OWL Representation of the Common Information Sharing Environment Data Scheme for the Maritime Domain

Nathalie Aussenac-Gilles, Catherine Comparot, Antoine Dupuy, Nabil El Malki, Ronan Tournier, Ba Huy Tran and Cassia Trojahn
[a] *Institut de Recheche en Informatique de Toulouse, Université de Toulouse, France*
E-mail: prenom.nom@irit.fr

Abstract. The Common Information Sharing Environment (CISE) is the result of a collaborative initiative aiming at promoting automated information sharing among maritime surveillance authorities in Europe. It provides a decentralized framework and a data model for point-to-point information exchange across sectors and borders. The exploitation of the CISE data model is however limited by its serialization via an XML schema. Such a serialization is known to be insufficient to provide semantic interoperability, ontology-based data access, and reasoning capabilities to the different systems relying on CISE. This paper presents an OWL representation of the two main versions of CISE: the CISE data model (current version, 1.5.3) and its extended version (eCISE) that enhances the CISE maritime vocabulary and expands its scope to include land surveillance and operational information exchange. These ontologies are the outcome of an improved process of transforming XML schemes (XSD) into OWL and of validation and correction efforts by domain experts. Both generated ontologies and the XML to OWL converter are publicly available.

Keywords: maritime domain, CISE, ontologies, XML

1. Introduction

Making maritime surveillance systems interoperable is crucial for the cooperation between countries, especially in case of maritime crises in border areas. However, the heterogeneity between national systems and the data structures of the different actors have raised many interoperability issues. In order to better enable maritime authorities to exchange information in an automatic and secure way, the Common Information Sharing Environment (CISE)[1] has been proposed. It provides a decentralized framework and a data model for point-to-point information exchange across sectors and borders. This initiative involves more than 300 European and national authorities with maritime surveillance responsibilities, performing different operational surveillance tasks. These authorities benefit directly from being connected to the CISE network, for purposes as diverse as maritime transport safety and security, fisheries control, pollution risk management, and defense. Since 2014, CISE has been selected to support the implementation of the European Union Maritime Security Strategy (EUMSS).

Adoption of the CISE data model[2] and its different versions – in particular, *Extended-CISE* (eCISE) [1] – is, however, limited by its serialization via an XML schema, the semantics of which is not rich enough to guarantee fully semantic interoperability of the data, to be used as a support for reasoning, or still to offer a support to ontology-based data access (OBDA).

[1] http://www.emsa.europa.eu/cise.html (accessed on 15th July 2022).

[2] https://emsa.europa.eu/cise-documentation/cise-data-model-1.5.3/ (current version) (accessed on 11th April, 2022).

These functionalities are however essential to associate data coming from different CISE data sources, to verify them or to infer new facts. In order to address these issues, a semantic representation the CISE data model is the appropriated solution. A first effort in such a direction has been proposed by the European project ROBORDER [2], which generated an ontological representation of the CISE data model as defined in the EUCISE2020 project [3], by converting the UML data model into OWL. Although the ontology and its construction process are described (incompletely) in the cited paper, both ontology and transformation tool are not publicly available. Hence, new efforts are required to offer to the maritime community such resources, considering also the fact that the underlying model has so far evolved.

Although moving from XML data or XSD schemes to a semantic representation (RDF, RDFS, or OWL) has been a long-standing issue in the Semantic Web field, a simple and automatic transformation is rarely correct. This process faces the difficulty of handling anonymous nodes, dealing with the representation of complex nodes (like enumerations), capturing the semantics of purely structural tags, or still producing structuring-related constructors, as addressed in several works on the topic [4–7]. However, those tools are rarely fully operational, available and up to date.

This paper contributes to overcome those issues, by proposing an OWL representation of the two main versions of CISE: the CISE data model (current 1.5.3 version) and its enriched version (eCISE) that enhances the CISE maritime vocabulary and expands its scope to include land surveillance and operational information exchange. In order to improve interoperability to existing ontologies, ontology alignments have been manually added. Rich ontology metadata and their publication on permanent (w3id) make these resources also compliant with the FAIR principles [8].

In addition to the ontologies themselves, we propose an original process of transforming an XML schema (XSD) into OWL. While generic, this process also takes into account the particularities of the CISE data model to generate an initial OWL model, which domain experts then validate and correct, as XSD source schemas do not always conform to the model specifications. It integrates and extends existing work, formally described in several papers, but which had never been put together in a single processing chain and were not fully reusable. The proposed process significantly improves the handling of collections, previously represented as collections of `owl:oneOf`, which slows down performance when querying this data. Moreover, the OWL translation of UML association classes which are numerous in the CISE models has been simplified. The code is fully available under MIT license[3].

This work has been carried out in the context of the H2020 EFFECTOR[4] project, which aims at proposing an interoperability framework and associated data fusion and analysis services for maritime surveillance and border security. Thus, EFFECTOR aims at improving the decision support process and fostering collaboration between maritime actors at local, regional and transnational levels. The CISE data model plays an essential role in the project acting as a pivot model. As the exchanged messages are stored in several internal databases, an ontology-based data access system (OBDA) is therefore set up to contribute to the interoperability of systems and to facilitate data exchanges between partners. Moreover, in order to help the operator in charge of maritime surveillance, the ontology allows to produce inferences and to generate new facts indicating potential anomalies.

The rest of this paper is organised as follows. Section 2 presents the state of the art on maritime ontologies and on existing solutions to generate OWL models from XSD schemes. Section 3 introduces the CISE model and its extension eCISE. The requirements in terms of ontologies are then discussed in Section 4). The ontology construction methodology is introduced in Section 5, followed by the presentation of the generated ontologies (CISE-OWL and eCISE-OWL in the rest of the paper) and their evaluation in Section 6. Finally, Section 7 concludes the paper and outlines the perspectives for future work.

2. Related work

Maritime ontologies. Automated maritime surveillance has attracted particular interest in recent years. This is corroborated by a number of projects addressing the various challenges of the domain (ROBORDER, EUCISE2020, ANDROMEDA, MARISA, datAcron, and CoopP, to name a few). Semantic technologies have proven their relevance in supporting interoperablity and facilitating automated information sharing between maritime systems. In particular, in the context of the datAcron project, an ontology has been proposed to represent trajectories of moving objects [9].

[3]https://github.com/EFFECTOR-IRIT/XTR

[4]https://www.effector-project.eu/

Closer to our proposal, in the context of the ROBORDER project, the EUCISE-OWL ontology results from a UML to OWL conversion of the EUCISE2020 data model (an extension of the CISE Data Model v1.0 to cover additional data sources). EUCISE-OWL was the first attempt to fully exploit the CISE data model to develop an ontology that facilitates information exchange in the maritime domain. However, the ontology and the transformation process are no publicly available.

Out of these projects, several maritime ontologies have also been proposed in the literature. In [10], an ontology is proposed to represent a number of ships types together with their relevant parameters. This ontology, based on the AIS (Automatic Identification System), has been used in the task of maritime traffic analysis. In [11], the analysis of semantic trajectories and geographical locations of maritime objects is performed using domain ontologies. The approach combines the processing of static and real-time data from different sources using ontology-based data access (OBDA) techniques. Another task of this domain that requires semantic representations concerns the detection or prediction of abnormal ship behavior. In [12, 13], spatial ontologies and semantic representations of trajectories are used to characterize abnormal ship behavior, based on formal semantic properties used to reason about the data. While these methods are mainly based on ontologies created manually or derived from non ontological resources such as XML schemas or UML diagrams, some works consider also constructing maritime ontologies from texts, as in [14].

