributes, in order to consider additional data sources and to ensure that EUCISE2020 services can be implemented in practice. As already mentioned in the introduction, the EUCISE2020 data model is available as an XML Schema specification and as a set of UML diagrams.

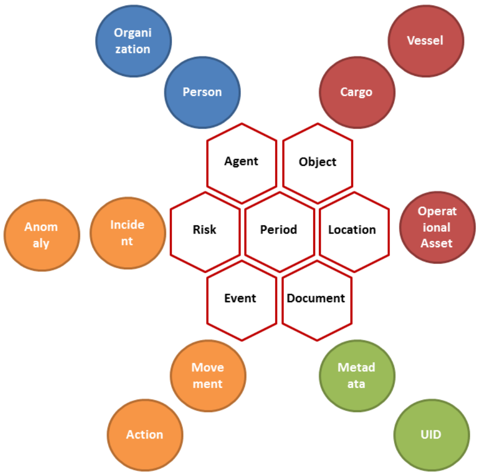

Fig. 1. CISE v1.0 core concepts [1].

3. Ontology creation

The Unified Modelling Language (UML) [21] and the Web Ontology Language (OWL 2) [32] are both established conceptual modelling languages that present significant similarities, despite being created on the basis of different contexts. A comparative overview of UML and OWL is presented in [10], [35]. Both language definitions are referred to comparable meta-models that follow the object-property modelling pattern. However, in contrast to UML, OWL 2 is fully built upon formal logic, which enables the application of logical reasoning in ontologies, a characteristic that can be used to discover inconsistencies in conceptual models and new knowledge that lies behind the asserted concepts and relations.

Many approaches already address the problem of reusing knowledge from existing UML class diagrams to develop ontologies, in automated or semi-automated procedures [10], [17], [34]. Regardless the degree of automation or the adopted technologies (XML, XSLT, translation algorithms, etc.), a precise conceptual correspondence between UML and OWL elements is defined, through a semantics-preserving schema translation [17], [34]. The model-conversion from UML to OWL follows simple conversion rules, the most common of which are presented in Table 1.

These mappings formed the groundwork in creating EUCISE-OWL, an ontology-based model of the domain of discourse that is fully compliant to the available well-established UML definitions presented in the EUCISE2020 data model [3]. The rules applied to convert the EUCISE2020 notions to OWL triples are presented in Turtle format [33] below, accompanied by indicative examples based on the `eucise:Agent` core class (see UML excerpt in Fig. 2).

Classes. Any core entity or class described in the EUCISE2020 data model is defined in EUCISE-OWL as an `owl:Class`, which is a subclass of `eucise:Entity` (subclass of `owl:Thing`). Below is the definition of class `eucise:Agent`.

```
eucise:Agent rdf:type owl:Class;
   rdfs:subClassOf eucise:Entity .
eucise:Entity rdf:type owl:Class;
   rdfs:subClassOf owl:Thing .
```

Attributes. In the EUCISE2020 data model, classes are related to other data types (either classes or literal values) through the declaration of relevant attributes. In EUCISE-OWL, object and data properties are naturally responsible for this representation. Two respective examples are illustrated below.

```
eucise:creator rdf:type
   owl:ObjectProperty ;
   rdfs:domain eucise:Metadata ;
   rdfs:range eucise:Agent .
```

```
eucise: isSuspect rdf:type
   owl:DatatypeProperty ;
   rdfs:domain eucise:Agent ;
   rdfs:range xsd:boolean .
```

Association Classes and Association Roles. In the EUCISE2020 data model, an association class is a specific type of class that defines the connection between the core entities of the model, using specific attributes called “association roles”. Association classes also have additional properties and datatypes of their own. Thus, we represent association classes in EUCISE-OWL as notions of type `owl:Class` (and not simply as object properties), whereas association roles define their related object properties. All association classes of the EUCISE2020 data model are grouped together under a top-level class named `eucise:AssociationClass` (subclass of `owl:Thing`). Below is an example of `eucise:AgentRisk`, as illustrated in Fig. 2.

```
eucise:AgentRisk rdf:type owl:Class ;
   rdfs:subClassOf eu-
   cise:AssociationClass .
eucise:AssociationClass
   rdfs:subClassOf owl:Thing .
eucise:involvedAgent rdf:type
   owl:ObjectProperty ;
   rdfs:domain eucise:AgentRisk;
   rdfs:range eucise:Agent .
eucise:involvedRisk rdf:type
   owl:ObjectProperty ;
   rdfs:domain eucise:AgentRisk;
   rdfs:range eucise:Risk .
```

Enumerations and Enumeration Types. Enumerations in the EUCISE2020 data model define the possible types of specific entities. In EUCISE-OWL, enumerations are represented as classes (`rdf:type owl:Class`) that additionally have a predefined list of asserted instances. All enumerations of the EUCISE2020 data model are grouped together under a top-level class named `eucise:EnumerationType` (subclass of `owl:Thing`). Below is the definition of `eucise:AgentRoleInRiskType`, which represents the role of an agent in relation to a risk; allowed values are: *cause*, *involved*, *reports*, *other*, *non-specified*.

```
eucise:AgentRoleInRiskType rdf:type
   owl:Class ;
   rdfs:subClassOf eu-
   cise:EnumerationType .
eucise:EnumerationType
   rdfs:subClassOf owl:Thing .
```

```
eucise:cause rdf:type eu-
   cise:AgentRoleInRiskType;
   rdf:type owl:NamedIndividual .
```

