*sed*Onto: A Web Enabled Ontology for Synthetic Environment Representation Based on the SEDRIS Specification

Mehul Bhatt, Wenny Rahayu
La Trobe University
Dept. of Computer Science, Bundoora
Victoria, Australia 3086.
+61-3-94791282
{**mbhatt, wenny}@cs.latrobe.edu.au**

Gerald Sterling
Air Operations Division, DSTO
PO Box 4331 Melbourne
Victoria Australia 3001
+61-3-96267728
Gerald.Sterling@dsto.defence.gov.au

Keywords: Synthetic Environment (SE), Simulation Interoperability, Ontologies, Description Logics, Web Ontology Language (OWL), SEDRIS.

Abstract: The application of Ontologies for solving interoperability problems has been widely recognised across multiple domains. Ontologies, by virtue of the shared conceptualization that they provide, may be communicated between people and application systems thereby facilitating interchange, interoperability and common understanding. The Modeling & Simulation (M&S) community's XMSF charter also prescribes the use of ontologies for the definition, approval and interoperability of complementary taxonomies that may be applied across multiple simulation domains.

Our current research investigates the use of a Web-Based, Ontological formalism as the basis of Synthetic Environment (SE) representational semantics. We leverage upon existing standards for SE representation by '***Web-Enabling***' the SEDRIS Data Representation Model (DRM). We propose ***sed*Onto** - ***S****ynthetic* ***E****nvironment* ***D****ata Representation* ***Onto****logy*, an ontology to be used within the M&S domain for the representation of data pertaining to a SE. *sed*Onto is based on the SEDRIS DRM, which is a *ISO/IEC* standard widely adopted within the M&S community for the representation of data pertaining to a SE. The recently released *W3C* recommendation for the representation of web ontologies, namely the ***W****eb* ***O****ntology* ***L****anguage* (OWL), is utilized for representing *sed*Onto. OWL provides a *Web-Based* formalism for representing taxonomy/domain hierarchies that could be reasoned upon. It consists of a rich set of knowledge representation constructors and is based on a expressive class of ***Description Logics*** (DL) thereby facilitating formal reasoning over OWL described resources. We illustrate how OWL's domain description constructors can be used to reflect the SE semantics.

Applications of our High-Level Web-Ontology based SE representation are aplenty; However, for the purposes of this paper, we shall highlight Web based sharing of SE semantics - both '***Structural***' and '***Thing Level***', and the application of existing OWL/DL based reasoning systems and agent frameworks for performing terminological reasoning over SE objects.

1. Introduction

The application of Ontologies for solving interoperability problems has been widely recognised across multiple domains. Ontologies, by virtue of the shared conceptualization that they provide, may be communicated between people and application systems thereby facilitating interchange, interoperability and common understanding.

An ontology typically consists of a hierarchical description of important concepts in a domain, along with descriptions of the properties of each concept. The degree of formality employed in capturing these descriptions can be quite variable, ranging from natural language to logical formalisms, but increased formality and regularity clearly facilitates machine understanding [1].

The e***X***tensible ***M***odeling and ***S***imulation ***F***ramework (XMSF), which is defined as a composable set of standards, profiles and recommended practices for web-based modeling & simulation (M&S), envisages the use of XML-based markup languages, Internet technologies and Web Services to facilitate the emergence of a new generation of interoperable distributed M&S applications. The XMSF findings and recommendations report [2] on the challenges of Web-Based Modeling and Simulation suggests the use of ontologies to allow the definition and approval of complementary taxonomies that can be applied across multiple XMSF application domains. The emphasis in XMSF is placed on establishing concensual common meaning not only within groups, but to be truely useful, also among groups. As specified in the XMSF charter, this would involve the use of such XML based technologies such as XML Schema, RDF, DAML+OIL etc.

This paper presents ***sed*Onto** — ***S**ynthetic **E**nvironment **D**ata Representation **Onto**logy*, an ontology to be used within the M&S domain for the representation of data pertaining to a SE. We leverage upon existing standards for SE representation by '***Web-Enabling***' the SEDRIS Data Representation Model (DRM), which is a *ISO/IEC* standard widely adopted within the M&S community for the representation of data pertaining to a SE. The recently released *W3C* recommendation for the representation of web ontologies, namely the ***W**eb **O**ntology **L**anguage* (OWL), is utilized for representing *sed*Onto. Since OWL is a relatively newer W3C recommendation [3] developed as a successor to DAML+OIL, our research completely involves the use of OWL. The novelty of the approach suggested in this paper lies in:

1. The ***use of an Ontology*** to represent synthetic environment semantics thereby providing a source of shared and precisely defined terms that can be used as metadata.
2. Utilizing a high-level ***formal Description Logic based language***, namely OWL, to model the ontology instead of using a pure schema or interchange medium such as XML. The formal basis of the language makes it more accessible to automated processes thereby facilitating machine understanting. Moreover, being the language of the Semantic Web, its use for SE representation contributes towards the XMSF web-based modeling and simulation vision.
3. Leveraging upon an existing ***SE representation and interhange standard***, namely SEDRIS, as the basis of our ontology and in effect Web-Enabling it.

*sed*Onto is a part of a bigger project involving the automatic transformation of a SEDRIS based synthetic environment to a web-ontology based representation scheme. Postponing further discussion pertaining to the automated transformation to Section 5, henceforth we refer to it as ***STOWL*** - ***S***EDRIS ***To OWL*** Transform.

The rest of the paper is organized as follows: **Section 2** presents the background material for this paper that may be selectively read depending on the readers area of expertise. **Section 3**, which is the core of this paper, presents our work involving the construction of the SE representation ontology. Our approach here is to engage the reader in a illustrative walkthrough of the ontology construction process — mapping the SEDRIS DRM meta-level to corressponding ontological constructors in OWL followed by demonstrating the actual use of each of those constructors. Here, we also discuss some of the problems encountered and the approach taken to solve them. In **Section 4**, we present two of the most important applications envisaged of *sed*Onto - Web-Based sharing of SE semantics and Terminological reasoning over SE objects. Finally in **Section 5**, we conclude with a few remarks on future and work-in-progress.

2. Background

2.1. SEDRIS DRM

Synthetic ***E***nvironment ***D***ata ***R***epresentation and ***I***nterchange ***S***pecification (**SEDRIS**) technology is fundamentally about two key functions[4]: **Representation of environmental data** and the **Interchange of environmental data sets**. To accomplish the first, SEDRIS technology contains a Data Representation Model (DRM), augmented with an Environmental Data Coding Specification (EDCS) thereby capturing and communicating meaning and semantics. To quote the SEDRIS reference document:

"The EDCS provides a mechanism to specify the environmental "things" that a particular data model construct is intended to represent. That is, a "tree" could be represented alternatively as a <Point Feature>, an <Aggregate Geometry>, a <Data Table>, a <Model>, or some combination of these and other data modeling constructs from the DRM. Which of these the data modeler (i.e., the data provider of a SEDRIS transmittal) chooses is orthogonal to the semantics of the "thing" that is represented (and its location)."

For the second function, it is not enough to be able to clearly represent or describe the data, we must also be able to share such data with others in an efficient manner. For the interchange part, the SEDRIS Application Program Interface (SEDRIS API), the SEDRIS Transmittal Format (STF) and all the associated tools and utilities play the primary role. Note however that all these component technologies are still semantically coupled to the DRM.