Finally, from another angle, in [15], a maritime regulation ontology has been defined describing for example port procedures and ship maintenance, in order to evaluate the impact of new regulations and to trace their legislative origin.

From XML/XSD to OWL. Semantic representation of XML data or XSD schemas has been the object of many proposals in the Semantic Web domain, since many years. However, a simple and automatic transformation is rarely efficient and correct. In particular, when the elements defined between tags are themselves complex, they fall under several related XSD types represented by multiple properties. They can even contribute to both enriching the ontology and populating it. This is the case of the exploitation of XML records in the MOANO project for example [16]. In all cases, the passage from XSD to OWL classes comes up against the difficulty of managing anonymous nodes, of dealing with the representation of complex nodes, of enumerations, of managing XSD types, or even tags linked to structuring and not semantic.

Several approaches have been implemented in the literature. A first set of tools, called "lifting" tools, convert an XML schema (XSD) into an RDF-based schema, such as RDFS or OWL. This is the case of XML2OWL (which starts from an XML document or an XSD) or XSD2OWL[5] (starting from an XSD). Another set of tools covers the most classical approaches, which consist in using XSLT as a XSD schema transformation language, by considering the XML serialization of RDF (RDF/XML syntax) as a particular XML schema. XSLT is the basis of GRDDL[6], a mechanism that allows to add tags in an XML document to indicate that the described data can be translated to RDF using XSLT. Other proposals as in [17], developed an XSLT script that transforms a semantic sitemap in XML to voID RDF/XML format, but these allow building only rudimentary descriptions, which should then be completed to by manually editing the RDF file. Unfortunately, the tools presented in research papers are rarely available and up to date.

A W3C wiki points out that while XSLT is suitable for converting a majority of XML models into RDF, it has several limitations: it generates very verbose models that are unreadable by a human; in the case of complex models, it does not know how to deal with all situations, such as nested structures or long text between tags. The MIT Simile site[7] provided a list of some other RDF-izers, most of which are no longer available or do not fully support XML. The W3C provides another list of XML-to-RDF conversion tools, including TopBraid Composer (a commercial software) with a plugin that handles XSD to OWL; KREXTOR, a platform that can handle several variants of XML using XSLT transformations; Rhizomik, which uses XML2RDF and XSD2RDF; GRDDL; XHTML, etc. A powerful alternative to XSLT is based on RML, which is more powerful in dealing with complex schemas, and whose format lends itself well to human-readable reformulation. Finally, the SDM-RDFIZER[8] software relies on RML to handle a variety of formats (CSV, JSON, RDB, XML). Its advantage is scaling, as the rules are optimized to handle large volumes of data.

[5] https://gist.github.com/pebbie/5704765
[6] Gleaning Resource Descriptions from Dialects of Languages
[7] http://simile.mit.edu/
[8] https://github.com/SDM-TIB/SDM-RDFizer

However, the generation of enumerations and association classes can not be fully customised.

3. CISE and eCISE Models

The CISE data model has the ambition to serve as a common European format for sharing information across countries and sectors. It identifies the most useful aspects for maritime monitoring authorities, as they were identified and validated by experts and represented all relevant sectors at EU and national level [2]. The current version of the CISE data model is the 1.5.3[9]. In [2], the CISE model defined in the EUCISE2020 project [3] has been used (version 1.0). This version, no longer publicly available, has been also used as basis to create the extended version of the CISE [1] (eCISE 2.4.0). In the following, the CISE and eCISE data models are introduced.

3.1. From CISE to eCISE

The CISE data model describes seven core entities: `Agent`, `Object`, `Location`, `Document`, `Event`, `Risk`, `Period`, together with eleven complementary entities: `Vessel`, `Cargo`, `Operational Asset`, `Person`, `Organization`, `Movement`, `Incident`, `Anomaly`, `Action`, `Metadata` and `Unique Identifier`. This model allows the different authorities to benefit from a common vocabulary to describe the observed events. In Figure 3.1, the entities of the CISE model correspond to the uncolored hexagons.

The eCISE data model [1] extends the CISE vocabulary for the maritime and land domains. It provides Automatic Identification System (AIS) and radar sensor information, and lists a more complete set of maritime and land-based anomalies and rules, with a significant number of types for each of its entities, such as Vessel. In Figure 3.1, the central entities of eCISE that complement those of the CISE model, correspond to the colored hexagons.

3.2. XML and UML of CISE and eCISE

The CISE and eCISE data models are described in a specification document (UML class diagrams) and implemented in XSD (XML schemes). The XSD files were produced using transformations, i.e. a set of mapping rules indicating how to generate XSD elements from UML elements. The choices made during this process have impacted the resulting XSD structures and must be taken into account when generating the corresponding OWL representation. For both CISE and eCISE, each .xsd file represents one or more entities of the model, where each entity is represented via a tag.

In these schemes, the notion of specialization between entities is represented by the tag `xs:extension`. The elements (in the XML sense) which more specifically make up an entity are then described in XSD within the the tag `xs:complexContent`. They are listed in the order in which they should appear in a `xs:sequence` tag. Each of these elements corresponds either to a property specific to the entity, or to an association with another entity. The fragment below describes the entity `Vehicle` as a subclass of `Object` linked to the `Cargo` entity by an association represented in XML by the property `CargoRel`; the explicit value 0 of `xs:minOccurs` and the implicit value 1 of `xs:maxOccurs` indicate a cardinality of (0,1); the property is therefore optional and can only appear once. Note that `Cargo` designates a cargo (i.e., a set of goods transported by a vehicle between two ports), and not a type of ship. The corresponding UML specification of the class `Object` is given in Figure 3.2.

```
<xs:complexType name="Vehicle" abstract="true">
 <xs:annotation>
  <xs:documentation>
   The Vehicle is a sub-class of Object [...]
  </xs:documentation>
 </xs:annotation>
 <xs:complexContent>
  <xs:extension base="object:Object">
   <xs:sequence>
    <xs:element name="CargoRel"  minOccurs="0">
     <xs:complexType>
      <xs:complexContent>
       <xs:extension base="rel:Relationship">
        <xs:sequence>
         <xs:element name="Cargo"
                     type="cargo:Cargo"
                     minOccurs="0"/>
        </xs:sequence>
       </xs:extension>
      </xs:complexContent>
     </xs:complexType>
    </xs:element>
   </xs:sequence>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>
```

[9] https://emsa.europa.eu/cise-documentation/cise-data-model-1.5.3/ (accessed on 12th 2022)

Fig. 1. Overview of the vocabularies of the CISE base model (entities in white) and the eCISE model (CISE entities enriched by the entities in color), from [1].

