Sémantisation du Modèle e-CISE pour l'Interoperabilité de Données Maritimes Géolocalisées

Nathalie Aussenac-Gilles, Catherine Comparot, Antoine Dupuy, Ronan Tournier, Ba-Huy Tran, Cassia Trojahn dos Santos

Sémantisation du Modèle e-CISE pour l'Interoperabilité de Données Maritimes Géolocalisées

Nathalie Aussenac-Gilles, Catherine Comparot, Antoine Dupuy, Ronan Tournier, Ba-Huy Tran, Cassia Trojahn

IRIT, Université de Toulouse, CNRS, Toulouse INP, UT3, UT1, UT2, Toulouse, France
prenom.nom ou prenom.nom-composé@irit.fr

Résumé. Une étroite coopération transfrontalière est essentielle pour la surveillance des frontières, et en particulier lors de crises maritimes, ce qui nécessite un partage d'informations transfrontalier. Le développement du *Common Information Sharing Environment* (CISE) constitue une initiative collaborative pour promouvoir le partage automatisé d'informations entre les autorités de surveillance maritime. L'adoption de CISE et de ses différentes versions est cependant limitée par sa sérialisation existante via uniquement un schéma XML. Contrairement aux ontologies, ce format est limité pour fournir une sémantique plus riche, une interopérabilité sémantique et des capacités de raisonnement sémantique. Nous présentons ici le processus de conversion de la version plus récente du schéma CISE (e-CISE) dans le format standard Ontology Web Langage (OWL), contribuant ainsi à améliorer les travaux antérieurs sur la transformation de schémas XML (XSD) en ontologies OWL.

1 Introduction

Rendre les systèmes de surveillance maritime interopérables est crucial pour la coopération entre les pays, en particulier en cas de crises maritimes dans des zones frontalières entre pays. Dans cet objectif, l'hétérogénéité entre les système nationaux et les structures de données des différents acteurs soulèvent de nombreux problèmes. En outre, la plupart des données partagées dans cette infrastructure ont une composante spatio-temporelle.

Afin de permettre aux autorités d'échanger des informations de manière automatique et sécurisée, l'environnement commun de partage d'informations (CISE) [1] a été proposé. Il fournit un cadre décentralisé et un modèle de données pour l'échange d'informations point à point entre les secteurs et les frontières. Il implique plus de 300 autorités européennes et nationales ayant des responsabilités en matière de surveillance maritime, effectuant de nombreuses tâches de surveillance opérationnelle différentes. Les autorités nationales bénéficient directement d'être connectées au réseau CISE, dans divers secteurs tels que la sûreté et la sécurité du transport maritime, le contrôle de la pêche, la pollution, et la défense. Depuis 2014, CISE est l'une des actions soutenant la mise en œuvre de la stratégie de sécurité maritime de l'Union Européenne (EUMSS).

1. `http://www.emsa.europa.eu/cise.html`

L'adoption du modèle de données CISE [2] et de ses différentes versions – en particulier, *Extended-CISE* (e-CISE) (Antonopoulos et al., 2020) – est cependant limitée par sa sérialisation existante en XML via un schéma XML uniquement, qui ne peut ni fournir une sémantique riche, ni garantir une interopérabilité sémantique des données ni fournir de support à un raisonnement. Or ces fonctionnalités s'avèrent très utiles pour associer des données venant de différentes sources de données CISE, les vérifier ou encore en déduire de nouveaux éléments. Un premier effort dans cette direction était la représentation ontologique du modèle de données CISE proposée dans (Riga et al., 2021). Cependant, la ressource et son processus de construction ne sont pas publiquement disponibles.

De manière pratique, le passage de données XML ou de schémas XSD vers une représentation sémantique, que ce soit en RDF, RDFS ou en OWL, est une question étudiée de longue date dans le domaine du Web sémantique. Cependant, une transformation automatique simple est reconnue rarement correcte. Ce passage se heurte à la difficulté de gérer des noeuds anonymes, de traiter la représentation de noeuds complexes, de capturer la sémantique des balises purement structurelles, ou encore de produire des constructeurs liés à la structuration (Minutolo et al., 2014; Hacherouf et al., 2017; Bedini et al., 2011; Vinasco-Alvarez et al., 2020).

