Implementation of the Geographic Information - Metadata (ISO 19115:2003) Norm using the Web Ontology Language (OWL)

Akm Saiful Islam[1], Luis Bermudez[1], Stephane Fellah[2],

Bora Beran[1] and Michael Piasecki[*1]

[1]Department of Civil, Architectural and Environmental Engineering, Drexel University, Philadelphia, PA 19104, USA

[2]PCI Geomatics Inc. Hull, QC, Canada

** corresponding author: e-mail:Michael.Piasecki@drexel.edu, Tel: +1 215 895 1721,*

Fax: +1 215 895 1363

Mailing address: Drexel University

Dept. of Civil, Architectural & Environmental Engineering

3141 Chestnut Street

Philadelphia, PA 19104

Keywords: Geographic Information, Metadata, Ontology, OWL, UML

Abstract. The International Organization for Standardization (ISO) has published the geographic information metadata norm, ISO 19115:2003, to provide a formal structure for describing digital geographic data. The standard is published using an abstract object model that is implemented via the Unified Modeling Language (UML) for conceptualizing the underlying structure of the norm. As a continuation for further enhancement ISO is currently working on a geographic information metadata implementation specification norm, ISO 19139.3 that seeks to find a new implementation for the UML-based abstract model for the ISO 19115:2003. The current implementation approach favors the use of a XML Schema to represent the 19115 norm in machine readable format. However, the use of XML Schema entails a number of disadvantages and as a result falls short in representing the full potential of the UML based ISO 19115:2003 standard. The use of the Web Ontology Language (OWL), built on XML, proves to be a better choice for encoding the standard because it permits a much richer table of semantics as well as a more flexible definition of classes and their attributes when compared to XML schema. In this paper we will discuss the shortcomings of XML schema, some critical issues that surround the conversion of a UML model into an OWL ontology, point out the critical dissimilarities, and suggest a number of mapping approaches that we employed during the mapping effort while trying to overcome some of the incompatibilities that exist between UML and OWL.

1. Introduction

The International Organization for Standardization (ISO) has published the geographic information metadata standard, ISO 19115:2003, to provide a structure for describing digital geographic data (ISO 2003b). Although this standard is primarily developed for digital datasets, its applicability can be extended to be used for many other forms of geographic data such as maps, chart, textual documents, and general purpose data. As a continuation of interoperability enhancement among distributed geographic information, ISO is currently working on a geographic information metadata implementation specification, ISO 19139.3 (ISO 2003a). This new implementation effort will specify a profile using Unified Modeling Language (UML) (OMG 2003b) interpretation and an Extensible Markup Language, XML, Schema (W3C 2001) for machine readability.

Yet, there is evidence that the use of XML-schema, the Resource Description Framework, RDF, (W3C, 2004), or RDF-schema is somewhat insufficient to translate the concepts provided in UML into a fully compatible machine understandable form. We will examine these shortcomings in more detail in the following sections.

XML Schema provides the syntax and grammar to validate documents, which in itself is a desirable feature for automated handling of documents, but the ability of XML Schema to represent semantics for validation and to link concepts across the World Wide Web (WWW) is quite poor (Hendler 2002). For example, a well formed XML document, that is a document that strictly adheres to the rules of XML, could have a <*watershed*> element which contains elements like <*name*>, <*area*>, <*operatedBy*> and <*outletLocation*>. Using an XML Schema we can also specify that a <*watershed*> can have only one <*outletLocation*>. If any <*watershed*> element contains more than one <*outletLocation*>, a validation check of this XML Schema will yield that this information is syntactically wrong. However, if two instances of a watershed have the same values or entries for all its elements, except for <*outletLocation*> (this could be two different names or two different longitude-latitude coordinates), an XML Schema validation will not reveal that one of the instances is wrong, solely based on the fact that only one <*outletLocation*> is allowed for any watershed. Moreover, it is also difficult to state more facts about a watershed, for example that it is a geographic area or that a watershed is similar to a hydrologic unit. Hence, although XML Schema provides a means to validate the structures of a XML document, it does not permit to capture the semantic aspect (correctness of an entry) of the web document (Klein 2001) prompting the need to seek for alternatives that better describe inherent semantic relationships.

Another model to better represent semantics is the Resource Description Framework (RDF), which was designed to improve the representation of information about web resources (Klyne and Carroll 2004). RDF is a data model that contains properties and statements. A property can be a specific characteristic, attribute or relation that describes a web resource. A statement consists of three parts: a subject, a predicate and an object. The subject represents a resource which has properties, while a predicate represents the property and, an object the value of that property. A property value can be either a resource or a literal, e.g. free text. The main difference between a literal and a resource is that literals cannot be the subject or the predicate of a statement. For example, this paper is a resource, which has a property <*author*> whose values are literals, Fig.1. This paper also has a property <*publisher*> which would be this journal. A journal is considered a resource type value because it may have or is related to other elements such as <*subscriber*>, <*number of copies published*>, <*cost of publication*>, etc, which does not permit it to be a literal.

As shown in the above example, RDF can provide a model to represent information, however, it provides no mechanism for describing the relationships between these properties and other resources (Brickley and

Guha 2004). For example, in the above RDF-outline we would need to require that the value of the property <*author*> must reference a person and not a car or a flower. However, the above example clearly demonstrates that <*author*> is a property about very little is said or be demanded from, i.e. that its range of acceptable values can only include persons.