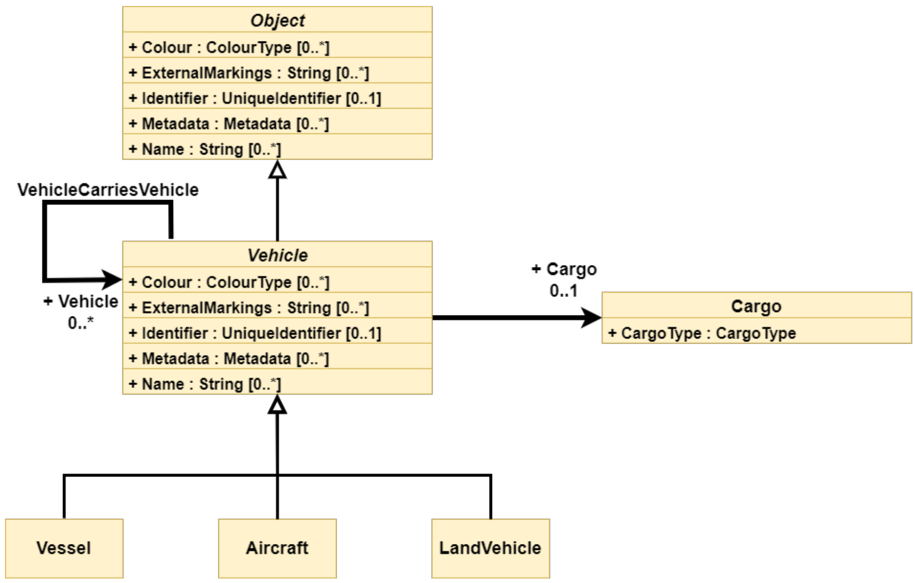

Fig. 2. UML specification of the class `Object`, adapted from [1].

4. The need for a common ontology

While CISE has been adopted as common exchange format and represents the vocabulary of maritime surveillance, each system in maritime coordination and rescue centers may have its own vocabulary and data schemes. A common ontology provides a high-level pivot model that facilitates interoperability on the one hand and allows reasoning on anomalies on the other hand. In the following, the two scenarios retained in EFFECTOR that justify the use of such a common ontology are presented. While the produced resources (ontologies and converter processes) could be made publicly available, for confidential reasons, the real EFFECTOR scenarios could not be reproduced.

4.1. Ontology-Based Data Access

Ontology Based Data Access (OBDA) [18] is a paradigm of accessing data trough a conceptual layer. Usually, the conceptual layer is expressed in the form of an RDF Schema or an OWL ontology, and the data is stored in relational databases. The terms in the conceptual layer are mapped to the data layer using mappings which associate to each element of the conceptual layer a (possibly complex SQL) query over the data sources. The virtual graph can then be queried using an RDF query language such as SPARQL.

While the different EFFECTOR partners receive CISE messages through the CISE network, the information exchanged is stored in their local databases that can also cover data from other sources (AIS, SAT-AIS, Radar, etc.). As EFFECTOR aims at improving the decision support process and foster collaboration of maritime actors, the possibility of accessing cross-actors databases is a key point. In that direction, adopting an OBDA strategy, where the conceptual layer relies on the use of an ontology built on a common cross-border, has been proposed and adopted.

4.2. Reasoning on CISE messages

Another use of the ontology concerns the reasoning capacity in order to infer new elements from the instantiation of specific parts of the ontology, according to the different scenarios defined in the project. It is in particular about offering a support to the human operator in the process of anomaly detection. As stated above, for confidentiality reasons, the details of the scenarios can not be communicated and the following illustrating examples are classical cases in the domain. The concepts and relations in these examples are the ones from the eCISE-OWL ontology, presented in Section 6.

Speed alert If a vessel is traveling faster than the speed allowed for the particular vessel type, a speed alert should be generated. Based on an example from [13], this alert can be generated via a SWRL rule (in a human readable syntax) adapted to the eCISE-OWL vocabulary, as follows:

```
ObjectLocation(?objectLocation) ∧
hasObjectLocationLocation(?objectLocation, ?location) ∧
hasObjectLocationObject(?objectLocation, ?vehicle) ∧
Vessel(?vehicle) ∧
speed(?objectLocation, ?speed) ∧
maximumSpeed(?vehicule, ?maxSpeed) ∧
greaterThan(?speed, ?maxSpeed) →
MaritimeAnomaly(?anomaly) ∧
hasMaritimeAnomalyMaritimeAnomalyType(?anomaly,
                                    :HighSpeed)}
```

Pollution risk If there is an imminent risk of collision between two vessels and at least one of the vessels involved has a dangerous cargo, then there is a risk of pollution (which is a type of risk):

```
ObjectEvent(?objectEvent) ∧
hasObjectEventEvent(?objectEvent,
 :VesselImminentCollision) ∧
hasObjectEventObject(?objectEvent, ?vehicle) ∧
Vessel(?vehicle) ∧
hasVehicleCargo(?vehicle, ?cargo) ∧
hasCargoContainedCargoUnit(?cargo, ?containmentUnit) ∧
hasContainmentUnit(?containmentUnit,
                  ?pollutionCode_PollutionCodeType)
→ Risk(?risk), hasRiskRiskType(?risk, :Pollution)}
```

5. Reengineering of non-ontological resources

The methodology for constructing the ontologies is consistent with "Scenario 2" *Reuse and Reengineer Non-Ontological Resources (NOR)* of the NeOn methodology [19, 20]. The process of reengineering NOR has been defined to transform NOR into ontologies. In our case, the NOR correspond to XSD schemas that are homogeneous in their underlying data model. In this case, [20] suggests to go to the conceptual level in order to study the correspondences between the source model and its OWL conversion, what corresponds here in explaining how each element of the XML schema should be translated into RDF or OWL. The (manual) formatting of these rules to form a conversion pattern is a key step of the translation, which allows to automate the translation of data described according to this model. Here, the problem is to transform a schema, and to treat each similar structure in this schema in the same way, according to the same rules, as discussed in the following.

5.1. From XML to OWL

Since there is no operational and freely available tool to transform XML models into RDF that can fully handle the specificity of CISE in a proper way, a transformation process has been implemented for converting XML to RDF that meet the requirements of the CISE model. It seemed desirable that this process should also automatically produce descriptions of elements (via `rdfs:comment` for example) from the documentation. In order to process the eCISE model, in particular, the difficulties to overcome are notably related to the large number of enumerations (possible types of specific entities) to be converted, the lack of information regarding the naming of association classes (n-ary relations) and the conversion of cardinality constraints.

The proposed transformation process, inspired by the work of [21], applies a set transformation rules. It is implemented in Python and reuses `rdflib`. It scans XSD sources and external sources to extract the elements needed to build the ontology. For each type of XSD element, a corresponding transformation rule is applied to the OWL formalism. In a second step, an other script retrieves the documentation of the CISE and eCISE data models to extract the comments (`rdfs:comment`) that will be used to document the entities of the ontologies.

Figure 3 shows the schematic of the XSD to OWL conversion process. This process makes use of several external data sources, as described below. The tool that transforms the data in XSD format into the OWL formalism is done using a configuration file, which provides a set of default values to be used for the construction of the ontology. The main script reads the XSD sources and external sources in order to extract the elements necessary to build the ontology. The collector tool requires an Internet connection to retrieve the CISE data model documentation from the EMSA (European Maritime Safety Agency) website. The text extraction is based on the pattern analysis, including the DOM structure of the web page and the paragraph order. Therefore, the extraction result depends on the typography and naming of the titles, subtitles and URLs of the pages. The eCISE data model documentation is not in on-line webpages but in the Andromeda D3.1 deliverable (in PDF format). The tool uses the PDFReader library and performs a page-by-page reading storing the contents of the pages and allowing retrieving the descriptions of the processed elements. The results obtained from processing the previously generated ontology and the retrieval of external Web or PDF sources lead to creating an ontology file containing the RDF triplets of the generated ontology and the associated comments. The scripts are fully available under MIT license[10].