Dans cet article, nous présentons un processus de conversion du modèle e-CISE en Ontology Web Langage (OWL). e-CISE étend le modèle CISE en améliorant le vocabulaire maritime de CISE et en étendant sa portée à la surveillance terrestre et à l'échange d'informations opérationnelles. Notre approche contribue à améliorer les travaux antérieurs sur la transformation de schémas XSD en ontologies OWL. Elle est basée sur l'extension et la combinaison d'outils existants. A l'issue de ce travail, une deuxième contribution sera de mettre à disposition le schéma e-CISE sous forme d'ontologie. A moyen terme, nous prévoyons ensuite de définir un processus d'instanciation de cette ontologie à partir du contenu de messages e-CISE. Ce travail est réalisé dans le cadre du projet H2020 EFFECTOR [3]. Ce projet propose un cadre d'interopérabilité et des services de fusion et d'analyse de données associés pour la surveillance maritime et la sécurité des frontières. Ainsi, EFFECTOR vise à améliorer le processus d'aide à la décision et de favoriser la collaboration des acteurs maritimes au niveau local, régional et transnational. Le modèle de données CISE joue un rôle essentiel dans le projet car les messages échangés par les différents acteurs sont basés sur les différentes versions de ce modèle. Une représentation sémantique et des alignements entre ces différentes versions contribue donc à l'enjeu de l'interoperabilité.

Le reste de l'article est organisé comme suit : la section 2 introduit le modèle CISE, ses variantes et son extension e-CISE. La section 3 présente les principaux travaux en lien avec la génération de fichiers OWL à partir des fichiers XML/XSD. La Section 4 décrit le processus de conversion implémenté pendant que la Section 4.4 discute les difficultés rencontrées et finalement la Section 5 conclut l'article et dessine les perspectives pour les travaux futurs.

2 Schémas de données CISE et e-CISE

Le modèle de données CISE a pour ambition de servir de format pour le partage d'information de surveillance maritime entre secteurs et pays. Dans cette optique, ce modèle de données décrit sept entités principales (`Agent`, `Object`, `Location`, `Document`, `Event`, `Risk`,

2. `http://emsa.europa.eu/cise-documentation/cise-data-model-1.5.3/`
3. `https://www.effector-project.eu/`

`Period`) et onze entités auxiliaires (`Vessel`, `Cargo`, `Operational Asset`, `Person`, `Organization`, `Movement`, `Incident`, `Anomaly`, `Action`, `Unique Identifier`, `Metadata`). Le modèle CISE permet aux différentes autorités de bénéficier d'un vocabulaire commun pour décrire les événements observés. Ce modèle est décliné dans les formalismes RDF et XSD. Sur la Figure 1, les entités du modèle CISE correspondent aux hexagones non colorés. Le modèle de données e-CISE enrichit le vocabulaire du modèle de données CISE concernant les domaines maritime et terrestre en introduisant de nouvelles associations d'entités, des capacités et un ensemble plus riche de types dans plusieurs énumérations. Entre autres, e-CISE fournit un ensemble plus riche de types de navires, d'Automatic Identification System (AIS) et de capteurs radar ; il liste également un ensemble plus complet d'anomalies et de règles maritimes, mais aussi terrestres ; il offre enfin de nouvelles capacités de détection des entités. Ce modèle est construit sur la dernière version du modèle de données CISE utilisé dans le projet EUCISE2020 (EUCISE2020, 2020). Sur la Figure 1, les entités centrales du modèle e-CISE correspondent aux hexagones de couleur, qui viennent compléter les entités du modèle CISE.

FIG. 1 – *Aperçu des vocabulaires du modèle de base CISE (entités en blanc) et du modèle e-CISE (CISE enrichi par les entités en couleur).*

Une composante clé du modèle CISE (et donc aussi de e-CISE) concerne la dimension spatiale. Pour ce faire, l'entité `Location`, une sous-classe de `Entity` est centrale. Une lo-

calisation peut être décrite de trois manières principales : en utilisant un nom de lieu, une géométrie ou une adresse. Le contexte spécifique déterminera quelle méthode de description d'un emplacement est la plus appropriée. L'ISO 19112 définit un emplacement comme un lieu géographique identifiable. Dans cet esprit, "Pont Neuf" et "Toulouse" sont deux emplacements identifiés sous forme de chaînes de caractères. Ce type d'identifiant est courant bien qu'il puisse être très ambigu car de nombreux endroits partagent des noms identiques ou similaires. Un emplacement est décrit par différentes propriétés et attributs, tels qu'une `Geometry` and `LocationZone`. En particulier, la propriété `Geometry` permet la définition d'une région geo-réferencée, ayant comme attributs `altitude`, `latitude`, `longitude`, `WKT` et `XMLGeometry`.