The SEDRIS DRM is an object-oriented data representation model, and provides a unified method for describing all data elements, and their logical relationships, needed to express environmental data in a seamless manner across all environmental domains. The DRM is at the heart of SEDRIS technologies, and is based on object-oriented techniques, the characteristics of which are described using the Unified Modeling Language (UML) [5]. It consists of a large variety of object-oriented classes that allow the description of any environmental data, regardless of resolution, domain, or density. The combination of these classes and their relationships provides a rich, powerful, and expressive schema that can be thought of as the grammar of a language for describing environmental data. The SEDRIS DRM supports the definition of elements such as abstract and concrete classes, logical relationships such as association, generalization and aggregation. Usual N-by-N (multiplicity) between elements as well order and the direction (one-way or two-way) of association between classes too may be specified.

SEDRIS technologies, to achieve the objectives of broad use, have been standardized under the International Standards Organization (ISO) and International Electrotechnical Commission (IEC) and as *Standardization for Agreements* (STANAGS) for NATO use. Additional information and access to these technologies, standards and technical publications can be found at [6].

2.2. Description Logics (DL)

Description Logics is the most recent name for a family of knowledge representation formalisms that represent the knowledge of an application domain (the world) [7]. Description Logics are descended from the so-called 'Structured Inheritance Networks (SIN)' [8], which were introduced to overcome the ambiguities of the early semantic networks and frames. However, unlike its predecessors, DL based languages are equipped with a formal, logic-based semantics. As such, the emphasis of these languages is on reasoning services that allow one to infer implicitly represented knowledge from the knowledge that is explicitly contained in the knowledge base. Description Logics support inference patterns that occur in many applications of intelligent information processing systems, and which are also used by humans to structure and understand the world, for example: classification of concepts and individuals, identifying subconcept-superconcept relationships (called subsumption relationships in DL) between the concepts of a given terminology, and thus allow one to structure the terminilogy in the form of a subsumption/inheritance heirarchy.

The Knowledge Base (KB) of a typical DL based system comprises of two components, the TBox and the ABox. The TBox introduces the terminology, i.e., the vocabulary of an application domain, while the ABox contains assertions about named individuals in terms of this vocabulary. A DL system not only stores terminologies and assertions, but also offers services that reason about them. Typical reasoning tasks for a terminology (TBox) are to determine whether a decsription is satisfiable (i.e., non-contradictory) , or whether one description is more general than another one, that is, whether the first subsumes the second. Likewise, important problems for an ABox are to find out whether its set of assertions is consistent, that is, whether it has a model, and whether the assertions in the ABox entail that a particular individual is an instance of a given concept description.

Constructor	DL Syntax	Example
intersectionOf	$C_1 \sqcap \ldots \sqcap C_n$	Human $\sqcap$ Male
unionOf	$C_1 \sqcup \ldots \sqcup C_n$	Doctor $\sqcup$ Lawyer
complementOf	$\neg C$	$\neg$Male
oneOf	$\{x_1\} \sqcup \ldots \sqcup \{x_n\}$	{john} $\sqcup$ {mary}
allValuesFrom	$\forall P.C$	$\forall$hasChild.Doctor
someValuesFrom	$\exists P.C$	$\exists$hasChild.Lawyer
maxCardinality	$\leqslant nP$	$\leqslant$1hasChild
minCardinality	$\geqslant nP$	$\geqslant$2hasChild

Figure 1: OWL Class Constructors

2.3. OWL

OWL is an ontology language specifically designed for use on the semantic Web; it exploits existing web standards (XML and RDF), adding the familiar ontological primitives of object and frame based systems, and the formal rigour of a very expressive Description Logic (DL) [9]. The logical basis of the language means that reasoning services can be provided in order to make OWL described resources more accessible to automated processes thereby allowing one to infer implicitly represented knowledge from the knowledge that is explicitly contained in the knowledge base. From a formal point of view, OWL can be seen to be equivalent to a very expressive description logic, with a OWL ontology corresponding to a DL ***terminology* (Tbox)**.

Axiom	DL Syntax	Example
subClassOf	$C_1 \sqsubseteq C_2$	Human $\sqsubseteq$ Animal $\sqcap$ Biped
equivalentClass	$C_1 \equiv C_2$	Man $\equiv$ Human $\sqcap$ Male
disjointWith	$C_1 \sqsubseteq \neg C_2$	Male $\sqsubseteq$ $\neg$Female
sameIndividualAs	$\{x_1\} \equiv \{x_2\}$	{President_Bush} $\equiv$ {G_W_Bush}
differentFrom	$\{x_1\} \sqsubseteq \neg\{x_2\}$	{john} $\sqsubseteq$ $\neg${peter}
subPropertyOf	$P_1 \sqsubseteq P_2$	hasDaughter $\sqsubseteq$ hasChild
equivalentProperty	$P_1 \equiv P_2$	cost $\equiv$ price
inverseOf	$P_1 \equiv P_2^-$	hasChild $\equiv$ hasParent$^-$
transitiveProperty	$P^+ \sqsubseteq P$	ancestor$^+$ $\sqsubseteq$ ancestor
functionalProperty	$\top \sqsubseteq \leqslant 1P$	$\top \sqsubseteq \leqslant$1hasMother
inverseFunctionalProperty	$\top \sqsubseteq \leqslant 1P^-$	$\top \sqsubseteq \leqslant$1hasSSN$^-$

Figure 2: OWL Axioms

OWL comes with three sub-languages, viz OWL Lite, OWL DL and OWL Full, which differ in terms of various constructs offered and/or the flexibility of constructor usage. OWL DL and OWL FULL essentially offer the same constructs. However, their diference lies in restrictions on the use of some of those features and on the use of RDF features. Fig. 1 and Fig. 2[1] illustrate the various class constructors and axioms provided by OWL that may be used to com-

1 Albeit anonymous, Figure 1 and 2 have been adapted from external sources

Table 1: DRM to OWL Construct Mapping

DRM Meta Level	OWL Equivalent
Class	owl:Class
Abstract Class	No direct support
Association Relationship	
One-Way Association	owl:ObjectProperty
Attributes	owl:DatatypeProperty and owl:ObjectProperty
Link Classes	owl:ObjectProperty
Aggregation Relationship	owl:ObjectProperty
Multiplicity	
Specific Value	owl:cardinality
Minimum Value	owl:minCardinality
Maximum Value	owl:maxCardinality
No Multiplicity Restrictions	Default OWL Semantics
Ordered Annotation for a Relationship	No direct support
Inheritance Relationship	owl:subClassOf

pose primitive and complex class expressions. Our motivation for using the OWL DL subset as the representation formalism for the SE ontology is the fact that OWL DL has desirable computational properties - all conclusions are guaranteed to be computable and all computations will finish in finite time [10]. Tool builders have already developed powerful reasoning systems that support ontologies constrained by the restrictions required by OWL DL, the best example here being RACER [11]. For the formal definitions of the diferences between OWL Full and OWL DL, we refer interested readers to the Semantics and Abstract Syntax document [12].

3. *sed*Onto: Synthetic Environment Data Representation Ontology

3.1. DRM to OWL Construct Mapping

Table 1 illustrates the mapping between the UML notation used by the SEDRIS DRM to the constructs provided by the OWL language. Note that the mapping utilizes only a subset of constructs provided by OWL. Moreover, not all DRM constructs, viz abstract classes and ordered annotations for relationships, are directly supported by OWL. Later in the section, we illustrate how such unsupported elements are accounted for in the mapping process.