Metadata. The EUCISE2020 data model contains metadata descriptions in each defined component, in order to enrich their comprehensibility and facilitate their reuse. Those data were completely integrated into the ontological model through the adoption of well-known object, datatype and annotation properties (e.g. `rdfs:comment`, `skos:example`, `rdfs:seeAlso` and `rdfs:label`). Indicative initialisations are presented in triples below.

```
eucise:ClassC rdfs:comment "Description
   text"^^xsd:string ;
   skos:example "Example
      text"^^xsd:string ;
   rdfs:seeAlso <source_URL> ;
   rdfs:label "Label text" .
```

Table 1

Mappings between UML and OWL elements

UML Definitions	*OWL definitions*
UML package name	The namespace of `owl:Ontology` that corresponds to the UML package
Class	`owl:Class`
Association class	`owl:Class`
Enumeration class	`owl:oneOf`
Instance	Individual (`ex:instance rdf:type ex:Class . ex:Class rdf:type owl:Class`)
Attribute	`owl:DatatypeProperty`
Binary association	Pair or properties (relation `owl:inverseOf`)
Generalization (Class)	`rdfs:subClassOf`
Generalization (Association)	`rdfs:subPropertyOf`
Set of subclasses	`owl:unionOf`
Multiplicity	`owl:cardinality`, `owl:minCardninality`, `owl:maxCardinality`, `owl:FunctionalProperty`, `owl:InverseFunctionalProperty`
Navigable association	`rdfs:domain, rdfs:range`
Inheritance (default annotation: {incomplete[1], disjoint})	`ex:ClassB rdfs:subClassOf ex:ClassA .` `ex:ClassC rdfs:subClassOf ex:ClassA .` `ex:ClassB owl:disjointWith ex:ClassC`
Inheritance (annotation: {complete[2], disjoint})	`ex:ClassB rdfs:subClassOf ex:ClassA .` `ex:ClassC rdfs:subClassOf ex:ClassA .` `ex:ClassB owl:disjointWith ex:ClassC .` `ex:ClassA owl:disjointUnionOf(ex:ClassB ex:ClassC)`
Inheritance (annotation: {incomplete, overlapping[3]})	`ex:ClassB rdfs:subClassOf ex:ClassA .` `ex:ClassC rdfs:subClassOf ex:ClassA` (i.e., only inheritance is declared)

[1] *Incomplete* means that there are instances of the parent class `ClassA` which are neither of type `ClassB` nor `ClassC`.
[2] *Complete* means that each instance of the parent class `ClassA` is either of type `ClassB` or `ClassC`.
[3] *Overlapping* means that instances of the parent class `ClassA` may be both of type `ClassB` and of type `ClassC`.

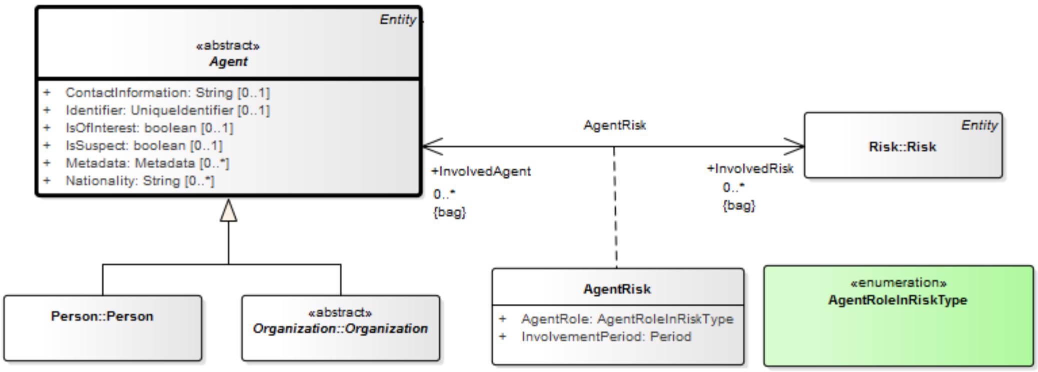

Fig. 2. A UML excerpt representing the core class `Agent` in the EUCISE data model.

4. EUCISE-OWL

The EUCISE-OWL ontology has been implemented in OWL 2, in accordance to the NeOn methodology Scenario 2: "*Reusing and re-engineering non-ontological resources*" [26], where the EUCISE2020 data model and UML diagrams played the role of the resources for the ontology we developed. The main purpose of EUCISE-OWL is to specify a common information sharing environment, based on a widely accepted format, in order to enhance the usability and adaptability of the EUCISE2020 data model. Such an ontology-based representation can easily be integrated in an information or decision support system for supporting knowledge representation, event triggering, action inference, and information dissemination to the authorities. In a nutshell, the proposed ontology enumerates a total number of 153 classes, 127 object properties and 135 data properties. The key ontology metrics are summarised in Table 2.

In compliance with the original model, there are 8 core elements defined in the ontology under class `Entity`: classes `Agent`, `Document`, `Event`, `Location`, `MeteoOceanographicCondition`, `Object`, `OperationalAsset` and `Risk`. Additional concepts are represented as subclasses of `owl:Thing`, as seen in Fig. 3. Moreover, we introduced two additional concepts: (i) the `AssociationClass` for representing classes that interconnect core classes, and (ii) the `EnumerationType` for representing sets of enumerated values that define different types of entities in specific concepts. In the EUCISE-OWL ontology, there are 10 association classes and 869 enumerated values (see *individual count* in Table 2).