The previously outlined shortcomings can be partially overcome if one uses a RDF Schema, which does provide a mechanism to describe properties and the relationship between these properties and resources. The use of RDF Schema permits the addition of information about the resource to better understand what it actually references. It allows the division of resources in classes and the instance of classes, and defines the property as an instance of a class (Ahmed et al. 2001). As such, RDF Schema represents a semantic extension of RDF by providing mechanisms for describing the relationships between resources. However, it does not provide all the possible forms of descriptions that are useful for representing sensible meanings among RDF classes and their properties. For example, for this paper we would need to specify that each author of the paper is a distinct individual; a demand that RDF Schema can not accommodate because it does not permit the inclusion of a restriction of this type on the property <*author*>.

It is evident from the previous discussion that neither XML Schema, nor RDF, nor RDF Schema fully satisfies the specific needs of a structure to resolve semantic problems that one can easily foresee emerging when using the ISO 19115:2003 metadata norm. On the other hand, the advent of the Web Ontology Language (OWL) promises to introduce a tool that will be able to overcome the shortcomings and problems pointed out earlier (McGuinness 2004). OWL is part of the ongoing effort of the World Wide Web Consortium (W3C) to work towards the Semantic Web which is defined as a future vision of the web by building machine understandable meaningful form of web resources (Berners-Lee, Hendler, and Lassila 2001). OWL essentially enhances RDF Schema by adding more vocabulary for describing properties and classes. For example, the *owl:differentFrom* property of OWL can define that two authors of this paper are distinct individuals. Moreover, we can define other vocabulary such as the maximum cardinality of the <*publisher*> property that can be set to unity to demand that any article can not be published in more than one journal. Hence, the use of OWL to formally specify in an ontology the relationship between elements (classes) and their properties (attributes) promises to provide an adequate means through which the full potential of the ISO 19115 :2003 norm can be utilized.

In this paper we have created an ontology for geographic information metadata that implements the norm in a machine understandable format. In the next section, the application of a geographic ontology will be discussed with some useful examples. In section 3, we will discuss the similarities and dissimilarities when attempting to map a UML model into an OWL ontology. In Section 4 we describe in more detail the mapping process and conventions necessary when mapping the ISO 19115:2003 UML model to the OWL ontology.

2. Ontology Concept

Ontologies can be applied in many ways such as for detecting inconsistencies, performing validations, extending metadata, ensuring interoperability, and for the validation of information integrity and hierarchy (Fensel 2003). To provide the reader with a better understanding how ontologies in OWL can achieve the above objectives, we will outline some small examples in the following paragraphs.

To demonstrate with a simple example how OWL can be utilized for detecting possible inconsistencies let us consider an instance of a small subset of geographic metadata. For example, as defined in ISO 19115:2003, the orientation of a point in a pixel is permitted in only five possible ways: center, lower left, lower right, upper right, and upper left, Fig.2 (a). This concept can be represented in an ontology by restricting the range of the property *pointInPixel* to a class, *MD_PixelOrientationCode*. The class *MD_PixelOrientationCode* can be enumerated by its five instances *center*, *lowerLeft*, *lowerRight*, *upperRight*, and *upperLeft.* If the *pointInPixel* has as value *bottom,* which is not inside the allowed range, a consistency error will be reported.

A second example highlights the ability of ontologies to be easily extended to support the completion of geographic information if new metadata elements or vocabulary must be added. For example, if a user is seeking to identify earthquakes that occur around Japanese islands, the user can define a new metadata term *JapanBox*, which is an extension from the ISO 19115:2003 metadata element *EX_GeographicBoundingBox* whose property *westBoundLongitude* is 130^0, *eastBoundaLongitude* is 145^0, *southBoundLatitude is* 30^0, and *northBoundLatitude* is 45^0 as shown in Fig.2 (b). This term can be reused for similar queries by the user.

Third, ontologies can be used as a bridge for interoperability demands among different metadata standards. For example, ISO 19115:2003 has a metadata element *title* in the *CI_Citation* package, which is similar to the Dublin Core element *title*. Using OWL it is possible to declare an *owl:equivalentProperty* such that both these terms refer to the same thing; while, XML Schemas and RDF schemas do not provide such mapping capabilities. One such effort uses a Dublin Core Metadata Ontology to provide meta information for other ontologies in the HealthCyberMap domain (Kamel Boulos, Roudsari , and CarSon 2001).

Fourth, ontologies can be useful for validation and verification of geographic data. For example, the maximum occurrence of the ISO metadata element *title* in *CI_Citation* is one. Let us suppose that an instance document of the metadata ontology has two different titles such as “water elevation” and “stage”.

As the maximum cardinality restriction is unity for the *title* property, it will be inferred that both “water elevation” and “stage” represents the same thing. In contrast, a XML Schema would produce an error indicating a data inconsistency. The above example highlights a significant difference between XML Schema and OWL, that is XML Schema provides only syntactic information whereas OWL provides semantic information for a domain.

Fifth, ontologies can be used to define the basis for system integrity. For example, the ISO 19115:2003 *CI_Address* class has properties *deliveryPoint*, *city*, *administrativeArea*, *postalCode*, *country* and *electronicMailAddress* that are intended to describe any contact address. However, the property *electronicMailAddress* has zero-infinite cardinality, making it on optional property. In contrast, if we were to define a new metadata class called *webGroupAddress* then we could demand that instances of this class must have *electronicMailAddress* for each contact, i.e. we would need to change the cardinality to unity-infinite. OWL permits the change of the cardinality. The result is that it is now possible to restrict users such that they are required to fill the email address while creating an instance of *webGroupAddress*.

Finally, ontologies can improve query results inferring over the hierarchies of contained terms. For example, a geographic information ontology can classify natural water bodies as stream, river, lake, pond or ocean. Lake can be further classified into saltwater lake and fresh water lake based on water salinity. For a person who is looking for a lake to go fishing, an ontology based search could not only provide the list of lakes but could also present a classification of the results in fresh water lakes and salt water lakes.