3.2. Representing DRM Semantics in OWL

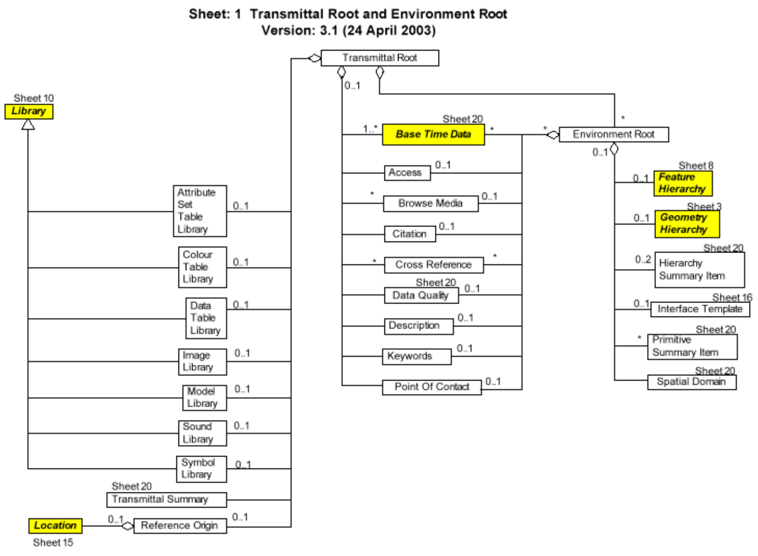


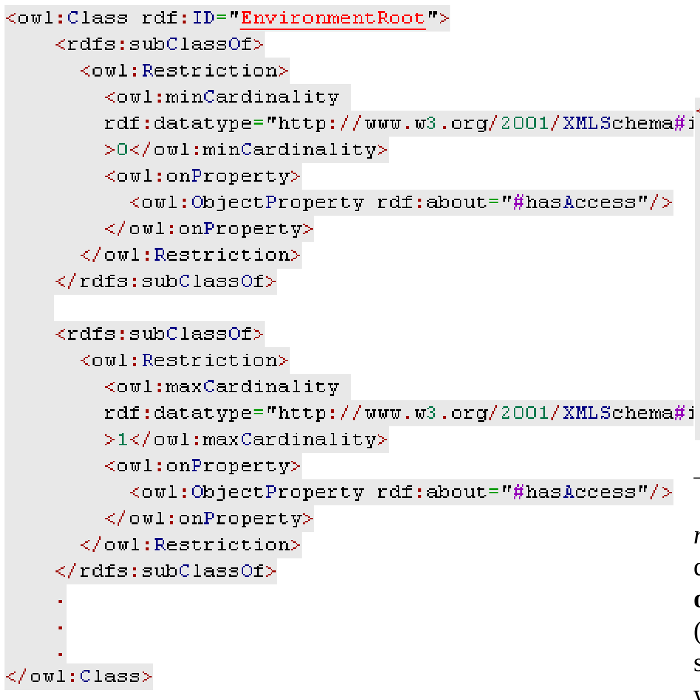
Figure 3: SEDRIS DRM Sheet 1

What follows is an illustration involving the mapping of a small portion of the SEDRIS DRM sheet number 1 shown in Fig. 3[2]. Note that the DRM is quite exhaustive and consists of a total of 23 similar sheets. The emphasis of the illustration is on highlighting the usage of the rather important OWL primitives for representing DRM semantics.

3.2.1. Aggregation Relationships An aggregation relationship, referred to as '*hasComponent*' is represented as a type of *owl:ObjectProperty*. All aggregation relationships in the DRM are then defined to be subproperties of this '*hasComponent*'. Moreover, we also define a '*hasAggregate*' relationship to be an inverse of '*hasComponent*'. Fig. 4, which exemplifies the use of aggregation relationships and cardinality restrictions, defines the *Environment Root* class to be an aggregation of 0 or 1 *Access* objects. Likewise, the snippet in Fig. 5 defines the *Environment Root* to consist of *exactly 1 Spatial Domain*.

3.2.2. Attributes The DRM is modelled along object oriented principles using UML. As such, there is direct support for declaring *Attributes* or the DRM *Field Elements* for various DRM classes. OWL is not object oriented and the attributes cannot be linked to classes directly. Instead, as ex-

2 Source: http://www.sedris.org/drm.htm


```
<owl:Class rdf:ID="EnvironmentRoot">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:minCardinality
        rdf:datatype="http://www.w3.org/2001/XMLSchema#i
        >0</owl:minCardinality>
        <owl:onProperty>
          <owl:ObjectProperty rdf:about="#hasAccess"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>

    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:maxCardinality
        rdf:datatype="http://www.w3.org/2001/XMLSchema#i
        >1</owl:maxCardinality>
        <owl:onProperty>
          <owl:ObjectProperty rdf:about="#hasAccess"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
    .
    .
    .
</owl:Class>
```

Figure 4: *Environment Root* is an aggregation of exactly 0 or 1 *Access* objects

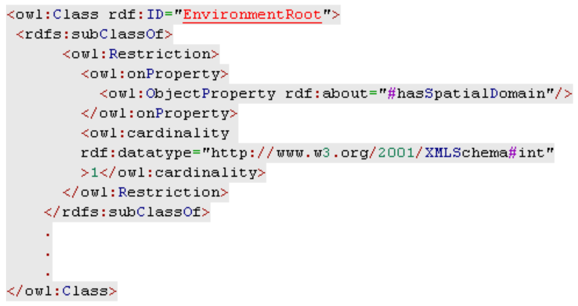
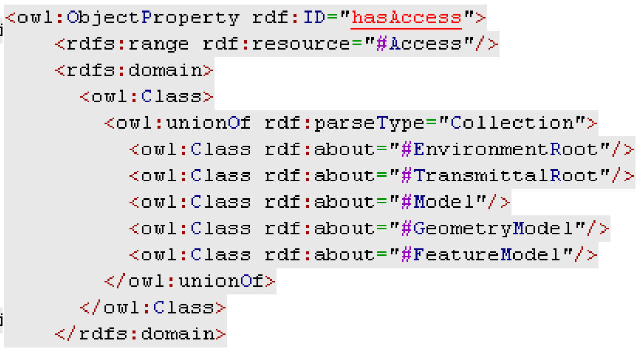

```
<owl:Class rdf:ID="EnvironmentRoot">
 <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:about="#hasSpatialDomain"/>
        </owl:onProperty>
        <owl:cardinality
        rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:cardinality>
      </owl:Restriction>
    </rdfs:subClassOf>
    .
    .
    .
</owl:Class>
```

Figure 5: *Environment Root* has exactly one *Spatial Domain*

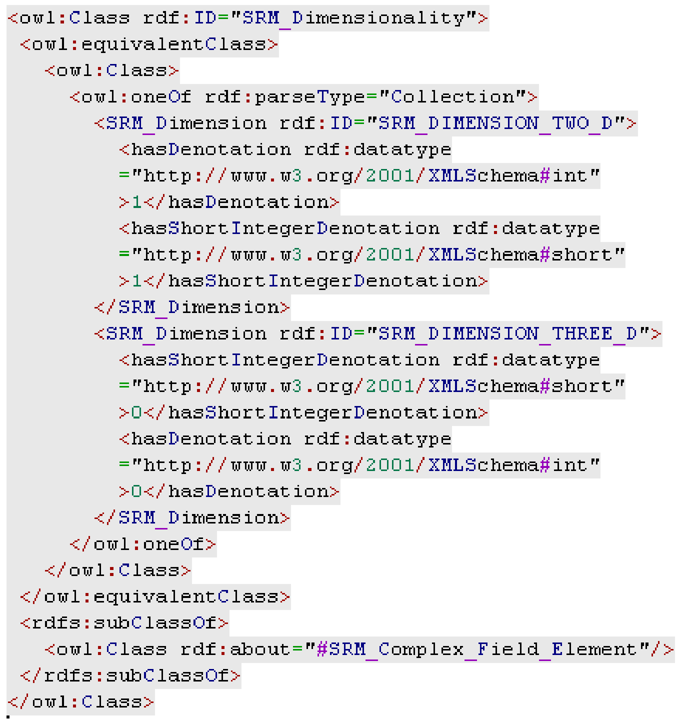
emplified in Fig. 6, we rely on a more formal approach for representing a particular class's field element as a binary relation or property between the class and the field element or attribute. Although not evident in Fig. 6, notice how the *attribute* too has to be declared as a *class* in order to achieve such a definition. Overall, Fig. 6 illustrates the use of object properties, domain & range restrictions and the OWL unionOf class constructor.