Table 2

EUCISE-OWL ontology metrics

Metric	*Value*
Class count	153 (4)[4]
Object property count	127 (17)
Object property – Domain axioms count	116
Object property – Range axioms count	116
Data property count	135 (1)
Data property – Domain axioms count	132
Data property – Range axioms count	132
Individual count	869
DL expressivity	*SHIF*$^{(D)}$
Number of triples	6,209 (257)

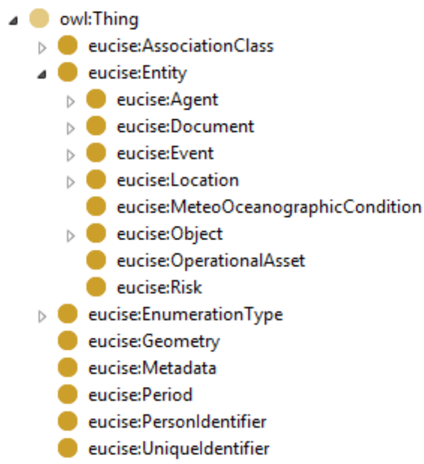

Fig. 3. Hierarchy of the EUCISE-OWL ontology's main notions.

In order to facilitate the semantic interoperability between EUCISE-OWL and existing data models, we performed an *ontology alignment*, which is the process of defining a set of correspondences between two or more ontologies. In general, the relation that holds between the matching entities may simply fol-

[4] The count of imported concepts is in parentheses.

low specific ontology language definitions (*equivalence*, *disjointness*, *hierarchy*) or even assert SPARQL queries, fuzzy relations and similarity measures [7]. An extensive survey of matching techniques is presented in [18]. Here, we focus only on hierarchy definitions via `rdfs:subClassOf` and `rdfs:subPropertyOf`; by linking concepts this way, we inherit all involved semantics declared in the adopted ontologies and we extend their definitions with respect to the context of our domain. Details of the aligned EUCISE-OWL entities with those from third-party ontologies are summarized in Table 3.

Table 3

EUCISE-OWL ontology alignment with third-party ontologies

EUCISE-OWL Class	*rdfs:subClassOf*
Agent	dce:Agent, dc:Agent, foaf:Agent, prov:Agent
Person	foaf:Person, prov:Person, dul:Person
Period	time:TemporalEntity, time:GeneralDateTimeDescription, time:GeneralDurationDescription, dc:PeriodOfTime
Geometry	geosparql:Geometry
Entity	geosparql:Feature
FileMediaType	dce:MediaType, dc:FileFormat
Event	event:Event, dul:Event, dcm:Event
Document	foaf:Document
SensorType	sosa:Sensor, ssn:SensingDevice
Incident	sosa:Observation
AgentEvent	prov:Association
AgentRoleInEventType, AgentRoleInRiskType, AgentRoleInAgentType,	prov:Role
Organization	prov:Organization
PlannedOperationsType, PlannedWorksType	prov:Plan

EUCISE-OWL Property	*rdfs:subPropertyOf*
latitude	geo:lat
longitude	geo:long
creator	foaf:maker
dateTime	time:hasTime
geometry	geosparql:hasGeometry, geo-sparql:hasDefaultGeometry

Prefix declarations	*Namespace*
dc	<http://purl.org/dc/terms/>
dce	<http://purl.org/dc/elements/1.1>
dcm	<http://purl.org/dc/dcmitype/>
dul	<http://www.ontologydesignpatterns.org/ont/dul/DUL.owl>
event	<http://purl.org/NET/c4dm/event.owl#>
foaf	<http://xmlns.com/foaf/0.1/>
geo	<http://www.w3.org/2003/01/geo/wgs84_pos#>
geosparql	<http://www.opengis.net/ont/geo sparql#>
prov	<http://www.w3.org/ns/prov#>
sosa	<http://www.w3.org/ns/sosa/>
ssn	<http://purl.oclc.org/NET/ssnx/ssn/>
time	<http://www.w3.org/2006/time#>

5. Example

To illustrate the efficiency and completeness of the implemented ontology, we present an operational scenario, inspired from [8] (*Use Case 25b: Investigation of antipollution situation (law enforcement)*). More specifically, the example presented here concerns a sea pollution incident reported when an oil spill was detected by a drone in its monitoring area. The main instances populated in the ontology as well as their interrelations are visualised in Fig. 4, using the Graffoo ontology visualization framework [9]. The circles indicate instances (real data), while their captions are written in the form of "`instance_XYZ::Class_ABC`", declaring the name and the type (class) of each instance correspondingly. All classes and relations mentioned in the diagram or the text below, belong to the EUCISE-OWL ontology, otherwise they are explicitly defined with their relevant prefixes.