5.2. *Transformation rules*

The transformation rules adopted here take up most of the transformation rules of the Ontmalizer tool [22]. Concerning the association classes and enumerations, the rules follow the proposal from [2]. Table 1 lists the correspondences between XSD elements and OWL definitions as well as elements defined for the treatment of specific types such as enumerations and association classes. Examples of transformation are introduced in the following, for the main OWL constructor types.

Prefix and URI The XSD source files have namespaces by entity group, such as `event`, `vessel`, `object`, and `location`. The choice here was to have the same namespace for all the ontology entities. Furthermore, in order to be fully compliant with the good practices for publishing linked data[11], dereferenced HTTP URIs have been considered as identifiers for the resources (Table 2).

Classes Any class described in the CISE data models, as in Table 1 is a subclass of `ecise:Entity` (subclass of `owl:Thing`). In the example below, the XML and OWL fragments corresponding to the entity `ecise:Vessel` are presented.

```
<xs:complexType name="Vessel">
 <xs:complexContent>
  <xs:extension base="object:Vehicle">
   <xs:sequence>
    [...]
   </xs:sequence>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>

ecise:Vessel
    rdf:type owl:Class ;
    rdfs:subClassOf ecise:Vehicle ;
    rdfs:comment "The class Vessel is a
      sub-class of the class Vehicle. A vessel
      refers to a ship or a boat. [...]" ;
    rdfs:label "Vessel" .
```

Properties Classes are linked to properties (either classes or values). For this type of element, the good naming practices introduced in [23] has been adopted

[10] https://github.com/EFFECTOR-IRIT/XTR
[11] https://www.w3.org/TR/ld-bp/#HTTP-URIS

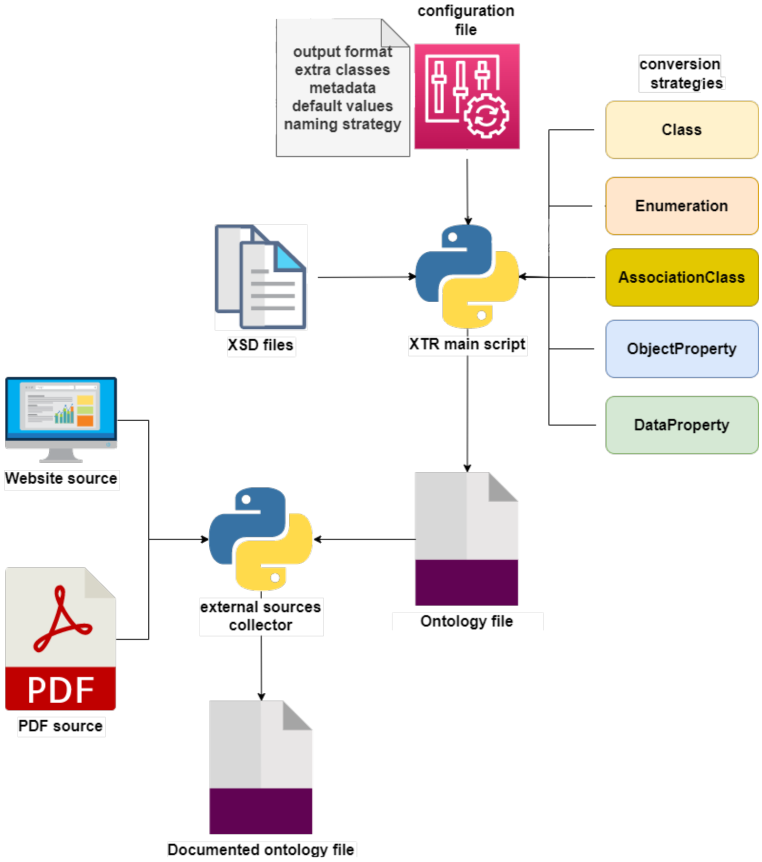

Fig. 3. Diagram of the process for generating OWL CISE and eCISE ontologies from their XSD schemes and other external sources.

XSD element	*OWL definition*
xs:simpleType	rdfs:Datatype
xs:simpleType	rdfs:Datatype
xs:enumeration	owl:Class et owl:Individual
xs:complexType over xs:complexContent	owl:Class
xs:complexType over xs:simpleContent	owl:Class
xs:element (global) with complex type	owl:Class and rdfs:subclassOf
xs:element (global) with simple type	owl:Datatype
xs:element (local to a type)	owl:DatatypeProperty or owl:ObjectProperty
xs:group	owl:Class
xs:attributeGroup	owl:Class

Table 1

Conversion rules from XSD to OWL.

(note the class property names prefixed with 'has'). In the example, the object property `ecise:hasCargo` associates `ecise:Vehicle` and `ecise:Cargo`, according to the extract of the XSD diagram reproduced in Section 3.1.

```
<xs:complexType name="Vehicle" abstract="true">
<xs:complexContent>
 <xs:extension base="object:Object">
  <xs:sequence>
  [...]
  <xs:element name="CargoRel"  minOccurs="0">
   <xs:complexType>
    <xs:complexContent>
```

prefix	namespace
cise	https://w3id.org/cise#
ecise	https://w3id.org/ecise#

Table 2

Ontology prefix and namespace.

```
    <xs:extension base="rel:Relationship">%
   <xs:sequence>
    <xs:element name="Cargo" type="cargo:Cargo"/>
</xs:sequence>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>
  </xs:element>
 </xs:sequence>
</xs:extension>
</xs:complexContent>
```

```
ecise:hasCargo
     rdf:type owl:ObjectProperty ;
     rdfs:domain ecise:Vehicle ;
     rdfs:range ecise:Cargo .
```

Furthermore, as different properties have the same ID (for instance, several occurrences of `hasStatus`, with different domains), their naming has been changed as follows: `concatenation('has', ID domain, ID object property)`.For instance, `ecise:hasRuleStatus`. For sake of traceability, we keep as a `rdfs:label` the original property ID.

Association classes An association class is a 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. Following the approach from [2], association classes are represented as `owl:Class` (not simply as object properties), whereas association roles define their related object properties `owl:ObjectProperty` having as domain the association class. The association classes of the CISE model systematically inherit from the `Relationship` class that has been renamed `ecise:AssociationClass` in the ontology.

```
ecise:AssociationClass a owl:Class ;
  rdfs:comment "Abstract class representing a
      relationship of the CISE data model." ;
  rdfs:label "AssociationClass" .
```