3.2.3. Enumerations Enumerated values play a very important role within the DRM and (as their purpose is) are utilized for constraining the values of various *Field Elements* (or attributes) of the respective DRM classes. OWL directly supports the definition of such types using the **owl:oneOf** construct. The following example illustrates our (somewhat roundabout) ***two-step*** approach for the representation of enumerated types in OWL: As a ***first step***, we classify the enumerated type depending upon its location in the DRM class dictionary. For example, if the type is *SRM_Dimensionality* from the Spatial Reference Model, we create a class called *SRM_Dimension* and all its instances, with each instance corresponding to the respective permissible enumerant. Moreover, we also classify the type according to its location within the taxanomy, which is *SRM_Globals* in this case. Each of these instances also has some form of an integral denotation depending on its definition in the DRM. The ***second step*** involves the creation of another class with a name directly corressponding to its SRM counterpart, i.e., *SRM_Dimensionality* under the category *SRM_Field_Element*. We define this class using the **owl:oneOf** construct, which in the knowledge engineering parlance refers to definition of the class's extension by exhaustive enumeration. Notice (in Fig. 7) how this definition involves the creation of an equivalence relation between *SRM_Dimensionality* (from the *SRM_Field_Element category*) to SRM_Dimension (in the *SRM_Globals category*). The most important advantage of this approach is the clear separation DRM field elements, DRM globals and their actual denotations (to integral values) into different and related categories thereby making the ontology more modular.

Figure 6: Attributes as Binary Relationships


```
<owl:ObjectProperty rdf:ID="hasAccess">
    <rdfs:range rdf:resource="#Access"/>
    <rdfs:domain>
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <owl:Class rdf:about="#EnvironmentRoot"/>
          <owl:Class rdf:about="#TransmittalRoot"/>
          <owl:Class rdf:about="#Model"/>
          <owl:Class rdf:about="#GeometryModel"/>
          <owl:Class rdf:about="#FeatureModel"/>
        </owl:unionOf>
      </owl:Class>
    </rdfs:domain>
```

3.2.4. Ordered Relationships As mentioned previously, there is no direct language support in OWL for creating ordered aggregation relationships. However, [13] suggests a simple approach involving the utilization of the *rdf:List* element for the definition of such relationships. The technique here is to make the aggregation relationship a functional one using the *owl:FunctionalProperty* con-

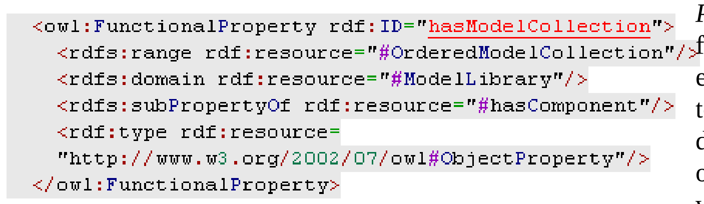


```
<owl:Class rdf:ID="SRM_Dimensionality">
 <owl:equivalentClass>
   <owl:Class>
     <owl:oneOf rdf:parseType="Collection">
       <SRM_Dimension rdf:ID="SRM_DIMENSION_TWO_D">
         <hasDenotation rdf:datatype
         ="http://www.w3.org/2001/XMLSchema#int"
         >1</hasDenotation>
         <hasShortIntegerDenotation rdf:datatype
         ="http://www.w3.org/2001/XMLSchema#short"
         >1</hasShortIntegerDenotation>
       </SRM_Dimension>
       <SRM_Dimension rdf:ID="SRM_DIMENSION_THREE_D">
         <hasShortIntegerDenotation rdf:datatype
         ="http://www.w3.org/2001/XMLSchema#short"
         >0</hasShortIntegerDenotation>
         <hasDenotation rdf:datatype
         ="http://www.w3.org/2001/XMLSchema#int"
         >0</hasDenotation>
       </SRM_Dimension>
     </owl:oneOf>
   </owl:Class>
 </owl:equivalentClass>
 <rdfs:subClassOf>
   <owl:Class rdf:about="#SRM_Complex_Field_Element"/>
 </rdfs:subClassOf>
</owl:Class>
```

Figure 7: Enumerated Classes

struct and restrict its range to the *rdf:List* element. Usage of *owl:FunctionalProperty* will permit a single value for the property whereas *rdf:List* will act as the container for the ordered elements. Fig. 8 illustrates the scenario from SEDRIS DRM Sheet 2 to express the relatisonship - *Model Library* is a ordered collection of *Model* objects. Note that we use OrderedModelCollection, which is a subclass of rdf:List, as the range of the functional property.

Figure 8: Ordered Aggregation Relationship



```
 <owl:FunctionalProperty rdf:ID="hasModelCollection">
   <rdfs:range rdf:resource="#OrderedModelCollection"/>
   <rdfs:domain rdf:resource="#ModelLibrary"/>
   <rdfs:subPropertyOf rdf:resource="#hasComponent"/>
   <rdf:type rdf:resource=
   "http://www.w3.org/2002/07/owl#ObjectProperty"/>
 </owl:FunctionalProperty>
```

3.2.5. Abstract Classes The notion of an abstract class as used in the context of sofware engineering is different from the one used in the context of knowledge representation. In software engineering, an abstract class is a class that does not have any direct instances whereas in knowledge representation, abstract classes refer to classes that have other classes as instances. OWL, being a knowledge representation language, does not support the former view and therefore it is not possible to prevent a class from having instances.

Within the scope of *sed*Onto, our notion of an abstract class is based on the software enginering view of it. However, OWL's inability to represent abstract classes is hardly a problem for *sed*Onto. This is because any instantiation of *sed*Onto (and hence the need to prevent certain class instantiations) will be governed by an automated transformation process (***STOWL***, see Section 5). STOWL can strictly regulate the instantiation process making sure none of the abstract classes from the SEDRIS DRM get directly instantiated.

3.3. Thing-Level Semantics and the EDCS

There are essentially two two main aspects to SE semantics - ***Structural*** and ***Thing level***. As mentioned previously in sub-section 2.1, in SEDRIS the DRM is responsible for the former whereas the latter is covered by complementing the DRM with a data coding specification; namely EDCS. The concept descriptions in the EDCS can be mapped in a straight forward manner to OWL, as was done for the classes and relationships from the DRM. Mapping the entire concept dictionary that makes up the EDCS shall be out of our research agenda until the implementation of STOWL (see Section 5) has been successfully achieved. For now, we simply map a small portion of the EDCS thereby serving our demonstrative purposes.

3.4. Implementation

The information provided in this section is only informative in that end users of *sed*Onto do not need to know these details unless they intend to modify or extend *sed*Onto.

We have utilised version 3.1 of the SEDRIS DRM and *Protege*. Protege is a open-source development environment for ontologies and knowledge based systems [14]. It has an excellent environment with a extensible plugin based architecture and support for various *Sematic Web* related standards. Although protege can be used in a plethora of ways, our usage has been restricted to using its OWL plugin [15], which enables the construction and manipulation of OWL ontologies. The OWL plugin allows users to load and save OWL and RDF ontologies, edit and visualize OWL classes and their properties, define logical class charecteristics as complex class expressions, execute reasoners such as DL classifiers and edit individuals for Semantic Web markup.