The above discussion was intended to outline possible difficulties when developing community specific metadata sets based on the ISO 19115:2003 and the role OWL could play in overcoming these difficulties. It is important, to realize that the conceptualization of the ISO 19115:2003 (which is provided in UML) and its conversion into a machine readable framework is not straightforward but fraught with shortcomings and potential pitfalls. Among the three possibilities shown, OWL, seems to be the most successful in sparing these problems. In the following section we will point out some of the problems that persist when attempting to map UML into OWL, and also review some of the newer concepts employed to date.

3. Mapping of UML model and OWL ontology

The basic difference between the concepts of ontology and object oriented modeling lies on their motivation (DSTC 2003). The intention of object oriented modeling is to capture sufficient knowledge for a specific purpose whereas the goal of an ontology is to capture facts or knowledge about a domain. Therefore, an object oriented model should always be an abstraction of an ontology. Object oriented models are typically visualized using UML diagrams that demonstrate well the static application structure of the model, the different aspects of dynamic behavior, and the organization and management of the application modules. It is because of its popularity and wide-spread use that the idea of using UML as an ontology representation language has recently received growing attention among researchers.

There are a number of initiatives working on representation of domain knowledge in UML. One such effort was made by Cranefield (2002) who attempts to represent a RDF Schema as a UML class and an object diagram using StyleSheet Language for XML; Transformations (W3C 1999). Some very basic work converting UML to OWL such as *owl:Class*, *owl:SubClassOf*, *owl:DatatypeProperty* and *owl:ObjectProperty* has been carried out by using XSLT (Price 2004). Kogut et al. (2001) suggested that using UML provides an excellent notation for ontologies by considering a number of factors such as: (1) UML is a graphical notation based on many years of experience, (2) UML is a open standard maintained by Object Management Group (OMG), (3) UML has been widely adopted in the software industry, and

(4) UML has a standard mechanism for defining extensions for specific application such as ontologies. They pointed out that the Object Management Group (OMG) has built a Meta Object Facility (MOF) to provide metadata and semantics of the UML model. The idea behind MOF is to provide an abstract language and a framework to define a metadata layer for models (OMG 2002a). Guizzardi et al. (2002), proposed to extend UML to represent a knowledge language for a conceptual model. One such extension is *Powertype* which is a special class whose instances are classes in the UML. Finally, Knublauch (2003) demonstrated a concept to map the Protégé (open source knowledge-modeling platform) metadata model into a UML meta-model. The basic idea is to create a metadata model for the Protégé's underlying object-model and then to convert this meta model into XML Metadata Interchange (XMI) (OMG 2002b) using Java Metadata Interface (JMI) (Microsystem 2002).

Despite these recent efforts, currently no standard specification exists to express an OWL ontology as a UML model and vice versa. It is worthwhile mentioning that the OMG group recently sought proposals for a specification consisting of three components: (1) a MOF metamodel for ontology, which is defined as ontology definition metamodel (ODM), (2) a UML profile for ontology, and (3) mapping between ODM with UML and OWL (OMG 2003a). At present, OMG is the selection process among four initial submissions, yet it remains unclear when the recommendation for a specification of a MOF metamodel for OWL could be expected. Hence, we will attempt to point similarities and dissimilarities between UML and OWL, resolve those by suggesting workable definitions and then manually map UML into OWL.

3.1 Similarities and Differences of UML and OWL Concepts

While there is no reference available that points out the similarities between UML and OWL, Baclawski (Baclawski et al. 2002) demonstrated mapping similarities between UML and the Darpa Agent Markup Language (DAML). DAML is a language that can be considered the predecessor of OWL that is based on

RDF and RDF Schema to represent ontologies. Hence, this comparison lends to itself to be used as a base to infer similarities for UML and OWL. For example, UML *class* can be expressed in a similar concept using *owl:Class*, UML *association* can be mapped into *owl:ObjectProperty* and UML *attribute* can be expressed as *owl:DatatypeProperty*. On other side, there are some concepts in OWL which cannot be mapped as straight forward into UML. For example, there is no equivalent concept in UML to express the *owl:unionOf* concept of OWL. A summary of the similarities between UML and OWL concepts has presented in Table 1, while a list of the most significant dissimilarities is given in Table 2.

Before we start to translate (or map) the metadata UML abstract model into an OWL-based ontology, it is worthwhile to further discuss some of the dissimilarities, i.e. potentially difficult points when mapping UML into OWL.

a. Incompatibility of properties

In OWL properties are defined as a *first-class* concept whereas UML *attributes* and *associations* are not first-class. What this means is that in OWL a property can exist without defining its range and domain, while in UML attributes need to have a range and domain definition. Therefore, in an ontology, every property must have a different name as they are declared independently from classes. On the other hand, different UML attributes and association with same name can exists in a single UML model in different classes. For example, the *language* attribute of ISO 19115:2003 can be found in the *MD_Metadata*, *MD_DataIdentification*, *MD_FeatureCatalogDescription* classes to represent metadata language, data language and feature catalog language, respectively. In contrast it is not possible to create three OWL owl:D*atatypeProperty* in a single ontology with the same name. This must be carefully when considered when mapping UML into OWL and we will provide some mapping guidelines to resolve this issue in the next section.