In the XSD fragment below, the entity `Object` has an element referring to the class `Event` and conversely. The XML elements `InvolvedObjectRel` (defined in the XML element Event) and `InvolvedEventRel` defined in the XML element Object) have the same attributes, except for the reference to the associated class (Object for Event and inversely). These two elements are considered to create an association class gathering the attributes common to the two XSD elements.

```
<xs:complexType name="Object" abstract="true">
 <xs:complexContent>
  <xs:extension base="entity:Entity">
   <xs:sequence>
    <xs:element name="InvolvedEventRel"
                minOccurs="0"
                maxOccurs="unbounded">
     <xs:complexType>
      <xs:complexContent>
       <xs:extension base="rel:Relationship">
        <xs:sequence>
         <xs:element name="Event"
              type="event:Event"
              minOccurs="0"/>
         <xs:element name="ObjectRole"
              type="event:ObjectRoleInEventType"
              minOccurs="0"/>
         <xs:element name="InvolvementPeriod"
              type="period:Period"
              minOccurs="0"/>
        </xs:sequence>
       </xs:extension>
      </xs:complexContent>
     </xs:complexType>
    </xs:element>
    [...]
   </xs:sequence>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>
```

```
<xs:complexType name="Event" abstract="true">
 <xs:complexContent>
  <xs:extension base="entity:Entity">
   <xs:sequence>
    <xs:element name="InvolvedObjectRel"
        minOccurs="0" maxOccurs="unbounded">
     <xs:complexType>
      <xs:complexContent>
       <xs:extension base="rel:Relationship">
        <xs:sequence>
         <xs:element name="Object"
              type="object:Object"
              minOccurs="0"/>
         <xs:element name="ObjectRole"
              type="event:ObjectRoleInEventType"
              minOccurs="0"/>
         <xs:element name="InvolvementPeriod"
              type="period:Period"
              minOccurs="0"/>
        </xs:sequence>
       </xs:extension>
      </xs:complexContent>
```

```
      </xs:complexType>
     </xs:element>
     [...]
    </xs:sequence>
   </xs:extension>
  </xs:complexContent>
</xs:complexType>

ecise:ObjectEvent rdf:type owl:Class ;
   rdfs:subClassOf ecise:AssociationClass ;
   rdfs:label "ObjectEvent" .

ecise:hasObject rdf:type owl:ObjectProperty ;
    rdfs:domain ecise:ObjectEvent ;
    rdfs:range ecise:Object ;

ecise:hasEvent rdf:type owl:ObjectProperty ;
    rdfs:domain ecise:ObjectEvent ;
    rdfs:range ecise:Event ;

ecise:hasInvolvementPeriod rdf:type
       owl:ObjectProperty ;
    rdfs:domain ecise:ObjectEvent ;
    rdfs:range ecise:Period ;
```

Enumerations Enumerations in XML define the possible types of specific entities. In OWL, enumerations can be handled as collections of `owl:oneOf`. However, this strategy slows down performance when querying this data. A more performing solution consists in defining a class `ecise:EnumerationType` whose enumerated possible values are instances, as proposed in [2]. Enumerations are then represented as classes (`owl:Class`) which have a predefined list of instances (`owl:NamedIndividual`). For eCISE, there are twenty-eight types of Vessel, including for instance `FishingVessel`, `OilTanker`, `PassengerShip`, `BulkCarrier` or `Special PurposeShip`, as in the example below.

```
<xs:simpleType name="VesselType">
 <xs:restriction base="xs:string">
  [...]
  <xs:enumeration value="FishingVessel"/>
  <xs:enumeration value="OilTanker"/>
  [...]
 </xs:restriction>
</xs:simpleType

ecise:VesselType rdf:type owl:Class ;
    rdfs:subClassOf ecise:EnumerationType ;
    rdfs:label "VesselType" .

ecise:OilTanker a owl:NamedIndividual,
        ecise:VesselType ;
    rdfs:label "OilTanker" .

ecise:FishingVessel a owl:NamedIndividual,
        ecise:VesselType ;
    rdfs:label "FishingVessel" .
```

Constraints The conversion of cardinality constraints from data schemes has raised questions about the relevance of transcribing them into a semantic model managed by the open world assumption. Indeed, because of this assumption, cardinality constraints can only indicate possible maximums or minimums, but cannot constrain multiple cardinality. Another difficulty comes from the default values of these cardinalities (minOccurs and maxOccurs) in XSD, which is 1, whereas it must be made explicit in OWL. Depending on the intended use of the ontology, the constraints are managed by systems manipulating the ontology (e.g. on receipt of a CISE or eCISE message) and before the ontology is used. We propose to add an option to choose whether or not cardinality constraints should be transcribed during ontology generation. These constraints are represented by OWL restrictions on elements of type `owl:ObjectProperty` and `owl:DataProperty`, as in the examples in the following.

```
<xs:complexType name="Vessel">
 <xs:annotation>
  <xs:documentation>The class [...].
  </xs:documentation>
 </xs:annotation>
 <xs:complexContent>
  <xs:extension base="object:Vehicle">
   <xs:sequence>
    <xs:element name="ConditionOfTheCargoAndBallast"
    type="vessel:ConditionOfTheCargoAndBallastType"
    />
    [...]
   </xs:sequence>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>

[  a owl:Restriction ;
  owl:onProperty
  ecise:hasConditionOfTheCargoAndBallast
         MinCardinality ;
  owl:minCardinality 1 .
]
```

Metadata Metadata are essential elements in order to provide additional information promoting ontology reuse. Standard metadata vocabularies have been adopted,to document the ontologies themselves, as presented in the fragment below:

```
[ rdf:type owl:Ontology ;
   dc:creator "Antoine Dupuy",
              "Cassia Trojahn" ,
              "Catherine Comparot" ,
              "Nathalie Aussenac-Gilles" ,
              "Ronan Tournier" ;
```

```
    dc:abstract "A semantic representation the
                eCISE data model." ;
    dc:created "2022-04-12" ;
    dc:licence "https://www.etalab.gouv.fr/
               wp-content/uploads/2018/11/
               open-licence.pdf" ;
    dc:publisher "IRIT laboratory
                  (https://www.irit.fr/)" ;
    dc:title "Extended CISE Ontology"@en ;
    dc:title "eCISE-OWL" ;
    dc:language "en" ;
    dc:description "Generated from e-CISE 2.4.0
                    model."@en ;
    dc:source "https://www.andromeda-project.eu/
               ecise/" ;
    owl:priorVersion "1.0.0" ;
    owl:versionInfo "1.0.0" ;
    owl:versionIRI <https://www.w3id.org/
                    ecise>
] .
```

Versioning A versioning strategy has also been adopted as further version of the CISE model are expected. The version is composed of three numbers (e.g. "1.0.0"): a major version (for a cut of the backward compatibility), a minor version (for an addition of features), and a patch (for a bug fix). It is important to note that the version of the underlying CISE data model and the version of the generated ontology itself are distinguished.

5.3. *Manual validation*

The generation of the ontologies from the CISE models requires the ontologies to be compliant to their specifications. In order to do so, it was necessary to validate the transformation of the model properties (generated entities vs. model specifications of the UML diagrams as described in the technical documents, i.e., the 1.5.3 CISE model specification and the ANDROMEDA project deliverable [1] for CISE-OWL and eCISE-OWL, respectively). This operation was manually performed by 3 experts, with the help of the Protege software. For each type of entity, its label, its class, associated annotations (labels and comments) and axioms were verified. This verification and correction work required several workshops and took place over about twenty hours.

During the validation process, the inconsistencies identified involved exclusively i) the names of the association classes and ii) the removal of duplicated entities associated to association classes. With respect to i), as several names of association classes do not correspond to the specification, we decided to create a renaming script. This script takes an input list containing the correct names, manually defined, and renames the corresponding classes. With respect to ii), association classes elements in XSD files (as for the example above about the entity `Object`) have elements referring to each other. As in the example, the elements `InvolvedObjectRel` (element of Event) and `InvolvedEventRel` (element of Object) have the same attributes, except for the reference to the associated class (Object for Event and inversely). During the conversion process, two association classes are created relative to the two different references: an `InvolvedObjectRel` class and an `InvolvedEventRel`. These two classes represent the same class in the CISE/e-CISE data models.