In addition to the flexibility and ease of use offered by Protege and its associated OWL plugin, its availability as a open source environment too may prove to be beneficial in the long run. Protege's large user community ensures that timely support services are offered and that new product updates are always in the offering. Snapshots from the ontology development process using Protege and its OWL plugin have been included in the appendices.

4. Applications

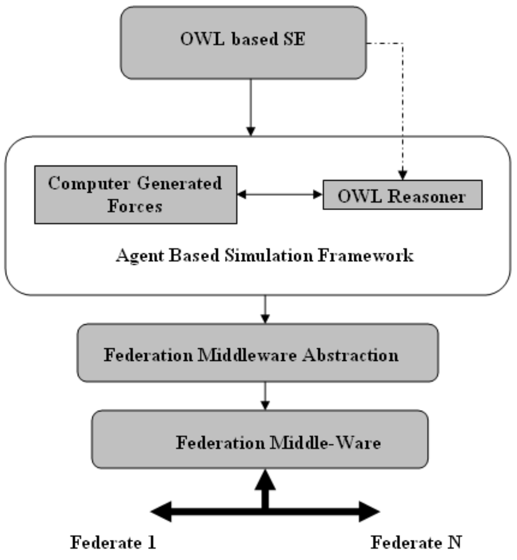

OWL based SE
Computer Generated Forces
OWL Reasoner
Agent Based Simulation Framework
Federation Middleware Abstraction
Federation Middle-Ware
Federate 1
Federate N

Figure 9: Example Application Scenario

Development of *sed*Onto is the first phase of our research. Work is in progress to implement ***STOWL*** (see Section 5), which will actually automate the process of converting a SEDRIS transmittal to a OWL (& *sed*Onto) based representation scheme. The applications discussed herein make sense only in the context of both *sed*Onto and *STOWL* taken together.

4.1. Web Based Sharing of SE Semantics

OWL exploits the power of the Web by utilizing XML/RDF as its base representation and intercange medium. We believe that this facility can go a long way in the sharing of independently developed SE's in a distributed simulation environment. Consider the scenario depicted in Fig. 9 consisting of an arbitrary number of federates participatng in a distributed simulation. Also, assume that HLA/RTI is being utlised as the federation middleware. HLA requires that individual federates be described by a object model, the Object Model Template (HLA OMT), which identifies all data exchanged by the federates at runtime [16]. The HLA OMT uses a XML based Data Interchange Format (DIF) for representing data pertaining to a federate/federation and for intialising its Run-Time Infrastructure (RTI). Since every OWL based specification is a valid XML document, the ***mapping of SE objects and their attributes to the OMT can be automated*** in a straight forward manner. Notice how this would also result in loss of semantics since XML purely serves interchange purposes. However, if the semantic information needs to be preserved, a target OWL based representation for the OMT will have to be used. Interestingly, as reported in [17], such an OWL based OMT has already been implemented. Notice how such an automation decouples the federate from the federation middleware (see Fig. 9) thereby acting as a ***Federation Middleware Abstraction Layer***.

4.2. Terminological Reasoning over SE Objects

The SEDRIS DRM is a conceptual model represented using UML. Its intent is to standardize SE representation semantics by providing the building blocks necessary to represent any SE pertaining to any domain. By mapping the DRM to the OWL language, we make explicit the structural semantics of the DRM using a language, which unlike UML is inherently suitable to do so. The OWL ontology based representation scheme can exploit the qualitative information that is present in SE. In the following, we provide a succinct (and informal) description of the various inference patterns supported over OWL described resources.

Tbox Inferences

- **Subsumption**: The subsumption inference task is to determine which out of two concept descriptions is a more general one. The more general one is said to subsume the more specific concept description.
- **Satisfiability**: The satisfiability task determines whether a concept has a non-empty extension, i.e., the Abox admits atleast one individual satisfying the class axioms for that concept.
- **Equivalence**: The equivalence task determines whether two concepts have the same extension in the Abox.
- **Disjointness**: The disjointness task determines whether the intersection of the extensions of two concepts is NULL.

Abox Inferences

- **Instance Checking**: The instance checking task is to determine whether a given concept is an instance or belongs to a particular class.

- **Consistency**: Verify whether every concept in the TBOX admits atleast one individual.
- **Realization**: Find the most specific concept from the TBOX that an individual is an instance of.
- **Retrieval**: Find the individuals from the ABOX that are instances of a given concept from the TBOX.

An existing OWL based reasoner, for instance RACER [11], can be integrated within a agent based simulation framework (see Fig. 9) with minimal effort so as to utilize the above mentioned Tbox and Abox reasoning services. Indeed, terminological reasoning coupled with other forms of reasoning within the agent framework can significantly enhance the much sought after intelligent behaviour of autonomous entities with the simulation system.

5. Future Work

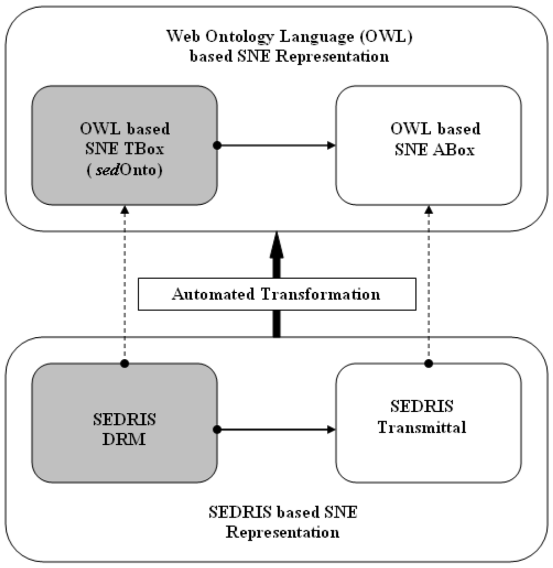

Web Ontology Language (OWL) based SNE Representation
OWL based SNE TBox (*sed*Onto)
OWL based SNE ABox
Automated Transformation
SEDRIS DRM
SEDRIS Transmittal
SEDRIS based SNE Representation

Figure 10: *STOWL* - STF to OWL Transform

The second stage of our research involves investigating the issues pertaining to the automation of the transformation of a SEDRIS based SE or SEDRIS transmittal to a Web-Ontology based form. As mentioned previously, we refer to the transformation as ***STOWL*** - ***S***EDRIS ***To OWL*** Transform. Obviously, the resulting OWL based representing scheme will be based on our synthetic environment data representation ontology and in actuality shall be an instantiation of it. Specifically, sedOnto represents the 'Terminology' or TBOX whereas the automatically transformed STF represents the 'Assertions' or ABOX. To make things clear, the precise situation is illustrated in Fig. 10. Currently, work is in progress to implement STOWL.

6. Conclusion

The application of Ontological formalisms as the basis of Synthetic Environment (SE) *representational semantics* has been proposed. From a design perspective, we have demonstrated how techniques from the knowledge engineering domain could be applied for the representation of a synthetic environment. By mapping the SEDRIS DRM to the OWL language, we make explicit the structural semantics of the DRM using a language, which unlike UML is inherently suitable to do so. The logical basis of the language means that automated reasoning procedures can be utilized to perform reasoning over SE objects. A case was made in this regard by illustrating the various (ontological) inference patterns supported over OWL described resources. Moreover, it was also demonstrated how such a high-level SE representation scheme could facilitate the web-based sharing of SE object semantics in a distributed simulation environment.