b. Monotonic worlds

OWL and other knowledge representation systems are monotonic in nature, which means that adding a new fact does not affect the correctness of a previously declared fact (Baclawski et al. 2002). In contrast, UML assumes a closed world, which means that if something is absent or is newly added, it will be assumed as false or non-existent. For example, every metadata instance document can only have a single metadata documentation creation date, which is defined as *dateStamp* element. If any particular instance has two *dateStamp* properties, the UML model will consider this situation as a violation of the maximum cardinality requirements. In contrast, monotonic logic does not imply the same conclusion, because it is possible that both metadata creation dates are the same, which in turn would establish equality of the dates and as such be consistent.

c. Modularization

The ISO presents metadata in different UML packages aimed at reducing the maintenance effort. It is easier to handle, reuse and maintain a UML model if its different analogous components are bundled into different packages. OWL on the other hand, does not support the package-concept as UML does. We can only interact with different ontologies using the OWL *owl:import* element and XML "namespaces", which provide unique names. But unlike UML packages, the *owl:import* element provides no semantic explanation about the use of resources within the imported ontology. To avoid this problem when using OWL, we did not modularize the ontology into different packages rather we kept it as a single file.

d. Generalization

While the UML *generalization* and *specialization* relationship can be expressed in similar fashion in OWL using the *sub-class* and *super-class* relationships, there are semantic differences between the *generalization* relationship of UML and the *subclass* relationship of OWL. The *generalization* relationship in UML is *behavioral*, which means that *specialization* can add new methods or attributes thereby overriding the method of the *generalized* class. On the other hand, the *sub-class* relationship of OWL is defined and set as *theoretic*, which means that *sub-class* can restrict the *super-class* without adding any attributes or methods (Baclawski et al. 2002). Despite these semantic differences mapping is important as these two concepts identify and represent parent and children relationship between two classes.

e. Abstract class

UML defines classes that are considered *abstract* for which an instance cannot be created. On the other side, OWL does not support the concept of an *abstract* class, which means that every class in OWL is available for creating a new instance. This incompatibility of mapping cannot be resolved directly and we can only provide an annotation property to tag this type of class as *abstract*.

There are several other incompatibilities, all of which we will address in the following section, including a suggestion or mechanism for resolving them.

4. General rules for transforming the ISO 19115:2003 UML model to OWL ontology

In this section we explain in detail the conventions we have adopted to translate the ISO 19115:2003 UML model into an OWL ontology. Section 3.1 to 3.8 addresses naming conventions, data types, and restriction conditions between the two concepts. In section 3.9 to 3.15, we address the conversion of different UML stereotypes described in ISO 19115:2003 into OWL ontology.

4.1 Name / role name

Each metadata entity or metadata element has a name or label which is derived from single or multiple concatenated words. Although metadata element names are typically unique within the entire specification, a metadata element may not be globally unique as other entities (sub-portions of the specification) can contain the same element name. This could be a problem if we are using name as identifiers (e.g. *rdf:ID*), which should be unique throughout the namespace. Fortunately, there are very few elements which have identical names. We use the following criteria to resolve this problem.

a. If duplicate elements have the same range, only one element is declared without defining its domain. A class that uses this property could refer to it by creating a restriction.

b. If duplicate elements have different range, create a new element using its short name.

c. If duplicate element is a code list element, an underscore (“_”) is added before its name. As we have only pairs of duplicate elements, this approach can resolve the conflict successfully.

For example, the metadata element *name* appears in seven metadata entities (ISO 19115:2003 classes), as presented in Table 3. To resolve this problem, a data type property *name* was created with range *xs:string* with an open domain for four identical range elements.

```
<owl:DatatypeProperty rdf:ID="name">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
```

Three *owl:ObjectProperty*: *refSysName, asName, medName* were created for different range elements using their short names. The ranges of the elements are kept and remain the same as that of their associated class.

Codelist element *point* appears in both code lists *MD_CellGeometryCode* and *MD_GeometricObjectTypeCode*. We use *rdf:ID=”point”* in *MD_CellGeometryCode* and *rdf:ID =“_point”* in *MD_GeometricObjectTypeCode*.

4.2 Short name and domain code

The possibility to use short names is a unique feature of the ISO 19115:2003 standard which also suggests its use with the XML or SGML language. On the other hand, the ISO 19139.3 norm uses the metadata element name as a unique identifier for the XML Schema to increase human readability. We suggest to take an approach that is similar to the ISO 19139.3 specification and to use the metadata element name as *rdf:ID*. However, in order to allow the use of short names as permitted in the UML conceptualization of the ISO 19115:2003, we created an annotation property *_shortname* to document this specific feature of ISO 19115:2003.

```
<owl:AnnotationProperty rdf:ID="_shortName” />
<owl:Class rdf:ID="MD_Metadata">
<iso:shortName>Metadata</iso:shortName>
</owl:Class>
```

ISO 19115:2003 defines a special data type class *enumeration* (see also 4.12) whose instances form a list of literal values. *Enumeration* gives a list of well understand values by assuming that all the values of the list are known and that this list cannot be extended in future. To provide support for a more flexible list to which more values can be added in future, ISO 19115:2003 also defines *codelist* which is an open type of *enumeration*. Every element in *codelist* has a name and a domain code. Domain codes are comprised of a 3 digit unique number within the *codelist*. To represent this domain code feature, unlike a *_shortname*, an annotation property *_domainCode* has been created to provide a higher degree of readability for each *codelist* element.

```
<owl:AnnotationProperty rdf:ID="_domainCode” />
<iso:MD_RestrictionCode rdf:ID="copyright">
<iso:_domainCode>001</iso:_domainCode>
</iso:MD_RestrictionCode>
```

4.3 Definition

The ISO 19115:2003 norm provides the utility of a data directory within which descriptions for a metadata entity or element can be outlined to describe the UML abstract model. These descriptions are quite useful when the need arises to better understand what the element tries to describe. This feature can be mapped into OWL via the *rdfs:comment* property. For example, the definition for the *MD_Metadata* entity can be described as follows;

```
<owl:Class rdf:ID="MD_Metadata">
<rdfs:comment> root entity which defines metadata about a resource or
resources
</rdfs:comment>
</owl:Class>
```

4.4 Obligation/condition

The ISO 19115:2003 defines three types of obligations for documentation of metadata entities or metadata elements which are classified as mandatory, conditional or optional. Mandatory and optional obligations are identified in UML as cardinality associations, that are mapped to an OWL ontology via cardinality restrictions. In OWL, a cardinality restriction is an anonymous class, which applies a restriction on a specific property and is declared as a super class of the class using that property.