As seen in Fig. 4, a drone (`drone_1`) is represented in the ontology as an `OperationalAsset` (`rdfs:subClassOf Object`), which is associated with an instance of *detection* event (`detection_1 rdf:type Action` and `Action rdfs:subClassOf Event`) via the `ObjectEvent` association class. Details of the detection event are included in `document_1` (`rdf:type AttachedDocument` and `AttachedDocument rdfs:subClassOf Document`). The event concerns an oil spill (`oilspill_1`), spotted in an area (`location_1`) under observation. The oil spill is represented in the ontology as an instance of class `PollutionIncident` (`rdfs:subClassOf Event`). Both events (`detection_1` and `oilspill_1`) are associated with each other through an instance of the association class `EventEvent`. Details of the pollution incident (e.g. the analysis dataset) may be potentially described through asserted values in `document_2` (`rdf:type EventDocument` and `EventDocument rdfs:subClassOf Document`). The occurred pollution incident may imply direct risks to the ecosystem and human health, the degree or details of which can be encoded through the assertions of relevant properties/values in an instance of `Risk` type (`risk_1`). On the basis of the observed pollution incident, of its severity and its implied risks, the interested authorities could be informed, the details of which can be represented as an instance of `Organization` type (`organization_1`).

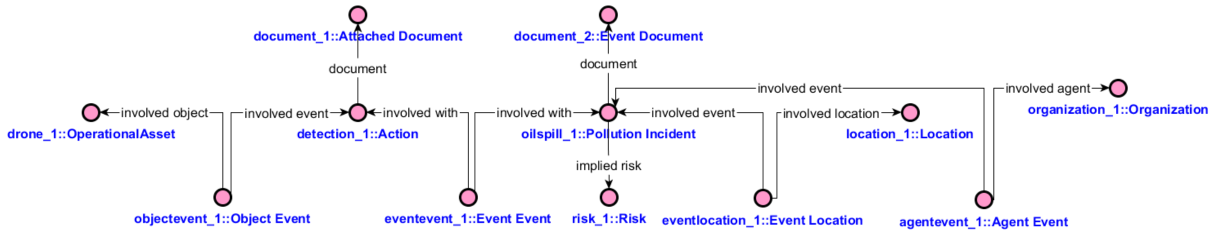

Fig. 4. Main instances in EUCISE-OWL for representing a sea pollution incident where an oil spill was detected.

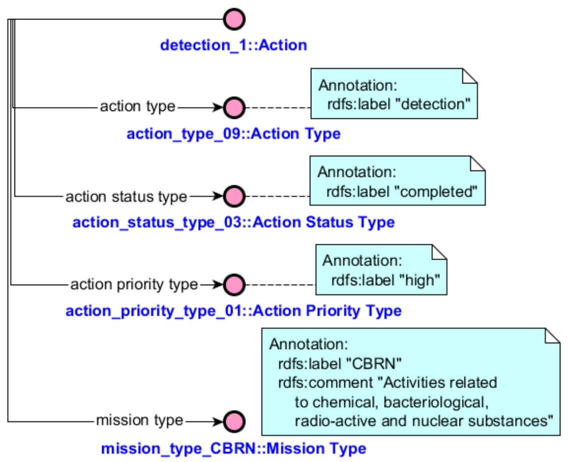

Fig. 5. Asserted instances to the `detection_1` instance of `Action` type.

Instances `detection_1` and `oilspill_1` represent events of different types (`Action` and `PollutionIncident`, respectively), and are associated with different properties and values. As seen in Fig. 5, an instance of `Action` type may be described through the assertion of relevant enumeration values that define the *mission type*, as well as the *type*, the *status* and the *priority* of the action. On the other hand, an instance of `PollutionIncident` may be described through the assertion of relevant enumeration values that define the *pollution type*, the *nature*, the *type*, the *severity* and *certainty* of the incident, as well as the *urgency* and *response type* of the event (Fig. 6). In the specific example, the severity of the incident was defined as *moderate*, i.e. possible threat to life or property (`severity_type_03`); considering this, responsive actions should be taken soon (`urgency_type_02`), according to the defined protocol (`response_type_04`).

Details about the actual geographical location (latitude and longitude) of the pollution incident can be presented through the assertion of properties and values in an instance of type `Location`. Additional metadata can be represented through relevant instantiations attached to instance `eventlocation_1` of the association class `EventLocation`. For example, as seen in Fig. 7, the *date* and *time* at which the oil spill was detected is represented through an instance of type `Period`; the location where the oil spill was detected is where the event *started* (enumeration value `location_role_in_event_type_01`); and, the area where the event takes place is now considered as *dangerous* (enumeration value `event_area_type_DGR`).

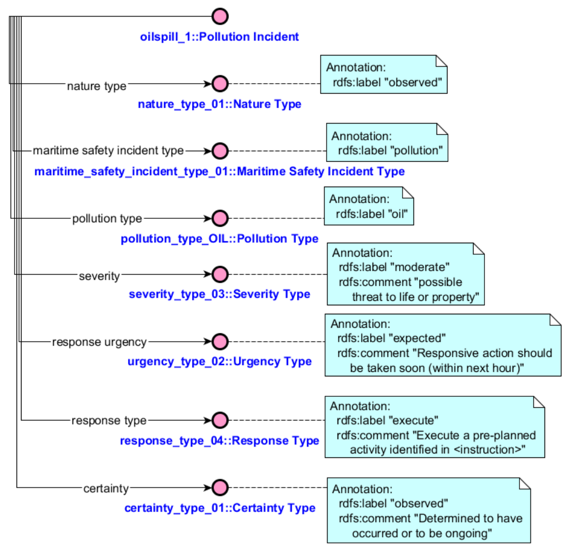

Fig. 6. Asserted instances to the `oilspill_1` instance of `PollutionIncident` type.