Les modèles CISE et e-CISE sont décrits dans un document de spécification (diagrammes de classes UML) et implémentés en XSD (schéma XML). Les fichiers XSD de CISE ont été produits à partir de transformations, i.e. un ensemble de règles de correspondance indiquant comment générer des éléments XSD à partir des éléments UML correspondants. Les choix effectués lors de ce processus impactent la structure XSD obtenue et doivent être pris en compte lors de la génération de sa représentation en OWL. Le choix de la transformation en XSD des classes d'association UML pour e-CISE (1 fichier XSD par classe) fait qu'une classe d'association est décrite 2 fois (une description par classe participant à l'association).

3 De XSD à RDF/OWL : approches existantes

Le passage de données XML ou de schémas XSD vers une représentation sémantique, que ce soit en RDF ou en OWL, est une question étudiée de longue date dans le domaine du Web sémantique. En effet, nombre de ressources du web sont disponibles sous ce format semi-structuré. De plus, il existe une sérialisation XML de RDF, sous forme de langage de balises. Lorsqu'un schéma XSD est disponible et qu'une collection de documents a été construite en conformité à ce schéma, il est tentant de considérer les balises XML comme définissant les classes d'une ontologie ou le type d'entités en RDF. La conversion des documents XML décrits selon un schéma peut être alors traitée comme la définition d'instances de ces classes, ce qui revient à peupler l'ontologie définie par le schéma. Pour gérer ce processus, si l'on reste dans un format XML, on peut considérer que le problème technique est le passage d'un schéma XML à un autre. Pour cela, il existe le langage XSLT[4], boîte à outils naturelle pour convertir un schémas XML vers un autre schéma XML.

Cependant, les balises ne se situant pas toutes au même niveau d'abstraction, une transformation automatique simple est rarement efficace et correcte. Lorsque les éléments définis entre balises sont eux mêmes complexes, et relèvent de plusieurs types en lien avec plusieurs propriétés, ils peuvent même contribuer à la fois à enrichir l'ontologie et à la peupler. Dans tous les cas, le passage du XSD à des classes OWL ou des types RDF se heurte à la difficulté de gérer des noeuds anonymes, de traiter la représentation de noeuds complexes, la génération d'énumérations, la gestion des types XSD, ou encore de représenter les balises purement structurelles et non sémantique, ou encore de produire des balises liées à la structuration.

Plusieurs outils ont été proposés dans l'état de l'art. Parmi eux, les outil dit de "lifting" convertissent un schéma XML (XSD) en un schéma RDF, tel que RDFS ou OWL. Un tel

4. https ://www.w3.org/community/rax/wiki/XML_to_RDF_Transformation_processes_using_XSLT#Preprocessing_of_XML

outil est XML2OWL (à partir d'un document d'instance XML ou d'un XSD), ou XSD2OWL [5] (à partir d'un XSD). Une autre option consiste à utiliser TopBraid Composer, et un plugin XSD vers OWL (notez toutefois qu'il s'agit d'un logiciel commercial). MIT Simile fournit une liste de quelques autres "RDF-izers". La Table 1 présente une synthèse des outils étudiés. Malheureusement, les outils présentés dans les articles de recherche (comme les deux que nous venons de citer) soient rarement accessibles. Il est encore regrettable que les deux autres outils génèrent des ontologies très complexes et négligent la représentation des relations n-aires entre classes. Il est aussi souhaitable d'avoir un mécanisme pour la construction automatique des descriptions des éléments (via `rdfs:comment` par exemple) à partir de la documentation. Pour répondre à ces besoins, nous avons développé un processus implémenté en Python. Il s'inspire du travail de (Aryan, 2013) et reprend en les étendant les règles de transformation telles que définies par Ontmalizer [6].