4.4.1 Mandatory (M)

Mandatory metadata entities or metadata elements are those which must be documented, i.e. must receive a value if the norm is used for a data-set description. If a class defines a minimum or exact cardinality greater than one for the usage of a property, it is mandatory. A cardinality restriction in OWL is created with the value of that cardinality, and it is applied on the entity (property) and is declared super class of the entity (class) using the property.

a. If a metadata entity or metadata element has more than a single occurrence and also has the same maximum and minimum occurrences, the same cardinality restrictions will be used.
b. If the metadata entity or metadata element has different minimum and maximum occurrences, the minimum cardinality restriction will be used.
c. If the metadata entity or metadata element has a maximum occurrence equal to 1, it will be considered as an *owl:FunctionalProperty*.

4.4.2 Conditional (C)

A conditional element defines an element that is mandatory under certain conditions. This definition is quite difficult to map in OWL and at present we do not know of any mechanism to map these conditional elements (or rather the conditions defining the mandatory use of this element) to an ontology in OWL. This is one of the shortcomings of the present ontology and requires further investigation. For now, conditions are set as an annotation property *condition*.

```
<owl:AnnotationProperty rdf:ID="condition” />
<owl:ObjectProperty rdf:ID="characterSet”>
<iso:condition>ISO 10646-1 not used ?</iso:condition>
</owl:ObjectProperty>
```

4.4.3 Optional (O)

Optional metadata entities or metadata elements provide a means towards the full documentation of the data but are not necessary. This feature is relatively easy to map into OWL by setting the minimum cardinality restriction of the optional metadata element to zero.

```
<owl:Restriction>
<owl:onProperty>
<owl:DatatypeProperty rdf:about="#metadataStandardName"/>
</owl:onProperty>
<owl:minCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">0
</owl:minCardinality>
</owl:Restriction>
```

4.5 Maximum Occurrence

Maximum occurrence indicates the possible maximum number of values that can occur for a metadata element in a metadata entity. In OWL, this can be expressed as a local cardinality restriction on the class.

The rules we followed are:

a. If the maximum and minimum occurrences of an metadata entity or metadata element are the same, maximum occurrence will be represented by the *owl:cardinality*

b. If a metadata entity or metadata element has a different maximum and minimum occurrence, maximum occurrence will be represented by the *owl:maxCardinality*, and the minimum as *owl:minCardinality* restriction.

```
<owl:Restriction>
<owl:onProperty>
<owl:DatatypeProperty rdf:about="#metadataStandardName"/>
</owl:onProperty>
<owl:maxCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1
</owl:maxCardinality>
</owl:Restriction>
```

4.6 Data Type

The ISO 19115:2003 norm uses a number of data types which are defined in “Geographic Information Conceptual Schema Language (ISO 19103)” norm to provide distinct values for the metadata elements (ISO 2004). We use a similar approach as taken in the ISO 19139.3, i.e. to translate data types into XML base types. The conversion lists from the ISO 19115:2003 data type class into XML base types are given in Table 4.

A Metadata element that has its domain as data type, is mapped as *owl:DatatypeProperty*. The range of the OWL property is then given by an XML base type. For example, the range of the metadata element "hoursOfService" is defined as follows.

```
<owl:DatatypeProperty rdf:ID="hoursOfService">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
```

4.7 Domain and Range

In the ISO:19115:2003 norm the data directory *domain* represents the line numbers covered by the entity. We use *rdfs:range* to represent the domain defined in the ISO 19115:2003. OWL classifies properties into two basic groups *owl:DatatypeProperty* and *owl:ObjectProperty*. If the domain in the ISO 19115:2003 is *Free Text, Integer, Real, Boolean, Date, DateTime* we map the property to *owl:DatatypeProperty*. If the domain of a property in ISO 19115:2003 is a class, we use *owl:ObjectProperty* as shown in Table 5.

4.8 UML Model Stereotypes

A UML *stereotypes* is an extension mechanism for existing UML concepts. We suggest to use an annotation property "*stereotypes*" to provide this information. For example, the *stereotypes* for the class MD_Metadata is "*class"*.

```
<owl:AnnotationProperty rdf:ID="stereotypes" />
<owl:Class rdf:ID="MD_Metadata">
```

```
<iso:stereotypes>class</iso:stereotypes>
</owl:Class>
```

4.9 Abstract class

Abstract class is defined in ISO 19115:2003 as a class that cannot be directly instantiated, Since OWL, as mentioned before, does not support an abstract class concept, we use an annotation property *stereotype* with a value of *abstractClass*. An example of OWL stereotype for an abstract class is shown below.

```
<owl:Class rdf:ID="MD_Identification">
<iso:stereotypes>abstractClass</iso:stereotypes>
</owl:Class>
```

4.10 Generalization

Generalization is defined in ISO 19115:2003 as a relationship between a generalized super-class and a specified subclass, Fig.3. Generalization in OWL can be expressed as *owl:Class* and *owl:subClassOf*.

```
<owl:Class MD_Identifiation></owl:Class>
<owl:Class rdf:ID="MD_ServiceIdentification">
<rdfs:subClassOf rdf:resource="#MD_Identification"/>
</owl:Class>
<owl:Class rdf:ID="MD_DataIdentification">
<rdfs:subClassOf rdf:resource="#MD_Identification"/>
</owl:Class>
```

4.11 Enumeration

Enumeration is defined in ISO 19115:2003 as a data type whose instances come exclusively from a list of named literal values, as shown in Fig.4. Enumeration is treated through the *owl:oneOf* property in OWL.