6. Ontology assessment

In order to assess the EUCISE-OWL ontology, we followed the guidelines in [22]. Initially, we focused on evaluating its modelling quality by submitting it to *OOPS!* (OntOlogy Pitfall Scanner!), an online system for testing an ontology against the most common modelling pitfalls [19]. OOPS! also provides an indicator (*critical*, *important*, *minor*) for each pitfall, according to the respective possible negative consequences. In the case of EUCISE-OWL, OOPS! did not detect any pitfall.

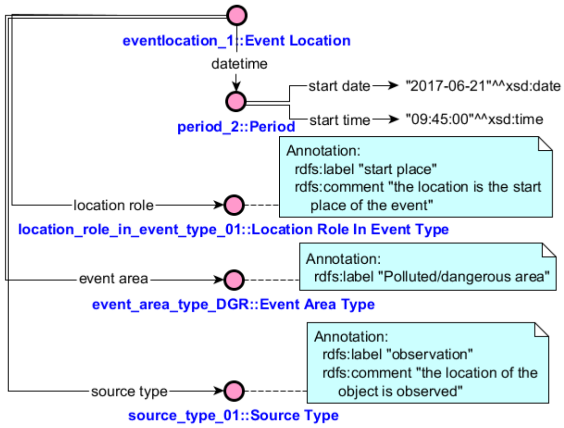

Fig. 7. Asserted instances to the `eventlocation_1` instance of `EventLocation` type.

Moreover, although the development of the proposed ontology was heavily based on a UML-to-OWL conversion from an existing data model (see Section 3), which was in turn designed with substantial contributions by domain experts (see Section 2), it would be interesting to get some insight into the assessment of the ontology's domain coverage. Thus, we submitted EUCISE-OWL to OntoMetrics [15], an online platform for calculating more advanced ontology metrics. Table 4 includes a subset of the metrics calculated by OntoMetrics that present the most interesting aspects of the ontology with regards to its domain coverage.

Table 4

EUCISE-OWL advanced metrics

Metric	*Value*
Attribute richness	1.694805
Inheritance richness	0.967532
Relationship richness	0.464029
Average population	5.603896
Class richness	0.558442

As indicated in [12], the first three metrics refer to the ontology's accuracy, while the other two refer to its conciseness:

– *Attribute richness* is defined as the average number of attributes (slots) per class, giving an indication of both the ontology design quality and the amount of information pertaining to instance data. The more slots that are defined, the more knowledge the ontology conveys. The value of 1.694805 demonstrates a high attribute richness for EUCISE-OWL, especially when taking into account the fact that a large subset of the classes in the ontology are enumeration types (see Sections 3 and 4), which correspond simply to sets of instances.
– *Inheritance richness* is defined as the average number of subclasses per class and describes the distribution of information across different levels of the ontology's inheritance tree. It is a good indication of how well knowledge is grouped into different categories and subcategories in the ontology. This metric distinguishes a horizontal from a vertical ontology. The value of 0.967532 for EUCISE-OWL indicates that the ontology covers a wide range of concepts, without delving too deep into their specialisations.
– *Relationship richness* is defined as the ratio of the number of (non-inheritance) relationships divided by the total number of relationships in the ontology and reflects the diversity of the types of relations. An ontology containing only inheritance relationships conveys less information than an ontology that contains a diverse set of relationships. The value for EUCISE-OWL in Table 4 indicates that the ontology has a mediocre richness of relationships, mostly due to the numerous enumeration types and association classes (see Sections 3 and 4).
– *Average population* corresponds to the number of instances compared to the number of classes and is an indication of the ontology population quality. Since, as already mentioned, EUCISE-OWL is rich in enumeration types, the specific value is considered very high.
– *Class richness* is related to how instances are distributed across classes. The number of ontology classes that have instances is compared with the total number of classes, giving an overview of how well the knowledge base utilises the knowledge modelled by the schema classes. The low value of the specific metric in Table 4 indicates that the ontology does not contain data that exemplifies all the class knowledge existing in the schema. This is reasonable, since EUCISE-OWL does not contain sample data, like e.g. the

instances discussed in the example presented in the previous section.

Furthermore, we wanted to assess the ontology's performance and efficiency with real data. Thus, we generated four (4) different serializations of the EUCISE-OWL ontology, with a different total number of populated incident instances per case (i.e., 100, 1K, 10K, 100K). These serializations were generated via a custom service we developed for populating the ontology with synthetic data, namely, individuals of specific subclasses of type `Event`; every such instance is asserted with:

- a severity value (1 out of 5 severity levels enumerated in the EUCISE-OWL),
- an instance of type `Period`, with timestamp referred to the event's occurrence,
- an instance of type `EventLocation`, which asserts to the event a `Geometry` instance of type `sf:Point` with random lat/long values,
- an instance of type `ObjectEvent`, for associating one or more instances of type `Object` with the event,
- a specification of the ship type (instance of corresponding enumeration), when the aforementioned `Object` is of type `Vessel`.

The experiments described below were executed with the use of *rdf4j*[5] and *OWL API*[6] for accessing/querying the serialized ontologies, stored in a GraphDB[7] installation located on a server machine (2xIntel® Xeon® Processor E5-2620 v4, 2.10 GHz, 125.8GB memory). We first measured the time elapsed for checking the consistency of the four ontologies, using the following reasoners: FaCT++ [29] (v1.6.2), Pellet [25] (v2.3.1), and HermiT [16] (v1.3.8). Indeed, all the reasoners verified the consistency of the ontologies, while Table 5 displays the average respective times (in ms) after 10 iterations for loading and consistency checking.