6. CISE-OWL and eCISE-OWL

Table 3 presents the main metrics from the generated ontologies. The CISE-OWL ontology contains a total of 151 classes, 160 object properties and 135 datatype properties, while eCISE-OWL represents a total of 270 classes, 344 object properties and 304 datatype properties. In eCISE-OWL, the enumerations represent the majority of conversions to be processed. The association classes represent a total of 31 classes out of the 270 of the ontology. These classes are populated by 16170 individuals. Figures 4 and 5 present the hierarchy of the ontology's main concepts of CISE-OWL and eCISE-OWL, respectively. The two ontologies are available online with permanent identifiers[12] .

6.1. *Ontology evaluation*

The generated ontologies have been evaluated with different metrics. In a first evaluation, the OOPS! (OntOlogy Pitfall Scanner!) tool [24][13] has been used to evaluate their modeling quality. This tool identifies modeling errors according to the structural, functional and profiling dimensions of usability. It also provides an indicator (critical, important, minor) for each identified pitfall, according to the respective index. No pitfalls were detected in the structural, functional, consistency, completeness, and conciseness dimensions.

[12]https://w3id.org/cise and https://w3id.org/ecise

[13]http://oops.linkeddata.es/catalogue.jsp (consulted on 20/04/2022)

Metric	*CISE-OWL*	*eCISE-OWL*	*EUCISE-OWL*
Number of classes	151	270	153
Number of objects properties	160	344	127
Object property - number of domain axioms	160	344	116
Object property - number of range axioms	160	344	116
Number of data properties	135	304	135
Data property - number of domain axioms	137	350	132
Data property - number of range axioms	135	305	132
Number of individuals	15773	16170	869
DL expressivity	SHIF(D)	SHIF(D)	SHIF(D)
Number of triples	65602	69929	6,209
Number of association classes	11	31	10
Number of enumeration classes	87	141	

Table 3

Ontology metrics for CISE-OWL, eCISE-OWL and EUCISE-OWL [2]

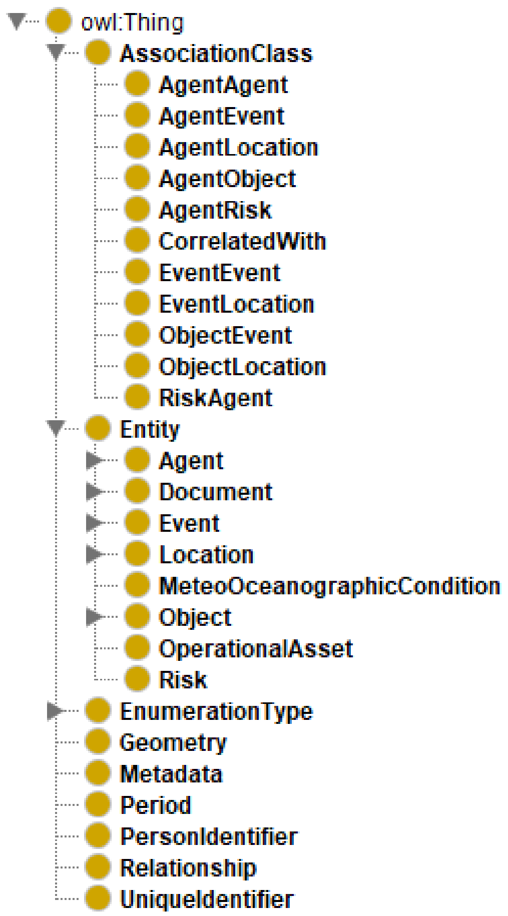

Fig. 4. CISE-OWL hierarchy

To assess the model compliance to the FAIR principles, the FOOPS [25] online tool has been used. It takes as input an OWL ontology and runs 24 different checks distributed across the 4 FAIR dimensions : 9 checks on F (unique, persistent and resolvable URI and version IRI, minimum descriptive metadata, namespace and prefix found in external registries); 3 checks on A (content negotiation, serialization in RDF, open URI protocol); 3 checks on I (references to pre-existing vocabularies); and 9 checks on R (human-readable documentation, provenance metadata, license, ontology terms properly described with labels and definitions). Following these criteria, a score of 79% of FAIRness is obtained for both *CISE-OWL* and *e-CISE-OWL*, thanks to their documentation using rich metadata and their publication with permanent IDs. This score can be further improved by indexing the models in a searchable resource (LOV, for instance).

6.2. Ontology alignments

In order to foster semantic interoperability, a set of ontology correspondences (i.e., an alignment) has been generated between the CISE-OWL, eCISE-OWL and well-known existing ones, involving class entities. Concerning CISE-OWL, the correspondences established in [2] for EUCISE-OWL have been reused. They are mostly expressed via `rdfs:subClassOf` and `rdfs:subPropertyOf` in a way the entities inherit all involved semantics declared in the adopted ontologies. It is assumed that the version 1.0 upon which EUCISE-OWL has been generated is the closer version of the CISE version 1.5.3 adopted here. The proposed correspondences have been re-evaluated and some of then have been discarded.

Out of the 29 correspondences in [2], 10 correspondences have been manually revised: i) 7 correspon-