The process in OWL is to: 1) create an enumerated class; 2) create all the code list elements as the instances of the class, 3) link the enumerated class with an equivalent class, which is a collection of the instances of that class, and 4) the range of the property, which will hold the values of the enumeration, is restricted to allow values of the equivalent created class.

For example, the enumerated class *MD_ObligationCode* is represented as a enumerated *owl:equivalentClass* using *owl:oneOf* collections:

```
<owl:Class rdf:ID="MD_ObligationCode">
<owl:equivalentClass>
<owl:Class>
<owl:oneOf rdf:parseType="Collection">
<iso:MD_ObligationCode rdf:ID="mandatory" />
<iso:MD_ObligationCode rdf:ID="optional" />
<iso:MD_ObligationCode rdf:ID="conditional" />
</owl:oneOf>
</owl:Class>
</owl:equivalentClass>
</owl:Class>
```

4.12 CodeList

CodeList are typically used to describe a long list of likely values, which makes it a more open or flexible enumeration. *CodeList* is expressed by creating instances of the class they belongs to, Fig.5. However, because *CodeList* needs to retain its ability to be extended in the future, its instances should not be enumerated using the *owl:oneOf* property. The stereotype annotation for CodeList is *codelist*.

For example, the *CodeList* class *MD_CellGeometryCode* is represented as an enumerated *owl:equivalentClass* rather than using *owl:oneOf* collections.

```
<owl:Class rdf:ID="MD_CellGeometryCode">
<owl:equivalentClass>
</owl:Class>
<iso:MD_CellGeometryCode rdf:ID="point" />
<iso:MD_CellGeometryCode rdf:ID="area" />
```

4.13 DataType class

A *datatype* class is a class that describes a set of values representing attributes of a particular object. For example, *citation* is an attribute of MD_Identification. The values of citation are of type CI_Citation, where CI_Citation is a *datatype*. Datatype classes are composed of one or more primitive *datatypes* (e.g. number, string and date), as well as other *datatype* classes.

While primitive *datatypes* are mapped to XML Schema *datatypes*, classes with stereotype *datatype* are mapped to OWL classes. To keep the stereotype information from the ISO specification for the datatype classes, an annotation property *iso:stereotypes* with value *datatypeClass* is created in the OWL class. DQ_Result is an example of a *datatype* class, and its encoding in OWL is shown below:

```
<owl:Class rdf:ID="DQ_Result">
<iso:stereotypes>datatypeClass</iso:stereotypes>
</owl:Class>
```

4.14 Union

A union class allows an alternative selection (one of) of two or more classes. This class is used in the ISO specification so there is no need to create a common super type class. An union class is mapped to an OWL class, and its union definition is stored in the iso:stereotypes annotation property with a value of unionClass. An example for MD_ScopeDescription is shown below.

```
<owl:Class rdf:ID="MD_ScopeDescription">
<iso:stereotypes>unionClass</iso:stereotypes>
</owl:Class>
```

Finally, the complete ISO 19115:2003 norm has been mapped into OWL and is being made available to the general public at http://loki.cae.drexel.edu/~wbs/ontology/list.html . Several links have been created at other web-pages, one of which is hosted at the Protégé Ontology Library at Stanford University at http://protege.stanford.edu/ontologies/ontologies.html.

5. Summary

In this study we have developed a mapping approach for geographic information metadata, ISO norm 19115:2003, from its conceptualization in UML into an ontology using the Web Ontology Language, OWL. This work has been motivated by the realization that the current implementation approach utilized in the ISO 19139.3, i.e. the use of XML schema, falls well short of being able to fully represent the concepts inherent in the UML conceptualization of the standard. These shortcomings have been discussed also including the alternative approaches using the Resources Description Framework schema. In contrast, we have shown that the use of an ontology, and its realization in OWL, has the potential of much better implementing the underlying UML concepts of the ISO 19115:2003. In addition, we demonstrated a step by step conversion process from ISO 19115:2003 UML model to OWL ontology addressing each of the items that, in our opinion, needed a specific mapping rule. While many of the dissimilarities have been resolved and a mapping rule has been derived, we also have to realize that a 100% perfect mapping has not been achieved and that the present specification of UML does not permit a complete conversion of an object oriented concept into a knowledge based concept. We hope that this work can help in establishing an automated conversion framework in future that would permit, through an ontology definition metamodel, to convert any object oriented model into knowledge based concept. We have made the ISO 19115:2003 ontology available for public use and hope that it will be used and also tested with a number of extensive real world examples for future modification and corrections.

6. Acknowledgements

This work has been partially supported by a grant from the National Oceanographic Partnership Program (NOPP), grant number NAG13-0040, and by the National Science Foundation (NSF) under grant number 0412904. This ontology has been created using the ontology software Protégé. We like to thank Dr. Holger Knublauch at Stanford University who is the developer of the OWL plug-in of Protégé for his continued support and feedback during the development of the ISO 19115:2003 OWL implementation.