Outil	**Langage**	**Avantages**	**Inconvénients**
PIXCO (Hacherouf et al., 2017)	Java		Inaccessibilité
JANUS (Bedini et al., 2011)	Java	GUI	Inaccessibilité
CityGML (Vinasco-Alvarez et al., 2020)	Python et Java	Simplicité	Résultat complexe
Ontmalizer (Yüksel, 2013)	Java	Simplicité	Résultat complexe
XSD2RDF.py (Aryan, 2013)	Python	Simplicité	Incapacité de traiter les relations n-aires et des individus

TAB. 1 – *Comparaison des outil de conversion de XML en RDF/OWL.*

4 Processus de conversion XML en OWL

Dans ce qui suit, nous présentons un schéma du processus de conversion de XSD en OWL ainsi que les règles de transformation que nous considérons et des fragments de l'ontologie générée à partir des spécifications du schéma de e-CISE.

4.1 Schéma du processus

La Figure 2 illustre le schéma du processus de conversion XSD en OWL. Ce processus est implémenté en Python et fait appel à plusieurs sources de données externes, comme décrit ci-dessous.

Le pilotage de l'outil de transformation des données au format XSD vers le formalisme OWL se fait via l'utilisation d'un fichier Makefile. Ce fichier permet de lancement via une console des commandes de génération des ontologies CISE et e-CISE (`make cise` et `make ecise`). Le lancement d'une commande lance la génération de l'ontologie souhaitée (CISE ou e-CISE dans la version actuelle du programme). Le script correspondant effectue la lecture des sources XSD et des sources externes afin d'extraire les éléments nécessaires à la construction

5. https ://gist.github.com/pebbie/5704765
6. `https://www.w3.org/wiki/HCLSIG/Tools\#Ontmalizer` (Yüksel, 2013)

de l'ontologie. Pour chaque type d'élément XSD, une règle de conversion est appliquée vers le formalisme OWL.

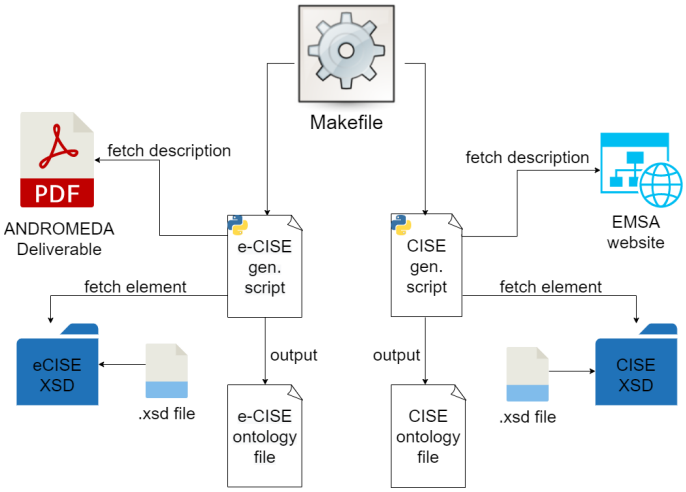

FIG. 2 – *Schéma du processus de génération des modèles OWL e-CISE et CISE à partir de leurs schémas XSD et d'autres sources externes.*

L'outil de transformation nécessite une connexion Internet pour récupérer la documentation du modèle de données CISE sur le site de l'EMSA (*European Maritime Safety Agency*). L'extraction du texte est basée sur l'analyse des patterns, incluant la structure DOM de la page web et l'ordre des paragraphes. Par conséquent, le résultat de l'extraction dépend de la typographie et de la dénomination des titres, sous-titres et de l'URL des pages. Le livrable D3.1 d'Andromeda (Antonopoulos et al., 2020) (au format PDF) est quant à lui nécessaire pour la documentation du modèle de données e-CISE. L'outil utilise la librairie PDFReader et effectue une lecture page à page ; il stocke le contenu des pages et permet de retrouver les descriptions des éléments traités. Les résultats obtenus suite au traitement des éléments XSD et à la lecture des sources Web ou PDF conduisent à la création d'un fichier au format turtle contenant les triplets RDF de l'ontologie générée.