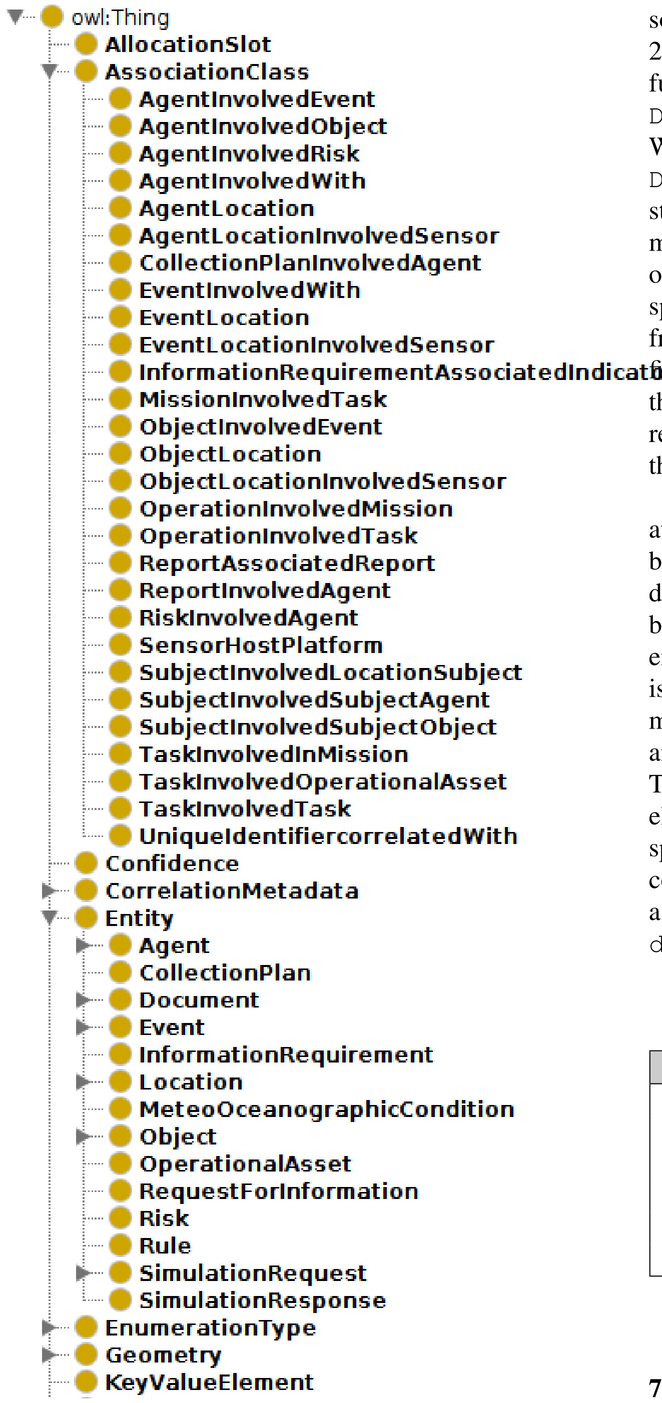

Fig. 5. eCISE-OWL hierarchy

dences involving enumerations types have been discarded (e.g., `cise:AgentRoleInAgentType` $\sqsubseteq$ `prov:Role` and `cise:PlannedOperationsType` $\sqsubseteq$ `prov:Plan`); ii) 1 correspondence involving an association class (`cise:AgentEvent` $\sqsubseteq$ `prov:Association`); 2 correspondences have also been evaluated as not fully correct: `cise:Period` $\sqsubseteq$ `time:GeneralDateTimeDescription` and `Entity` $\sqsubseteq$ `geosparql:Feature`). While a `Period` defines a time interval, `time:GeneralDateTimeDescription`, describes date and time structured with separate values for the various elements of a calendar-clock system without information on the interval. With to respect to the second correspondence, as `Person` and `Organization` inherit from `cise:Entity` by transitivity, one can not affirm that a `cise:Person` is a `geosparql:Feature` that has an associated geometry. The reminder 19 correspondences involving 9 entities have been kept (as the alignment cardinality is 1-N).

Concerning eCISE-OWL, entities involving association classes (35) and enumeration types (136) have been discarded. The manually generated correspondences entities are listed in Table 5. The process has been carried out by two experts that compared the entity definitions with those of entities from the existing ontologies involved in EUCISE-OWL alignments (e.g., SOSA and DUL) and from ontology and alignment repositories as LOV[14], BioPortal[15]. These correspondences involve mostly the first levels of the eCISE-hierarchy and added entities with respect to CISE. From these correspondences, several correspondences can be derived by transitivity (e.g., a `Vessel` is a subclass of `envo:Vehicule` and `dul:physical-endurant`).

Table 4

Prefix of the aligned ontologies.

prefix	namespace
sweet	http://sweetontology.net#
prov	http://www.w3.org/ns/prov#
sosa	http://www.w3.org/ns/sosa/
dul	http://www.loa-cnr.it/ontologies/DOLCE-Lite.owl#
foaf	http://xmlns.com/foaf/0.1/
envo	http://purl.obolibrary.org/obo/envo.owl#

7. Conclusions and future work

This paper presented two maritime ontologies derived from the CISE data model. The process used to generated them is presented, which involved two main

[14] https://lov.linkeddata.es/dataset/lov/
[15] https://bioportal.bioontology.org/

eCISE-OWL	*Existing ontology*
Animal	sweet:Animal
CollectionPlan	prov:Plan
CrisisIncident	sosa:Observable
Location	prov:Location
Metadata	sweet:Metadata
MeteoOceanographicCondition	sosa:ObservableProperty
Mission	dul:Event
Object	dul:physical-endurant
Operation	dul:Event
Report	foaf:Document
Sensor	sosa:Sensor
SensorMetadata	sweet:Metadata
Vehicle	envo:Vehicule

Table 5

Alignments between eCISE-OWL and existing ontologies.

steps: i) automatic conversion of XSD files into OWL, and ii) manual validation by domain experts. This is the first attempt to enhance the CISE data model using a semantic approach, made public and available to all.

In the future, this work will be continued along several lines. A first set of perspectives aims at improving the process, described in this paper, of ontology generation from an XSD source: improve the extraction of terminologies from external sources because for some association classes and for some enumeration values, the extraction of terms from the tables present in the source files requires the use of more sophisticated information extraction tools than the ones chosen here. Second, ontology matching involving automatic selection and matching of multiple ontologies together with the exposition of FAIR [26] is a plan. Finally, the ontology will be used within the information system being developed in the EFFECTOR project, which will require the implementation of a process for converting CISE messages into eCISE messages through their representation in RDF.

Acknowledgements

This research was conducted with the help of the EFFECTOR project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No.883374. This material reflects only the author's view and the Research Executive Agency (REA) and European Commission cannot be held responsible for any use that may be made of the information it contains.

References

[1] S. Antonopoulos, M. Tsogas, M. Moutzouris, A. Kostaridis, A. Aggelis and L. Perlepes, Andromeda D.3.1 e-CISE Data Model description, Andromeda Project (n. 833881) Deliverable, EU, 2020. https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5ce61d7a4&appId=PPGMS.

[2] M. Riga, E. Kontopoulos, K. Ioannidis, S. Kintzios, S. Vrochidis and I. Kompatsiaris, EUCISE-OWL: An ontology-based representation of the Common Information Sharing Environment (CISE) for the maritime domain, *Semantic Web* **12**(4) (2021), 603–615. doi:10.3233/SW-200403.

[3] D. Berger, J. Hermida, F. Oliveri and G. Pace, The Entity Service Model for CISE–Service Model Specifications, Technical Report, Joint Research Centre of the European Commission, 2017.

[4] A. Minutolo, A. Esposito, M. Ciampi, M. Esposito and G. Cassetti, An Automatic Method for Deriving OWL Ontologies from XML Documents, in: *2014 Ninth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, Guangdong, China, November 8-10, 2014*, IEEE Computer Society, 2014, pp. 426–431. doi:10.1109/3PGCIC.2014.88.

[5] M. Hacherouf, S.N. Bahloul and C. Cruz, Transforming XML schemas into OWL ontologies using formal concept analysis, *Software & Systems Modeling* **18** (2017), 2093–2110.

[6] I. Bedini, C. Matheus, P. Patel-Schneider, A. Boran and B. Nguyen, Transforming XML Schema to OWL Using Patterns, in: *2011 IEEE Fifth International Conference on Semantic Computing*, 2011, pp. 102–109.

[7] D. Vinasco-Alvarez, J.S. Samuel, S. Servigne and G. Gesquière, From CityGML to OWL, Technical Report, LIRIS UMR 5205, 2020. https://hal.archives-ouvertes.fr/hal-02948955.

[8] M. Wilkinson, M. Dumontier and e. Aalbersberg, The FAIR Guiding Principles for scientific data management and stewardship, *Scientific data* **3**(1) (2016), 1–9.

[9] G.M. Santipantakis, G.A. Vouros, A. Glenis, C. Doulkeridis and A. Vlachou, The datAcron Ontology for Semantic Trajectories, in: *The Semantic Web: ESWC 2017 Satellite Events*, E. Blomqvist, K. Hose, H. Paulheim, A. Ławrynowicz, F. Ciravegna and O. Hartig, eds, Springer International Publishing, Cham, 2017, pp. 26–30. ISBN 978-3-319-70407-4.

[10] G.K.D. de Vries, V. Malaisé, M. van Someren, P. Adriaans and A.T. Schreiber, Semi-Automatic Ontology Extension in the Maritime Domain, in: *In Proceedings of the 20th Belgian-Netherlands Conference on Artificial Intelligence (BNAIC 2008)*, 2008, DeVries:BNAIC2008 University of Twente, Enschede, the Netherlands.