References

Ahmed K, Ayers D, Birbeck M, Cousins J, Doods D, Lubell J, Nic M, Rivers-Moore D, Watt A, Worden R, and Wrightson A 2001 *Professional XML Metadata*. Birmingham, Wrox Press
Baclawski K, Kokar M K, Kogut P A, Hart L, Smith J, Letkowski J, and Emery P 2002 Extending the Unified Modeling Language for ontology development. *Software System Model* 1: 1-15.
Berners-Lee T, Hendler J A, and Lassila O 2001 The Semantic Web. *Scientific American* 284 (5):34-43.
Brickley D, and Guha R V 2004 RDF Vocabulary Description Language 1.0: RDF Schema. WWW document, http://www.w3.org/TR/rdf-schema/
Cranefield S 2002 UML and the Semantic Web. WWW document, http://www.semanticweb.org/SWWS/program/full/paper1.pdf
DSTC 2003 Ontology Definition Metadmodel -DSTC Initial Submission. WWW document, http://neptune.irit.fr/Biblio/03-09-04.pdf
Fensel D 2003 *Spinning the semantic Web: bringing the World Wide Web to its full potential*. Cambridge, Mass., MIT Press
Guizzardi G, Herre H, and Wagner G 2002 Towards Ontological Foundations for UML Conceptual Models. In *Proceedings of the 1st International Conference on Ontologies, Databases and Applications of Semantic (ODBASE 2002)*. Irvine,California, USA
Hendler J A 2002 XML and the Semantic Web. *XML Journal*.
ISO 2003a Geographic Information - Metadata - Implementation Specification (ISO 19139.3).
ISO 2003b Geographic Information - Metadata (ISO 19115:2003).
ISO 2004 Geographic Information - Conceptual Schema Language (ISO 19103).
Kamel Boulos M N, Roudsari A V, and CarSon E R 2001 Towards a Semantic Medical Web: HealthCyberMap's Dublin Code Ontology in Protege-2000. In *Proceedings of the Fifth*

International Protege Workshop, Sowerby Centre for Health Informatics at NewCastle (SCHIN). Newcastlem, England
Klein M 2001 XML, RDF, and Relatives. *IEE Intelligent System*, 26-28.
Klyne G, and Carroll J J 2004 Resource Description Framework (RDF): Concepts and Abstract Syntax. WWW document, http://www.w3.org/TR/rdf-concepts/#ref-rdf-semantics
Knublauch H 2003 Ontology Design and Software Technology - Protégé and Java, UML & Model-Driven Architecture. WWW document, http://protege.stanford.edu/plugins/xmi/download/2003-06-12%20SMI-Knublauch.pdf
Kogut P, Cranefield S, Hart L, Dutra M, Baclawski K, Kokar M, and Smith J 2001 UML for Ontology Development. *Knowledge Engineering* 17 (1):61-64.
McGuinness D L 2004 OWL Web Ontology Language Overview. WWW document, http://www.w3.org/TR/2004/REC-owl-features-20040210/
Microsystem S 2002 Java Metadata Interface (JMI).
OMG 2002a Meta-Object Facility Specification (MOF), version 1.4. WWW document, http://www.omg.org/technology/documents/formal/mof.htm
OMG 2002b XML Metadata Interchange Specification. WWW document, http://www.omg.org/technology/documents/formal/xmi.htm
OMG 2003a Ontology Definition Metamodel. WWW document, http://www.omg.org/cgi-bin/doc?ad/2003-03-40
OMG 2003b Unified Modeling Language Specification. WWW document, http://www.omg.org/technology/documents/formal/uml.htm
Price D 2004 An Introduction to ISO STEP Part 25. WWW document, http://www.exff.org./
W3C 1999 XSL Transformations (XSLT) Version 1.0. WWW document, http://www.w3.org/TR/xslt
W3C 2001 XML Schema. WWW document, http://www.w3.org/XML/Schema

Table.1. Similarities between UML and OWL concepts

UML concept	OWL concept
Package	Ontology
Class	Class
Attributes, Associations	Property
Generalization Relation	Hierarchy
Data Type	Data Type
Object	Instance
Multiplicity Constrains	Restrictions

Table.2. Major Incompatibilities of conversion from UML to OWL concepts

Incompatible UML Concept	**OWL Concept**
Associations as second class concept	Property as first class concept
Closed world	Open world (monotonic)
Modularity	Not supported
Behavioral specialization/generalization	Set theoretic subclass/ superclass
Multiple meta levels	Single meta level
Abstract class	Not supported
Scope – private , public and protected	Not supported
Associations as second class concept	Property as first class concept

Closed world	Open world (monotonic)
Modularity	Not supported
Behavioral specialization/generalization	Set theoretic subclass/ superclass
Multiple meta levels	Single meta level
Abstract class	Not supported
Scope – private , public and protected	Not supported
Associations as second class concept	Property as first class concept
Closed world	Open world (monotonic)
Modularity	Not supported
Behavioral specialization/generalization	Set theoretic subclass/ superclass
Multiple meta levels	Single meta level
Abstract class	Not supported
Scope – private , public and protected	Not supported

Table.3. Conversion of duplicate metadata element, *name*

ISO 19115:2003 Class	***owl:domain***	***owl:Range***	***rdf:ID***
MD_ExtendedElementInformation	-	*xs:string*	*name*
CI_Series	-	*xs:string*	*name*
RS_ReferenceSystem	*RS_ReferenceSystem*	*RS_Identifier*	*refSysName*
MD_Format	-	*xs:string*	*name*
MD_ApplicationSchemaInformation	*MD_ApplicationSchemaInformation*	*CI_Citation*	*asName*