4.2 Règles de transformation

La plupart des règles de transformation ont été inspirées de celles de l'outil Ontmalizer qui a été réutilisé aussi dans le projet Shift2Rail[7] (SprintProject, 2020). Concernant les classes d'association et les énumérations, nous avons repris la proposition de (Riga et al., 2021). Plus précisément, les classes d'association sont représentées comme des sous-classes d'une nouvelle classe appelée `AssociationClass`. Pour les énumérations, une nouvelle classe `EnumerationType` est également introduite, et les valeurs des énumérations sont des instances (individus) de cette classe. Des exemples sont donnés en Section 4.3.1.

7. https ://shift2rail.org/

Element XSD	Définitions OWL
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, and OWL Restrictions
xs :group	owl :Class
xs :attributeGroup	owl :Class

TAB. 2 – *Règles de conversion XSD en OWL.*

4.3 e-CISE en OWL

La Figure 3 présente un extrait de l'ontologie générée, où les classes centrales du modèle e-CISE (Figure 1) sont représentées. La Figure 4 présente un extrait de l'ontologie obtenue visualisée à l'aide du logiciel GraphBD. L'extrait concerne l'entité `Location`, une des entités centrales du modèle représentant la dimension spatiale.

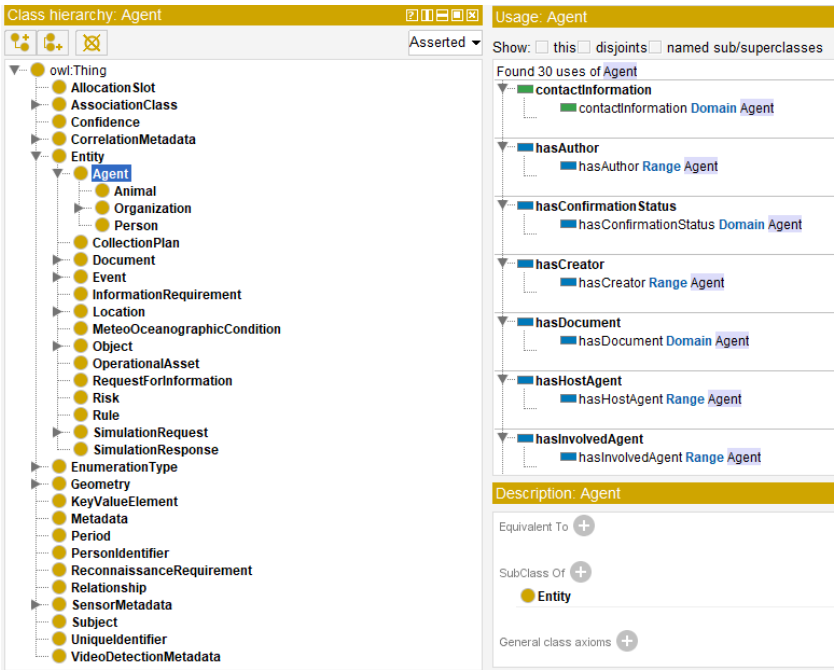

FIG. 3 – *Visualisation de l'ontologie sur Protégé.*

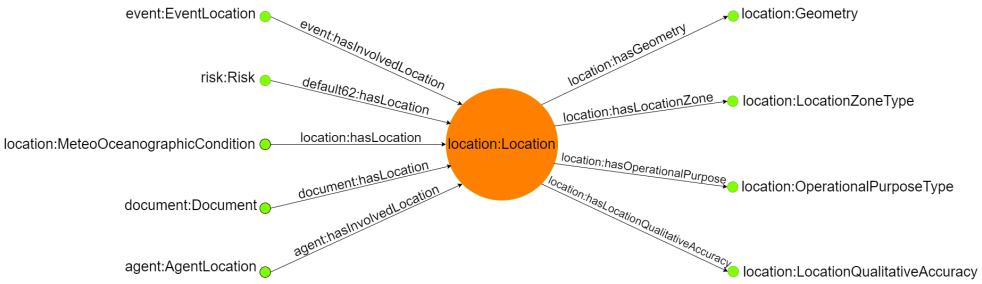

FIG. 4 – *Visualisation dans GraphDB des triplets impliquant la classe* `Location`.

Dans les sections qui suivent, des exemples de classes d'association, d'énumérations en XSD et en Turtle sont présentés.