References

[1] Ian Horrocks. Daml+oil: A reasonable ontology language. In *Proceedings of EDBT-02, Volume 2287 of LNCS*, pages 2–13, 2002. 1

[2] Don Brutzman, Mike Zyda, Mark Pullen, and Katherine Morse. XMSF 2002 Findings and Recommendations Report: Technical challenges workshop and strategic opportunities symposium. Technical Report: XMSF Basis, October 2002. 1

[3] W3C. *OWL Web Ontology Language: A W3C Recommendation*. http://www.w3.org/News/2004. 1

[4] DMSO. Synthetic environment data representation and interchange specification. https://www.dmso.mil/public/transition/sedris/. 2.1

[5] Jim Rumbaugh, Ivar Jacobson, and Grady Booch. *The Unified Modeling Language Reference Manual*. Addison-Wesley, 1998. 2.1

[6] SEDRIS. The source for environmental representation and interchange. http://www.sedris.org. 2.1

[7] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider. *The description logic handbook: theory, implementation, and applications*. Cambridge University Press, 2003. 2.2

[8] Ronald Brachman. What's in a concept: Structural foundations for semantic networks. *International Journal Of Man Machine Studies*, 9(2):127–152, 1977. 2.2

[9] I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for very expressive description logics. *Logic Journal of the IGPL*, 8(3):239–264, 2000. 2.3

[10] W3C. *OWL Web Ontology Language Guide*. http://www.w3.org/TR/owl-guide/, February 2004. 1

[11] RACER. Racer: Renamed abox and concept expression reasoner. http://www.sts.tu-harburg.de/ r.f.moeller/racer/. 1, 4.2

[12] W3C. *OWL Web Ontology Language Semantics and Abstract Syntax*. http://www.w3.org/TR/2004/REC-owl-semantics-20040210/. 2.3

[13] Protege Website. Frequently asked questions. http://protege.stanford.edu/plugins/owl/protege-owl-faq.html. 3.2.4

[14] Protege. An ontology and knowledge base editor. http://protege.stanford.edu. 3.4

[15] Protege OWL Plugin. Ontology editor for the semantic web. http://protege.stanford.edu/plugins/owl/index.html. 3.4

[16] Simulation Interoperability Standards Committee. IEEE Std 1516.2: IEEE Standard For Modeling and Simulation High Level Architecture (HLA) - Object Model Template (OMT) Specification. Approved 21 September 2000. 4.1

[17] Paul F. Reynolds Jr., Arsalan Tavakoli, and David Chu. Richer semantic representations of simulations using daml/owl. *Spring Simulation Interoperability Workshop*, 2004. 4.1

Author Biographies

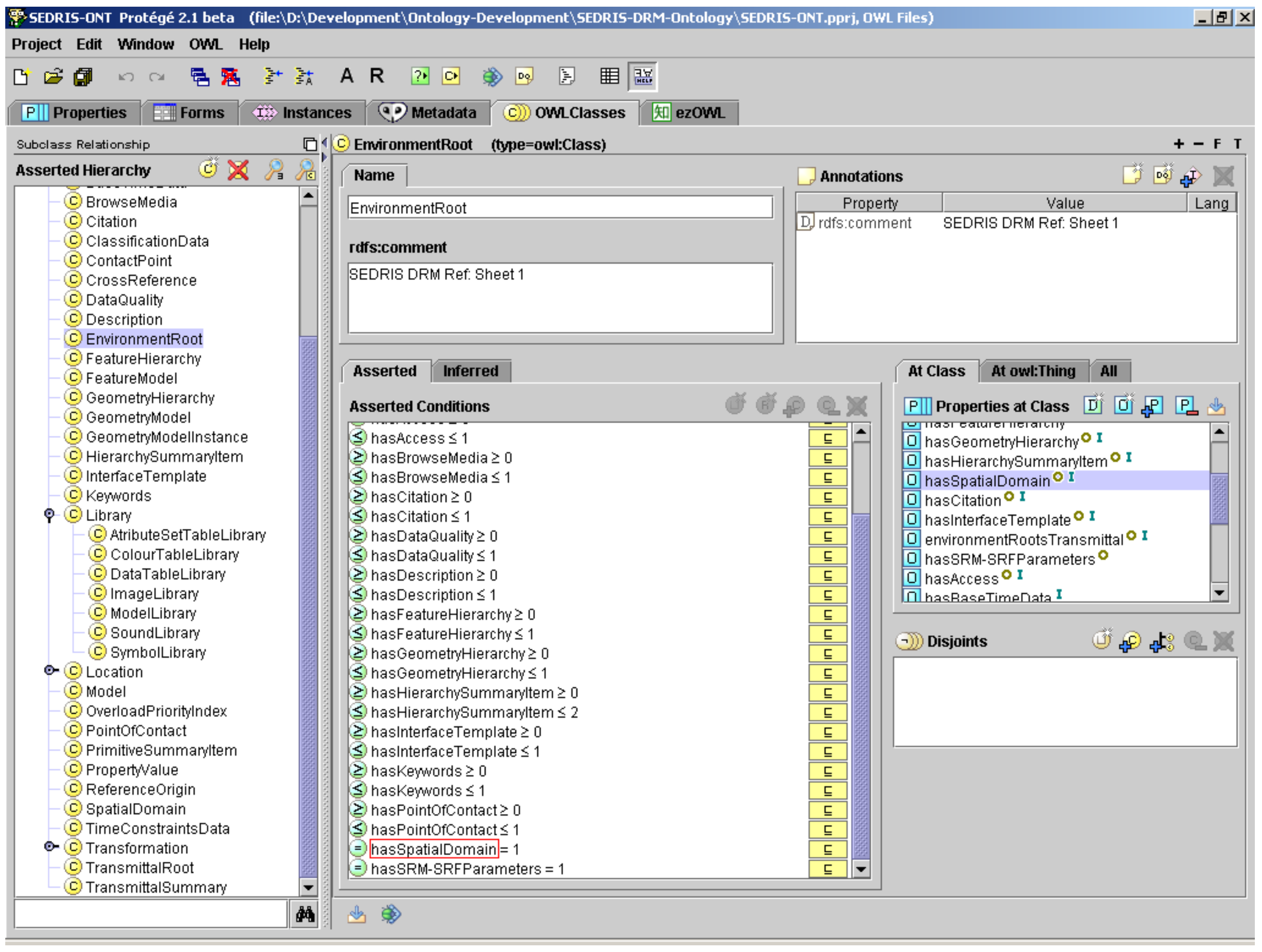
MEHUL BHATT is studying for a Ph.D. with the Computer Science department at La Trobe University, Australia. He holds a *Bachelor of Commerce & Economics* degree from Mumbai University (India) and a *Master of Information Technology* degree from La Trobe University (Australia). His honors thesis investigated the use of High-Level Logic based Cognitive Support in Dynamic Environments, using Soccer Simulation as an exemplar. He has also published in the area Ontology based knowledge representation and Parallel & Distributed approaches to sub-ontology extraction. His current research interests include: Parallel & Distributed Simulation Systems, Simulation Interoperability and Knowledge Representation & Reasoning issues pertaining to Synthetic Environments in particular and Simulation Systems in general. He can be contacted at: **mbhatt@cs.latrobe.edu.au**

Dr. WENNY RAHAYU is a Senior Lecturer at the Computer Science Department, La Trobe University, Australia. Dr. Rahayu has been actively working in the areas of database design and implementation covering object-relational databases, web and e-commerce databases, semi-structured databases and XML, data warehousing, and parallel databases. She has worked with industry and expertise from other disciplines in a number of projects including bioinformatics databases, parallel data mining, and e-commerce catalogues. Her PhD thesis has been awarded the Best PhD Thesis in Australia by the Computer Science Association of Australia in 2001. She has been invited to give seminars in the area of e-commerce databases in a number of international institutions. She has published two books and over 35 papers in international journals and conferences. She can be contacted at: **wenny@cs.latrobe.edu.au**