MD_Medium	*MD_Medium*	*MD_MediumName Code*	*medName*
CI_OnlineResource	-	xs:string	name

Table.4. Data type Implementation

ISO 19103 Class	**XML base type**
Binary	xs:base64Binary
Boolean	xs:Boolean
Character	xs:string
CharacterString	xs:string
Date	xs:date
DateTime	xs:dateTime
Decimal	xs:decimal

Integer	xs:integer
Number	xs:decimal
Real	xs:decimal
Time	xs:time
URI	xs:anyURI

Table.5. ISO 19115 Domain and OWL property mapping

ISO 19115:2003 Domain	**Property**
Free Text, Integer, Real, Boolean, Date, DateTime	*owl:DatatypeProperty*
Class or *Association*	*owl:ObjectProperty*

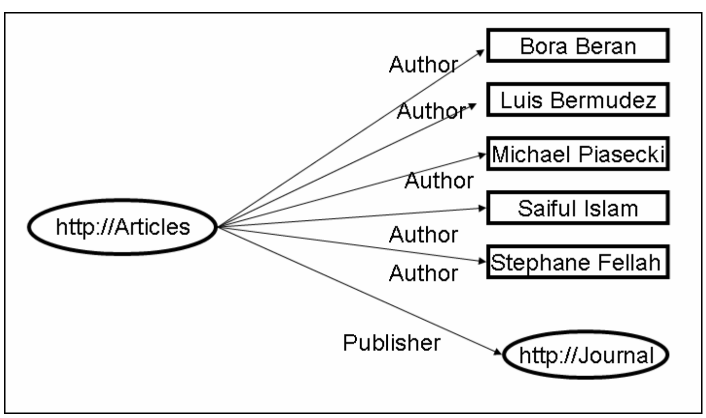
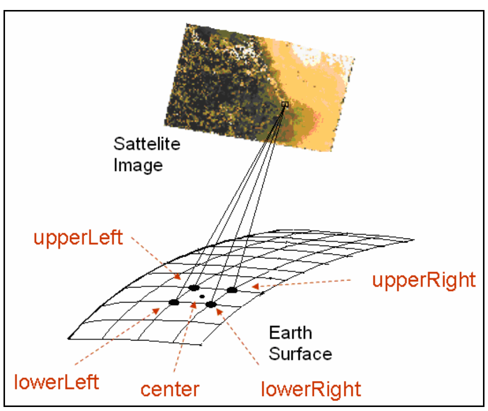
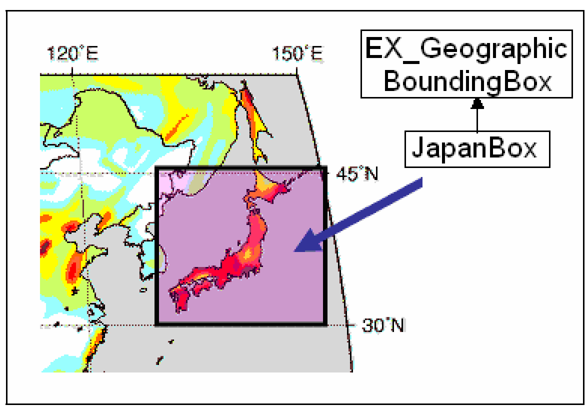


Fig.1. RDF graph to describe this paper which has five authors and a publisher of type journal.


Sattelite
Image
upperLeft
upperRight
Earth
Surface
lowerLeft
center
lowerRight


120°E
150°E
EX_Geographic
BoundingBox
JapanBox
45°N
30°N

(a) (b)

Fig.2. (a) Orientation of the point in a pixel corresponding to the earth location of the pixel for a regularly spaced grid, and (b) New metadata *JapanBox* is extended from *EX_GeographicBoundingBox* metadata to locate earthquakes around Japan.

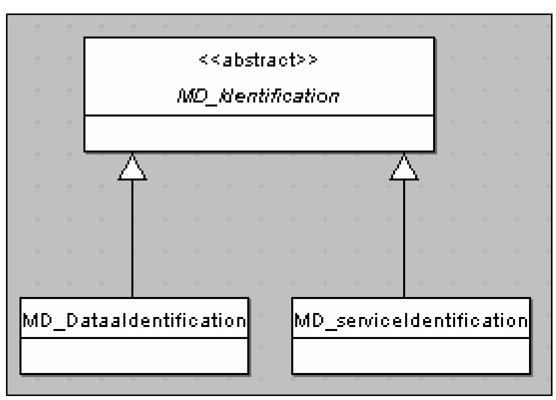


Fig.3. ML Generalization relationship in ISO 19115:2003

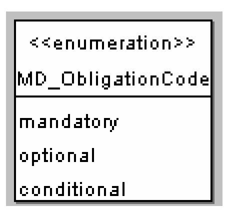


Fig.4. UML of Enumeration in ISO 19115:2003

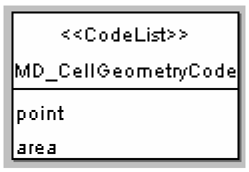


Fig.5. UML Code list in ISO 19115:2003

Table of figures

Fig.1. RDF graph to describe this paper which has five authors and a publisher of type journal.

Fig.2. (a) Orientation of the point in a pixel corresponding to the earth location of the pixel for a regularly spaced grid , and (b) New metadata *JapanBox* is extended from *EX_GeographicBoundingBox* metadata to locate earthquakes around Japan.

Fig.3. ML Generalization relationship in ISO 19115:2003

Fig.4. UML of Enumeration in ISO 19115:2003

Fig.5. UML Code list in ISO 19115:2003