Table 5

Average times (ms) for loading (A) and consistency checking (B)

		No of incidents			
	Reasoners	100	1K	10K	100K
A	Fact++	273	679.9	2,156.7	21,968.3
	Pellet	331.2	699.2	2,313.9	19,811.9
	HermiT	327.9	946.5	2,202.5	24,205.5
B	Fact++	44.3	313.4	3,035.8	385,72.9
	Pellet	45.9	192.1	1,780.8	308,52.1
	HermiT	56.1	425.8	30,374.5	1,854,432.8

We then proceeded to evaluating the semantic reasoning performance, by creating three inference tasks (SPARQL queries Q1-Q3) with increasing complexity:

- Q1 searches for events of specific type (`PollutionIncident`) with a "severe" or "extreme" severity value
- Q2 filters from the populated ontologies those events that involve a high-speed vessel and occur after a specific period of time (date), ordered from the most to the least recent on
- Q3 reports those three events that took place in an area close to the one reported in a specific incident of interest. For calculating the distance between two locations, we utilize the `geof:distance` function from the GeoSPARQL query language[8].

For brevity, Fig. 8 presents only the last SPARQL query. Results are summarized in Table 6.

Table 6

Experimental set and evaluation results

Metadata	*Exp1*	*Exp2*	*Exp3*	*Exp4*
Total # of instances of type `Event`	100	1K	10K	100K
Total # of statements	11,891	46,414	391,571	3,841,119
Query	*Average elapsed time (ms) from 10 iterations*			
Q1	10.6	30.3	67.4	69.1
Q2	18	30.3	31	616.70
Q3	44.8	121	850	7,349.50

On the basis of the aforementioned results, it can be stated that the ontology's responses are given almost real time, regardless the number of triples, when queries do not involve complicated relations (like for example Q1). High response times are recorded in complex SPARQL queries, when the populated instances of `Event` type reach up to 100K; such a delay can be justified due to the additional effort that is required in Q3, for calling the GeoSPARQL function (`geof:distance`) in each repetition.

[5] `https://rdf4j.eclipse.org/`

[6] `http://owlcs.github.io/owlapi/`

[7] `http://graphdb.ontotext.com/`

[8] `https://www.opengeospatial.org/standards/geosparql`

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX eucise: <http://160.40.51.22/mklab_ontologies/ROBORDER/eucise#>
PREFIX sf: <http://www.opengis.net/ont/sf#>
PREFIX geo: < http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
PREFIX uom: <http://www.opengis.net/def/uom/OGC/1.0/>

SELECT DISTINCT ?incident
WHERE {
?incident rdf:type eucise:Event .
?eventlocation rdf:type eucise:EventLocation .
?eventlocation eucise:involvedEvent ?incident .
?eventlocation eucise:involvedLocation ?location .
?location eucise:geometry ?geometry .
?geometry rdf:type sf:Point .
?geometry geo:asWKT ?geomWKT .

?ref_eventlocation eucise:involvedEvent eucise:ref_incident_UUID_goes_here .
?ref_eventlocation eucise:involvedLocation ?ref_location .
?ref_location eucise:geometry ?ref_geometry .
?ref_geometry rdf:type sf:Point .
?ref_geometry geo:asWKT ?ref_geomWKT .