[11] S. Brüggemann, K. Bereta, G. Xiao and M. Koubarakis, Ontology-Based Data Access for Maritime Security, in: *Proceedings of the 13th International Conference on The Semantic Web. Latest Advances and New Domains - Volume 9678*, Springer-Verlag, Berlin, Heidelberg, 2016, pp. 741–757–. ISBN 9783319341286.

[12] A. Vandecasteele and A. Napoli, An enhanced spatial reasoning ontology for maritime anomaly detection, in: *2012 7th International Conference on System of Systems Engineering (SoSE)*, 2012, pp. 1–6. doi:10.1109/SYSoSE.2012.6384120.

[13] A. Vandecasteele and A. Napoli, Spatial ontologies for detecting abnormal maritime behaviour, in: *2012 Oceans - Yeosu*, 2012, pp. 1–7. doi:10.1109/OCEANS-Yeosu.2012.6263532.

[14] D.X. Zheng and X.D. Sun, A knowledge acquisition model in maritime domain based on ontology, in: *Proceedings IEEE International Conference on Security, Pattern Analysis, and Cybernetics, SPAC 2014, Wuhan, China, October 18-19, 2014*, IEEE, 2014, pp. 372–375. doi:10.1109/SPAC.2014.6982718.

[15] M. Hagaseth, L. Lohrmann, A. Ruiz, F. Oikonomou, D. Roythorne and S. Rayot, An Ontology for Digital Maritime Regulations, *Journal of Maritime Research* **XIII**(II) (2016), 7–18–.

[16] N. Aussenac-Gilles, M. Kamel, D. Buscaldi and C. Comparot, Construction semi-automatique d'ontologies à partir d'une collection de pages web structurées, in: *IC 2013 : 24es Journées francophones d'Ingénierie des Connaissances (Proceedings of the 24th French Knowledge Engineering Conference), Lille, France, July 1-5, 2013*, R. Troncy, ed., 2013, pp. 165–180. https://hal.archives-ouvertes.fr/IC_2013/hal-00854428.

[17] M. Frosterus, E. Hyvönen and J. Laitio, DataFinland - A Semantic Portal for Open and Linked Datasets, in: *8th Extended Semantic Web Conference, ESWC, Heraklion, Crete, Greece*, G. Antoniou, M. Grobelnik, E.P.B. Simperl, B. Parsia, D. Plexousakis, P.D. Leenheer and J.Z. Pan, eds, Lecture Notes in Computer Science, Vol. 6644, Springer, 2011, pp. 243–254.

[18] D. Calvanese, B. Cogrel, S. Komla-Ebri, R. Kontchakov, D. Lanti, M. Rezk, M. Rodriguez-Muro and G. Xiao, Ontop: Answering SPARQL queries over relational databases, *Semantic Web* **8**(3) (2017), 471–487. doi:10.3233/SW-160217.

[19] M.C. Suárez-Figueroa, A. Gómez-Pérez and M. Fernández-López, The NeOn Methodology framework: A scenario-based methodology for ontology development, *Appl. Ontology* **10**(2) (2015), 107–145. doi:10.3233/AO-150145.

[20] B. Villazón-Terrazas and A. Gómez-Pérez, Reusing and Re-engineering Non-ontological Resources for Building Ontologies, in: *Ontology Engineering in a Networked World*, M.C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta and A. Gangemi, eds, Springer, 2012, pp. 107–145. doi:10.1007/978-3-642-24794-1_6.

[21] P.R. Aryan, Converting XML Schema to OWL in Python, 2013. https://pebbie.wordpress.com/2013/06/04/converting-xml-schema-to-owl-in-python/.

[22] M. Yüksel, A semantic interoperability framework for reinforcing post market safety studies, Technical Report, Middle East Technical University, 2013.

[23] V. Svátek and O. Sváb-Zamazal, Entity Naming in Semantic Web Ontologies : Design Patterns and Empirical Observations, in: *9th Czecho-Slovak Knowledge Engineering Conference*, 2009.

[24] M. Poveda-Villalón, A. Gómez-Pérez and 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.

[25] D. Garijo, Ó. Corcho and M. Poveda-Villalón, FOOPS!: An Ontology Pitfall Scanner for the FAIR principles, in: *Proceedings of the ISWC 2021 Posters, Demos and Industry Tracks: From Novel Ideas to Industrial Practice co-located with 20th International Semantic Web Conference (ISWC 2021), Virtual Conference, October 24-28, 2021*, O. Seneviratne, C. Pesquita, J. Sequeda and L. Etcheverry, eds, CEUR Workshop Proceedings, Vol. 2980, CEUR-WS.org, 2021. http://ceur-ws.org/Vol-2980/paper321.pdf.

[26] N. Matentzoglu, J.P. Balhoff, S.M. Bello, C. Bizon, M. Brush, T.J. Callahan, C.G. Chute, W.D. Duncan, C.T. Evelo, D. Gabriel, J. Graybeal, A. Gray, B.M. Gyori, M. Haendel, H. Harmse, N.L. Harris, I. Harrow, H. Hegde, A.L. Hoyt, C.T. Hoyt, D. Jiao, E. Jiménez-Ruiz, S. Jupp, H. Kim, S. Koehler, T. Liener, Q. Long, J. Malone, J.A. McLaughlin, J.A. McMurry, S. Moxon, M.C. Munoz-Torres, D. Osumi-Sutherland, J.A. Overton, B. Peters, T. Putman, N. Queralt-Rosinach, K. Shefchek, H. Solbrig, A. Thessen, T. Tudorache, N. Vasilevsky, A.H. Wagner and C.J. Mungall, A Simple Standard for Sharing Ontological Mappings (SSSOM), arXiv, 2021. doi:10.48550/ARXIV.2112.07051. https://arxiv.org/abs/2112.07051.

[27] SprintProject, D4.2 A lightweight solution to automate the generation of ontologies, mappings and annotations (C-REL), Project Deliverable, D4.2, EU, 2020. http://www.sprint-transport.eu/download.aspx?id=634cb6ad-e478-4a46-b181-b4f523a23d2a.

[28] EUCISE2020, EUCISE2020: Technical Specifications. Deliverable 4.3, Revision 1, Annex B, Technical Report, EUCISE Data Model, 2020.

[29] C. Iphar, Formalisation of a data analysis environment based on anomaly detection for risk assessment : Application to Maritime Domain Awareness, Theses, Université Paris sciences et lettres, 2017. https://pastel.archives-ouvertes.fr/tel-01783958.

[30] A. Vandecasteele, R. Devillers and A. Napoli, A semi-supervised learning framework based on spatio-temporal semantic events for maritime anomaly detection and behavior analysis, in: *CoastGIS 2013 - The 11th International Symposium for GIS and Computer Cartography for Coastal Zone Management*, Victoria, Canada, 2013, p. 4 pages. https://hal-mines-paristech.archives-ouvertes.fr/hal-00850278.

[31] B. Lantow, OntoMetrics: Putting Metrics into Use for Ontology Evaluation, in: *Proceedings of the 8th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - KEOD, (IC3K 2016)*, SciTePress, 2016, pp. 186–191, INSTICC. ISSN 2184-3228. ISBN 978-989-758-203-5. doi:10.5220/0006084601860191.

[32] E. Amdouni, S. Bouazzouni and C. Jonquet, O'FAIRe: Ontology FAIRness Evaluator in the AgroPortal semantic resource repository, in: *ESWC 2022 - 19th Extended Semantic Web Conference, Poster and demonstration.*, Hersonissos, Greece, 2022. https://hal-lirmm.ccsd.cnrs.fr/lirmm-03630543.