Dr. GERALD STERLING is a senior engineer in the Air Operations Division of the Systems Sciences Laboratory (SSL). SSL is a laboratory of the Defence Science and Technology Organisation (DSTO) in Melbourne, Australia. In the Crew Environments and Training Branch, his more recent research activity has focused upon the application of synthetic environments to training and to the enhancement of sensor system interfaces and operator environments. Over some 25 years of research and engineering, he has been engaged in simulator systems engineering & training system evaluation, air vehicle & engine control system engineering and development, aircraft & engine simulation for analysis, design & hardware and human in the loop evaluation, airworthiness management & regulation and teaching. He holds Doctor of Philosophy and Bachelor of Engineering degrees in Computer Systems and Aeronautical Engineering respectively, a Graduate Diploma in Computer Science and a Diploma of Education. He is also an occasional private pilot and sometime gliding pilot. He can be contacted at: Gerald.Sterling@dsto.defence.gov.au

Appendices - *sed*Onto Construction Using Protege-OWL

A. Protege-OWL Classes Tab



SEDRIS-ONT Protégé 2.1 beta (file:\D:\Development\Ontology-Development\SEDRIS-DRM-Ontology\SEDRIS-ONT.pprj, OWL Files)

EnvironmentRoot (type=owl:Class)

Name: EnvironmentRoot

rdfs:comment: SEDRIS DRM Ref: Sheet 1

Property	Value	Lang
rdfs:comment	SEDRIS DRM Ref: Sheet 1	

Asserted Hierarchy:

- BrowseMedia
- Citation
- ClassificationData
- ContactPoint
- CrossReference
- DataQuality
- Description
- EnvironmentRoot
- FeatureHierarchy
- FeatureModel
- GeometryHierarchy
- GeometryModel
- GeometryModelInstance
- HierarchySummaryItem
- InterfaceTemplate
- Keywords
- Library
  - AtributeSetTableLibrary
  - ColourTableLibrary
  - DataTableLibrary
  - ImageLibrary
  - ModelLibrary
  - SoundLibrary
  - SymbolLibrary
- Location
- Model
- OverloadPriorityIndex
- PointOfContact
- PrimitiveSummaryItem
- PropertyValue
- ReferenceOrigin
- SpatialDomain
- TimeConstraintsData
- Transformation
- TransmittalRoot
- TransmittalSummary

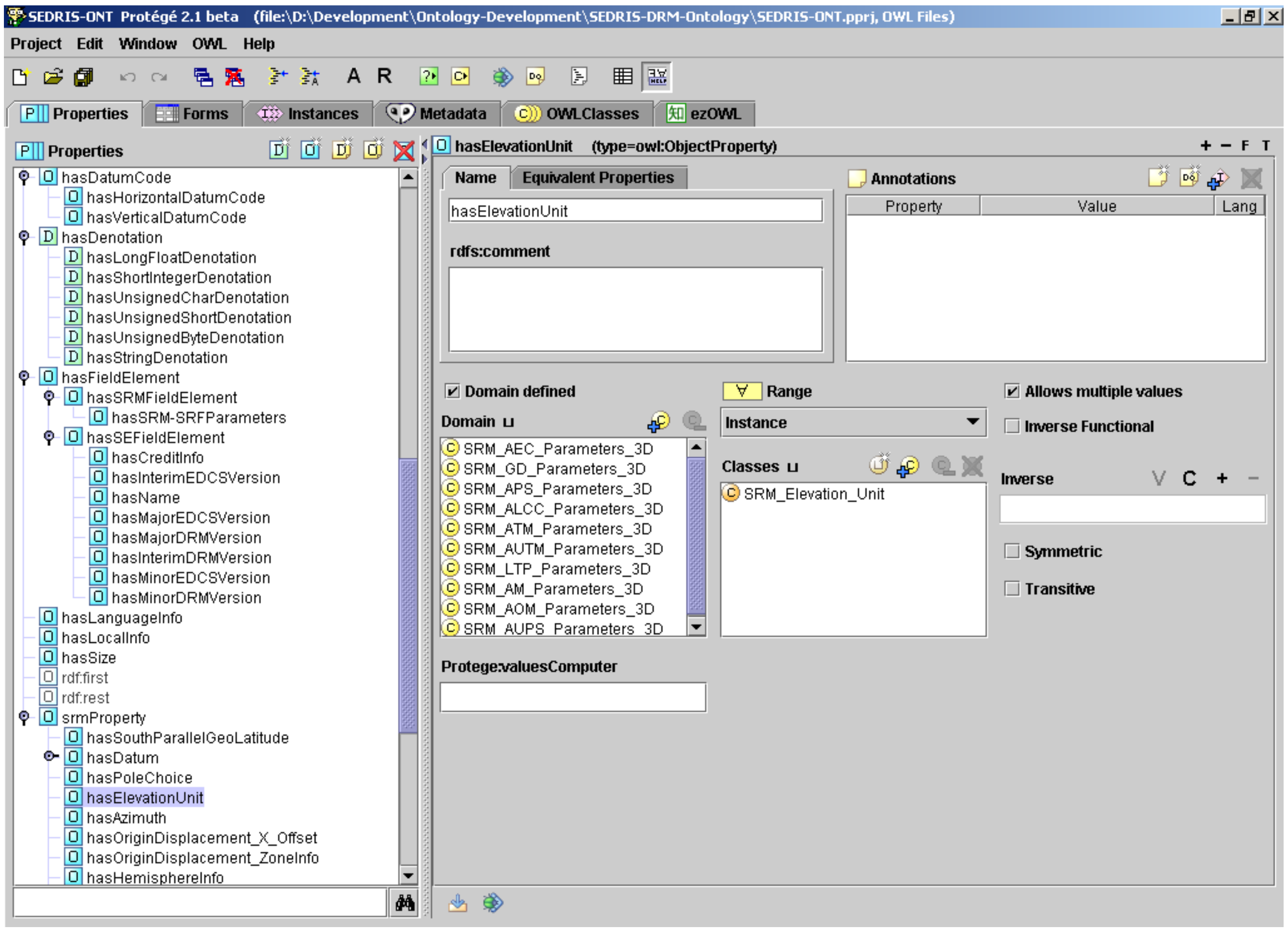
Asserted Conditions:

```
hasAccess ≤ 1
hasBrowseMedia ≥ 0
hasBrowseMedia ≤ 1
hasCitation ≥ 0
hasCitation ≤ 1
hasDataQuality ≥ 0
hasDataQuality ≤ 1
hasDescription ≥ 0
hasDescription ≤ 1
hasFeatureHierarchy ≥ 0
hasFeatureHierarchy ≤ 1
hasGeometryHierarchy ≥ 0
hasGeometryHierarchy ≤ 1
hasHierarchySummaryItem ≥ 0
hasHierarchySummaryItem ≤ 2
hasInterfaceTemplate ≥ 0
hasInterfaceTemplate ≤ 1
hasKeywords ≥ 0
hasKeywords ≤ 1
hasPointOfContact ≥ 0
hasPointOfContact ≤ 1
hasSpatialDomain = 1
hasSRM-SRFParameters = 1
```

Properties at Class:

- hasGeometryHierarchy
- hasHierarchySummaryItem
- hasSpatialDomain
- hasCitation
- hasInterfaceTemplate
- environmentRootsTransmittal
- hasSRM-SRFParameters
- hasAccess
- hasBaseTimeData