4.3.1 Classes d'associations

Les concepteurs du processus de conversion en XSD des classes d'association UML de e-CISE ont choisi de définir un fichier XSD par classe associée dont au moins un des deux, et parfois les deux, décrit aussi la classe d'association. De ce fait, il se peut qu'une classe d'association soit décrite deux fois. Les classes d'association du modèle e-CISE sont exprimées en XSD à l'aide du constructeur `xs:complexContent`).

Nous illustrons comment notre processus gère ces classes et les représente dans l'ontologie. Dans l'exemple qui suit, `InvolvedSubjectLocation` est une classe d'association dont le label est `SubjectLocation`. Cette classe implique les classes `Subject` et `Location` via deux `ObjectProperty` : `hasInvolvedSubject` et `hasInvolvedLocation`. Cette classe d'association est également composée d'attributs permettant de qualifier cette association : `LocationType`, `ThreatCertainty`, `IsThreat`, `InvolvedInIncidentRel`. Dans un message CISE ou e-CISE, ces classes d'association permettent d'indiquer les éléments associés à un objet principal. Par exemple, un navire est représenté par une instance de la classe `Vessel`. Ce navire peut être associé à un événement représenté par une instance de la classe `Event`.

```
<xs:element name="InvolvedSubjectLocation"  minOccurs="0" maxOccurs="unbounded">
<xs:complexType>
<xs:complexContent>
<xs:extension base="rel:Relationship">
<xs:sequence>
<xs:element name="Location"
            type="location:Location"
            minOccurs="0"/>
<xs:element name="LocationType"
            type="location:LocationZoneType"
            minOccurs="0" />
<xs:element name="ThreatCertainty"
            type="utils:Confidence"
            minOccurs="0" />
<xs:element name="IsThreat"
            type="xs:boolean"
            minOccurs="0" />
```

```
<xs:element name="InvolvedInIncidentRel"
            type="incident:Incident" minOccurs="1"
            maxOccurs="unbounded" />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
</xs:element>


:SubjectInvolvedSubjectLocation a owl:Class ;
    rdfs:label "SubjectLocation" ;
    rdfs:subClassOf <http://melodi.irit.fr/ontologies/ecise#AssociationClass> .

:hasInvolvedSubject a owl:ObjectProperty ;
    rdfs:domain :SubjectInvolvedSubjectAgent,
        :SubjectInvolvedSubjectLocation,
        :SubjectInvolvedSubjectObject ;
    rdfs:range :Subject .

:hasInvolvedLocation a owl:ObjectProperty ;
    rdfs:comment "Association relation" ;
    rdfs:domain :SubjectInvolvedSubjectLocation ;
    rdfs:range location:Location .

:hasLocationType a owl:ObjectProperty ;
    rdfs:domain :SubjectInvolvedSubjectLocation ;
    rdfs:range location:LocationZoneType .

:isThreat a owl:DatatypeProperty ;
    rdfs:domain :Subject,
        :SubjectInvolvedSubjectLocation ;
    rdfs:range xsd:boolean .

:hasInvolvedInIncidentRel a owl:ObjectProperty ;
    rdfs:domain :SubjectInvolvedSubjectAgent,
        :SubjectInvolvedSubjectLocation,
        :SubjectInvolvedSubjectObject ;
    rdfs:range incident:Incident .

:hasThreatCertainty a owl:ObjectProperty ;
    rdfs:domain :SubjectInvolvedSubjectAgent,
        :SubjectInvolvedSubjectLocation,
        :SubjectInvolvedSubjectObject ;
    rdfs:range utils:Confidence .
```

4.3.2 Enumérations

Le modèle e-CISE comprend un nombre important de classes d'énumération exprimées en XSD à l'aide du constructeur `xs:simpleType`. L'exemple qui suit correspond à la transformation de la classe d'énumération `ConfirmationStatusType`. Celle-ci comprend plusieurs valeurs d'énumération converties en instances (`individuals`) de la classe d'énumération. La classe d'énumération est quant à elle une sous-classe de la classe abstraite `EnumerationType`. Ces énumérations permettent de spécifier les éléments échangés par message. Par exemple, une instance de la classe `MaritimeAnomaly` est spécifié par un type `MaritimeAnomalyType`