FILTER (?geometry != ?ref_geometry)
}
ORDER BY
ASC(geof:distance(?ref_geomWKT, ?geomWKT, uom:metre))
LIMIT 3
```

Fig. 8. Indicative SPARQL query applied for performance evaluation purposes.

7. Related work

There is currently a great interest in automated, on-time maritime surveillance, with an increasing attention towards efficient data handling. Besides EUCISE2020, an indicative list of ongoing relevant maritime EU-funded projects includes MARISA[9], AtlantOS[10], MARSUR[11], EMODnet[12], RANGER[13] and datAcron[14]. From these projects, only RANGER is aimed at being in line with established EU maritime standards, while a similar process is also underway for MARISA. On the other hand, only datAcron proposes an ontology-based solution for the representation of trajectories of moving objects' [23].

There are additional semantic approaches that model concepts relevant to the maritime domain, but usually they have a narrower scope than EUCISE-OWL. More specifically, in [31] an ontological representation of the different types of ships and relevant parameters is implemented, according to the AIS (Automatic Identification System), for maritime traffic analysis. Moreover, the detection or prediction of abnormal ship behaviour is investigated, by analysing semantic trajectories and geographical localizations of the maritime objects [2], [30]. In [11], an ontology-based representation of maritime regulations is proposed, for formulating maritime decision support rules in a machine-readable way.

To the best of our knowledge, CISE is the most concrete and complete model for implementing a common information sharing environment across countries and involved authorities, where all maritime surveillance operations can cooperate with one another and share data, following a common set of rules. Thus, its availability in an interoperable and easily adoptable form, as our proposed EUCISE-OWL representation, is of vital importance for operational use. As also stated before, to the best of our knowledge, EUCISE-OWL is the first attempt to fully capitalize on the EU-wide CISE framework, in order to develop an ontology model for facilitating information exchange in the maritime domain.

8. Conclusions and future work

This paper presented EUCISE-OWL, an ontology representation of the CISE data model that constitutes an EU-wide collaborative initiative for facilitating information sharing between maritime monitoring authorities. EUCISE-OWL is an outcome from the ROBORDER EU-funded project, and we are currently deploying it as a common platform for semantically integrating analysed data from heterogeneous sensors and for performing semantic reasoning on top of this data, in order to facilitate decision support for authorities. Within ROBORDER,

[9] https://www.marisaproject.eu/

[10] https://www.atlantos-h2020.eu

[11] https://www.eda.europa.eu/what-we-do/activities/activities-search/maritime-surveillance-(marsur)

[12] http://www.emodnet.eu

[13] https://ranger-project.eu/

[14] http://datacron-project.eu/

EUCISE-OWL is addressing the project's pilot use cases, which include addressing pollution incidents at sea (see Section 5), tracking suspicious vessels, countering illegal activities etc. Therefore, the ontology has only been evaluated in experimental settings (e.g. see Section 6), but the upcoming pilot demonstrations will provide an excellent opportunity for actually evaluating the utility of the ontology in practice.

As for our future goals, ROBORDER serves as a good testbed for the wider adoption of our proposed ontology and its potential extensions in a wider variety of scenarios, like e.g. border trespassing in the sea or on the land, or in applying robotics for enhanced security [24]. Within ROBORDER, we have the substantial advantage of collaborating with maritime authorities who have been deploying CISE in practice and could greatly contribute in encouraging the wider use of our EUCISE-OWL model through disseminating the outcomes of this work within their respective networks. On our part, we will be able to support the required technical implementations, in order to facilitate the wider and easier adoption of the model. Finally, a more long-term goal is to work towards including EUCISE-OWL in the EU's SEMIC action[15] for promoting semantic interoperability amongst the EU Member States.

Disclaimer

The work within the ROBORDER project that involves the EUCISE-OWL ontology is characterised as "classified". Thus, the ontology is not (yet) publicly available. Access can be granted after submitting to the corresponding author a request that will be evaluated on a per case basis.

Acknowledgements

This work was supported by the ROBORDER project, partially funded by the European Commission under grant agreements No 740593.

References

[1] D. Berger, J. Hermida, F. Oliveri, & G. Pace. The Entity Service Model for CISE - Service Model Specifications. Technical Report, Joint Research Centre of the European Commission, 2017. [online] Available at: https://webgate.ec.europa.eu/maritimeforum/en/node/4039. Last accessed: February 2020.

[2] S. Brüggermann, K. Bereta, G. Xiao & M. Koubarakis. Ontology-based Data Access for Maritime Security, in: *European Semantic Web Conference*, Springer, 2016, pp. 741-757, https://doi.org/10.1007/978-3-319-34129-3_45 .

[3] EUCISE2020 Consortium. Deliverable D4.3 – Technical Specifications Revision 1.0, ANNEX B: EUCISE2020 Data Model, 2015. [online] Available at: http://www.eucise2020.eu/media/1131/d4_3-annexb.pdf.

[4] EUCISE2020 FP7 project: http://www.eucise2020.eu.

[5] European Commission – Directorate-General for Maritime Affairs and Fisheries. Integrating Maritime Surveillance – Communication from the Commission to the Council and the European Parliament on a draft roadmap towards establishing the Common Information Sharing Environment for the surveillance of the EU maritime domain. Publications Office of the EU, 2011. DOI: https://doi.org/10.2771/64104.

[6] European Commission. Maritime Affairs, Integrated Maritime Policy. [online] Available at: https://ec.europa.eu/maritimeaffairs/policy_en.