indiquant si l'anomalie représente un acte de contrebande, un navire qui dévie de sa trajectoire ou encore un navire qui se stoppe.

```
<xs:simpleType name="ConfirmationStatusType">
CISE Legacy System. It can be local / incoming / transient
<xs:restriction base="xs:string">
</xs:enumeration>
<xs:enumeration value="Failed">
DESCRIPTION: Classification of the entity is AssumeFriend
</xs:enumeration>
<xs:enumeration value="Confirmed">
DESCRIPTION: Classification of the entity is AssumeFriend
</xs:enumeration>
<xs:enumeration value="Other"></xs:enumeration>
<xs:enumeration value="NonSpecified">
DESCRIPTION: Classification of the entity is not Specified
</xs:enumeration>
</xs:restriction>
</xs:simpleType>
```

```
@prefix : <http://www.ecise.eu/datamodel/v1/entity/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

:ConfirmationStatusType a owl:Class ;
    rdfs:subClassOf <http://melodi.irit.fr/ontologies/ecise#EnumerationType> .

:ConfirmationStatusType_Confirmed a :ConfirmationStatusType,
        owl:NamedIndividual ;
    rdfs:label "Confirmed" .

:ConfirmationStatusType_Failed a :ConfirmationStatusType,
        owl:NamedIndividual ;
    rdfs:label "Failed" .

:ConfirmationStatusType_NonSpecified a :ConfirmationStatusType,
        owl:NamedIndividual ;
    rdfs:label "NonSpecified" .

:ConfirmationStatusType_Other a :ConfirmationStatusType,
        owl:NamedIndividual ;
    rdfs:label "Other" .
```

4.4 Discussion

La construction des ontologies via l'utilisation de fichier sources au format .xsd soulève des difficultés de conversion particulières. Dans le cas du modèle CISE, ces difficultés sont notamment dues au nombre important d'énumérations à convertir, au manque d'information quant au nommage des classes d'association et à la conversion des contraintes de cardinalité.

Les énumérations représentent la grande majorité des conversions à traiter. Les requêtes effectuées depuis le point d'accès SPARQL au sein de GraphDB sur l'ontologie e-CISE permettent de chiffrer la proportion de classes d'énumération et d'individus peuplant celles-ci. Les classes d'association représentent un total de 141 classes sur les 269 de l'ontologie. Ces classes sont peuplées par des individus au nombre de 16 648. La plupart des axiomes de l'ontologie proviennent des énumérations du modèle de données. Contrairement aux propositions

de conversion des outils existants, impliquant l'utilisation du constructeur `owl:oneOf` sur les valeurs possibles, une solution plus élégante consiste à définir une classe `EnumerationType` dont les valeurs possibles énumérées sont des instances.

Une autre difficulté lors de la conversion de ces schémas concerne les classes d'association. En effet, le nommage de ces classes n'est pas spécifié dans les fichiers sources .xsd. Deux solutions sont alors possibles pour obtenir un résultat de conversion en accord avec les spécifications : reconstruire le nommage pour qu'il corresponde aux spécifications du modèle de données, en spécifiant dans un fichier le nom de la classe d'association pour deux classes données ; ou alors effectuer la conversion en s'appuyant uniquement sur le nommage spécifié dans les fichiers .xsd. Le premier cas implique une programmation spécifique aux cas des modèles de données CISE et e-CISE. Dans le second cas, il est possible de proposer une solution plus générique qui serait utilisable pour la génération d'ontologies à partir d'autres modèles de données. Ces problèmes de nommage ont été identifiés lors d'une validation manuelle des modèles générés, par la vérification de leur conformité à leur description dans les documents PDF.

Enfin, la conversion des contraintes de cardinalité des schémas de données a soulevé des points de réflexion quant à la pertinence de leur retranscription dans un modèle sémantique géré par l'hypothèse du monde ouvert. En effet, à cause de cette hypothèse, les contraintes de cardinalités ne peuvent indiquer que des maximums ou minimums possibles, mais en aucun cas on ne peut garantir qu'un propriété soit présente avec une cardinalité n. Suivant l'utilisation prévue de l'ontologie générée, les contraintes sont gérées par des systèmes annexes à l'ontologie (à la réception d'un message CISE ou e-CISE par exemple) et avant l'utilisation de cette ontologie. Il est intéressant d'ajouter une option permettant de choisir si les contraintes de cardinalité doivent être retranscrites ou non lors de la génération des ontologies. Les résultats des conversions de ces contraintes sont représentés par des restrictions OWL sur les éléments de type `owl:ObjectProperty` et `owl:DataProperty`.

Finalement, la conversion soulève un dernier problème que nous n'avons pas traité et qui reste à améliorer. Il s'agit du traitement de sources externes de données servant à enrichir la couche terminologique des ontologies (noms des classes et propriétés, labels et commentaires). Pour certaines classes d'association et pour certaines valeurs d'énumérations, l'extraction de termes des tableaux présents dans ces sources exigerait l'utilisation d'outils d'extraction d'information plus sophistiqués que ceux que nous avons retenus.

5 Conclusions

Cet article a présenté un processus de conversion de fichiers XSD du modèle de données e-CISE en langage OWL. Nous avons discuté les difficultés rencontrées et les pistes pour les améliorer. Comme travaux futurs en vue d'améliorer l'ontologie, nous envisageons de poursuivre plusieurs pistes : gérer les versions des ontologies en utilisant des vocabulaires dédiés ; rendre le code générique et modulaire ; générer des alignements entre les versions ontologiques des modèles CISE et e-CISE ; rendre le code et les ontologies publiquement disponibles. Un autre type de perspectives concerne l'utilisation de l'ontologie : définir un processus d'instanciation des ontologies CISE et e-CISE à partir de messages conformes à ces schémas ; permettre la conversion d'un message CISE en message e-CISE en passant par leur représentation en RDF.

Remerciements

Ces travaux sont financés par le programme Horizon 2020 d'innovation et de recherche de l'Union Européenne avec l'accord de financement No. 883374. Ce document ne reflète que la vision des auteurs et la REA (Research Executive Agency) ainsi que la Commission Européenne ne peuvent être tenue responsables pour tout usage de l'information qu'il contient.

Références

Antonopoulos, S., M. Tsogas, M. Moutzouris, A. Kostaridis, A. Aggelis, et L. Perlepes (2020). Andromeda D.3.1 e-CISE Data Model description. Andromeda project (n. 833881) deliverable, EU.

Aryan, P. R. (2013). Converting xml schema to owl in python.

Bedini, I., C. Matheus, P. Patel-Schneider, A. Boran, et B. Nguyen (2011). Transforming xml schema to owl using patterns. In *2011 IEEE Fifth International Conference on Semantic Computing*, pp. 102–109.

EUCISE2020 (2020). Eucise2020 : Technical specifications. deliverable 4.3, revision 1, annex b. Technical report, EUCISE Data Model.

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

Minutolo, A., A. Esposito, M. Ciampi, M. Esposito, et G. Cassetti (2014). 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*, pp. 426–431. IEEE Computer Society.

Riga, M., E. Kontopoulos, K. Ioannidis, S. Kintzios, S. Vrochidis, et I. Kompatsiaris (2021). EUCISE-OWL : an ontology-based representation of the common information sharing environment (CISE) for the maritime domain. *Semantic Web 12*(4), 603–615.

SprintProject (2020). D4.2 A lightweight solution to automate the generation of ontologies, mappings and annotations (C-REL). Project Deliverable D4.2, EU.

Vinasco-Alvarez, D., J. S. Samuel, S. Servigne, et G. Gesquière (2020). From CityGML to OWL. Technical report, LIRIS UMR 5205.

Yüksel, M. (2013). A semantic interoperability framework for reinforcing post market safety studies. Technical report, Middle East Technical University.

Summary

Close cooperation across borders is essential for border surveillance, and in particulier during in case of maritime crises, which requires cross-border information sharing. Existing efforts from the Common Information Sharing Environment (CISE) constitutes a collaborative initiative for promoting automated information sharing between maritime monitoring authorities. The adoption of CISE and its different versions is however limited by the existing serialisation using only XML Schema. This format fails to deliver richer semantics, semantic

interoperability and semantic reasoning capabilities. We discuss here the process of converting the most recent CISE model (e-CISE) in the Ontology Web Langage (OWL) stadard format, thus contributing to improve previous work on transforming XSD schemas into OWL ontologies.