[7] J. Euzenat, & P. Shvaiko. *Ontology Matching,* 2nd Edition, Springer, 2013, https://doi.org/10.1007/978-3-642-38721-0.

[8] G. Fabbri, C. Medaglia, J. Tyni, I. Mattila, C. Matarazzi, and G. Codispoti. EUCISE2020: European Test Bed for the Maritime Common Information Sharing Environment, in *Proc. of the 91st IASTEM Int. Conf.*, 2017.

[9] R., Falco, A., Gangemi, S., Peroni, D., Shotton, & F. Vitali. Modelling OWL ontologies with Graffoo, in *Proc. of the European Semantic Web Conference*, Springer, 2014, pp. 320-325, https://doi.org/10.1007/978-3-319-11955-7_42.

[10] K. Falkovych, M. Sabou, & H. Stuckenschmidt, UML for the semantic web: Transformation-based approaches, *Knowledge Transformation for the Semantic Web* **95** (2003), p. 92.

[11] M. Hagaseth, L. Lohrmann, A. Ruiz, F. Oikonomou, D. Roythorne, & S. Rayot. An Ontology for Digital Maritime Regulations, *Journal of Maritime Research,* **13**(2) (2016), 7-18.

[12] H. Hlomani, & D. Stacey, D. Approaches, methods, metrics, measures, and subjectivity in ontology evaluation: A survey. *Semantic Web Journal*, **1**(5) (2014), 1-11.

[13] M. Klein, D. Fensel, F. van Harmelen, F., & I. Horrocks. The relation between ontologies and XML schemas. *Electronic Trans. on Artificial Intelligence,* **6**(4) (2001).

[14] P. Klimek, J. Varga, A.S. Jovanovic, & Z. Székely. Quantitative resilience assessment in emergency response reveals how organizations trade efficiency for redundancy. *Safety science*, **113** (2019), 404-414, https://doi.org/10.1016/j.ssci.2018.12.017.

[15] B. Lantow. OntoMetrics: Putting Metrics into Use for Ontology Evaluation. In *KEOD* (2016), 186-191, https://doi.org/10.5220/0006084601860191.

[16] B. Motik, B., R. Shearer, & I. Horrocks. Optimized reasoning in description logics using hypertableaux, in: *Proc. of the Int. Conf. on Automated Deduction (CADE)*, Springer, 2007, pp. 67-83, https://doi.org/10.1007/978-3-540-73595-3_6 .

[17] H.-S. Na, O.-H. Choi, & J.-E. Lim. A method for building domain ontologies based on the transformation of UML models, in: *Proc. of the 4th Int. Conf. on Software Engineering Research, Management and Applications*, 2006, pp. 332-338, https://doi.org/10.1109/SERA.2006.4.

[15] http://semic.eu/

[18] L. Otero-Cerdeira, F.J. Rodríguez-Martínez, & A. Gómez-Rodríguez. Ontology matching: A literature review. *Expert Systems with Applications*, **42**(2) (2015), pp. 949-971, https://doi.org/10.1016/j.eswa.2014.08.032.

[19] M. Poveda-Villalón, A. Gómez-Pérez, & M.C. Suárez-Figueroa. OOPS! (OntOlogy Pitfall Scanner!): An on-line tool for ontology evaluation. *International Journal on Semantic Web and Information Systems (IJSWIS)*, **10**(2) (2014), 7-34, https://doi.org/10.4018/ijswis.2014040102.

[20] ROBORDER H2020 project: `https://roborder.eu/`.

[21] J. Rumbaugh, I. Jacobson, & G. Booch. *The Unified Modelling Language Reference Manual*, Addison Wesley, 2017.

[22] M. Sabou, & M. Miriam Fernandez. Ontology (network) evaluation. *Ontology engineering in a networked world*, Springer, 2012, pp. 193-212, https://doi.org/10.1007/978-3-642-24794-1_9.

[23] G. M. Santipantakis, G. A. Vouros, A. Glenis, C. Doulkeridis, & A. Vlachou. The datAcron Ontology for Semantic Trajectories, in: *European Semantic Web Conference*, Springer, 2017, pp. 26-30, https://doi.org/10.1007/978-3-319-70407-4_6.

[24] Z. Székely. Application of Robotics for Enhanced Security: European Research on Security Robots, in: Péter, Korondi (Ed.) *Proceedings of ARES'14: Workshop on Application of Robotics for Enhanced Security*, 2014, pp. 11-15.

[25] E. Sirin, B. Parsia, B.C. Grau, A. Kalyanpur, & Y. Katz, Pellet: A practical OWL-DL reasoner, *Journal of Web Semantics* **5**(2) (2007), 51-53, https://doi.org/10.1016/j.websem.2007.03.004.

[26] M. Suárez-Figueroa, A. Gómez-Pérez, & B. Villazón-Terrazas. How to write and use the Ontology Requirements Specification Document, in: *Proc. of the Confederated International Conferences, CoopIS, DOA, IS, and ODBASE 2009 on On the Move to Meaningful Internet Systems: Part II (OTM '09), Robert Meersman, Tharam Dillon, and Pilar Herrero (Eds.)*, Springer, 2009, pp. 966-982, https://doi.org/10.1007/978-3-642-05151-7_16.

[27] I. Tikanmäki. Common Information Sharing on Maritime Domain – A Qualitative Study on European Maritime Authorities' Cooperation, in: *Proc. of the 9th Int. Joint Conf. on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, 2017, pp. 283-290, http://dx.doi.org/10.5220/0006582502830290 .

[28] I. Tikanmäki, & H. Ruoslahti. Increasing Cooperation between the European Maritime Domain Authorities. *Int. J. of Environmental Science,* International Association of Research and Science **2** (2017), 392–399.

[29] D. Tsarkov, & I. Horrocks. FaCT++ Description Logic Reasoner: System description, in: *Int. Joint Conf. on Automated Reasoning (IJCAR)*, Springer, 2006, pp. 292-297, https://doi.org/10.1007/11814771_26.

[30] A. Vandecasteele, & A. Napoli. An Enhanced Spatial Reasoning Ontology for Maritime Anomaly Detection. In *Proc. of the 7th Int. Conf. on System of Systems Engineering*, 2012, https://doi.org/10.1109/SYSoSE.2012.6384120.

[31] G. de Vries, V. Malaisé, M. van Someren, P. Adriaans, & G. Schreiber. Semi-Automatic Ontology Extension in the Maritime Domain. In *Proc. of the 20th Belgian-Dutch Conference on Artificial Intelligence*, University of Twente, Faculty of Electrical Engineering, Mathematics and Computer Science, 2008, pp. 265-272.

[32] W3C Recommendation. *OWL 2 Web Ontology Language Document Overview (Second Edition)*, 2012. [online] Available at: `http://www.w3.org/TR/owl2-overview/`.

[33] W3C Recommendation. *RDF 1.1 Turtle Terse RDF Triple Language*, 2014. [online] Available at: `https://www.w3.org/TR/turtle/`.

[34] Z. Xu, Y. Ni, W. He, L. Lin, & Q. Yan, Automatic extraction of OWL ontologies from UML class diagrams: a semantics-preserving approach. *World Wide Web,* **15**(5-6), (2012), 966-982, https://doi.org/10.1007/s11280-011-0147-z.

[35] J. Zedlitz, & N. Luttenberger. Transforming between UML conceptual models and OWL 2 ontologies. *Terra Cognita 2012 Workshop*, 2012, p. 15.