B. Protege-OWL Properties Tab



SEDRIS-ONT Protégé 2.1 beta (file:\D:\Development\Ontology-Development\SEDRIS-DRM-Ontology\SEDRIS-ONT.pprj, OWL Files)

Project Edit Window OWL Help

Properties | Forms | Instances | Metadata | OWLClasses | ezOWL

Properties

```
hasDatumCode
    hasHorizontalDatumCode
    hasVerticalDatumCode
hasDenotation
    hasLongFloatDenotation
    hasShortIntegerDenotation
    hasUnsignedCharDenotation
    hasUnsignedShortDenotation
    hasUnsignedByteDenotation
    hasStringDenotation
hasFieldElement
    hasSRMFieldElement
        hasSRM-SRFParameters
    hasSEFieldElement
        hasCreditInfo
        hasInterimEDCSVersion
        hasName
        hasMajorEDCSVersion
        hasMajorDRMVersion
        hasInterimDRMVersion
        hasMinorEDCSVersion
        hasMinorDRMVersion
hasLanguageInfo
hasLocalInfo
hasSize
rdf:first
rdf:rest
srmProperty
    hasSouthParallelGeoLatitude
    hasDatum
    hasPoleChoice
    hasElevationUnit
    hasAzimuth
    hasOriginDisplacement_X_Offset
    hasOriginDisplacement_ZoneInfo
    hasHemisphereInfo
```

hasElevationUnit (type=owl:ObjectProperty)

Name | Equivalent Properties

Name: hasElevationUnit

rdfs:comment

Annotations

Property	Value	Lang

☑ Domain defined

Domain ⊔
- SRM_AEC_Parameters_3D
- SRM_GD_Parameters_3D
- SRM_APS_Parameters_3D
- SRM_ALCC_Parameters_3D
- SRM_ATM_Parameters_3D
- SRM_AUTM_Parameters_3D
- SRM_LTP_Parameters_3D
- SRM_AM_Parameters_3D
- SRM_AOM_Parameters_3D
- SRM_AUPS_Parameters_3D

Range: Instance

Classes ⊔
- SRM_Elevation_Unit

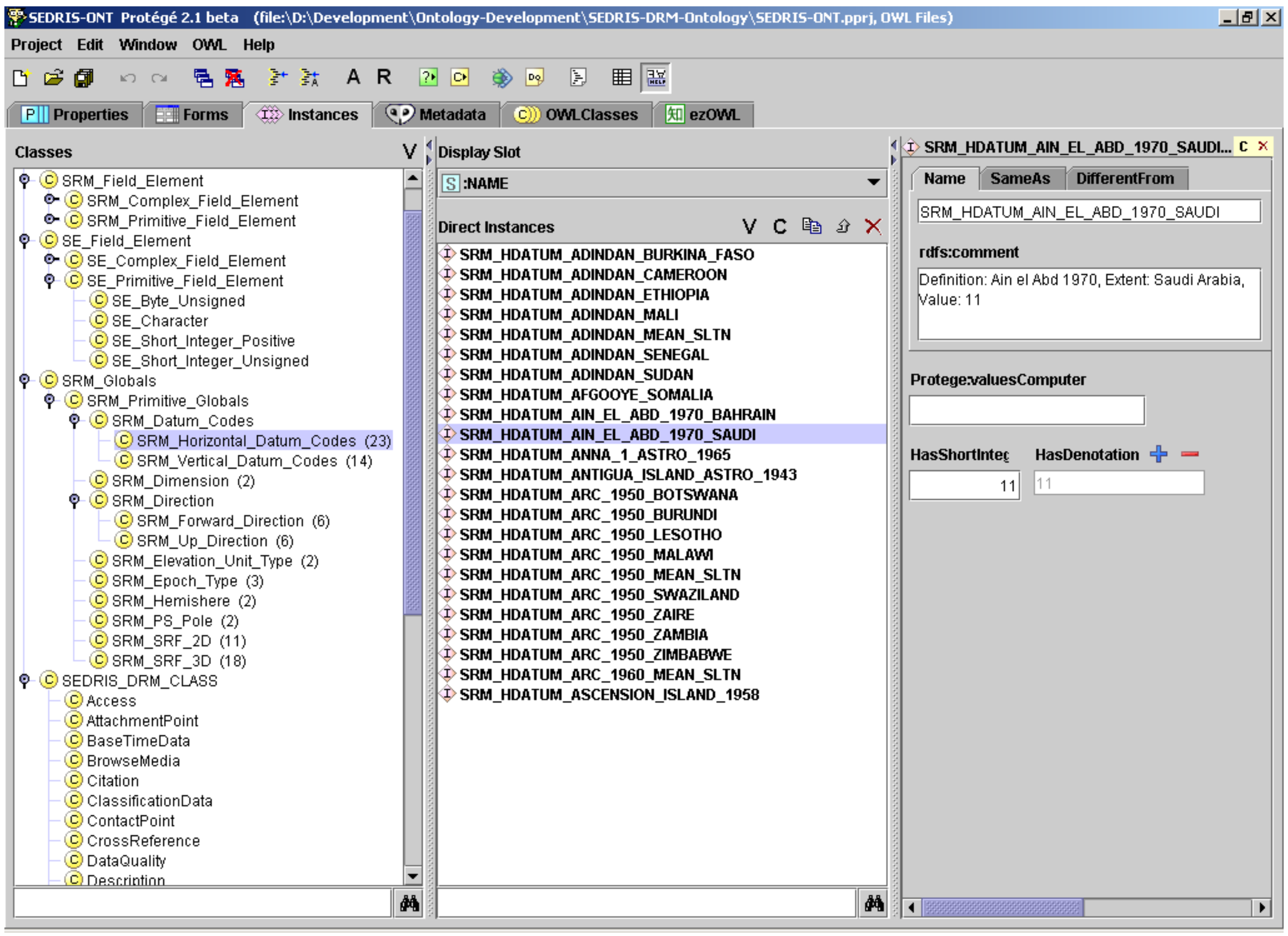
☑ Allows multiple values
☐ Inverse Functional

Inverse

☐ Symmetric
☐ Transitive

Protege:valuesComputer

C. Protege-OWL Instances Tab



Classes

- SRM_Field_Element
  - SRM_Complex_Field_Element
  - SRM_Primitive_Field_Element
- SE_Field_Element
  - SE_Complex_Field_Element
  - SE_Primitive_Field_Element
    - SE_Byte_Unsigned
    - SE_Character
    - SE_Short_Integer_Positive
    - SE_Short_Integer_Unsigned
- SRM_Globals
  - SRM_Primitive_Globals
    - SRM_Datum_Codes
      - SRM_Horizontal_Datum_Codes (23)
      - SRM_Vertical_Datum_Codes (14)
    - SRM_Dimension (2)
    - SRM_Direction
      - SRM_Forward_Direction (6)
      - SRM_Up_Direction (6)
    - SRM_Elevation_Unit_Type (2)
    - SRM_Epoch_Type (3)
    - SRM_Hemishere (2)
    - SRM_PS_Pole (2)
    - SRM_SRF_2D (11)
    - SRM_SRF_3D (18)
- SEDRIS_DRM_CLASS
  - Access
  - AttachmentPoint
  - BaseTimeData
  - BrowseMedia
  - Citation
  - ClassificationData
  - ContactPoint
  - CrossReference
  - DataQuality
  - Description

Display Slot: :NAME

Direct Instances

- SRM_HDATUM_ADINDAN_BURKINA_FASO
- SRM_HDATUM_ADINDAN_CAMEROON
- SRM_HDATUM_ADINDAN_ETHIOPIA
- SRM_HDATUM_ADINDAN_MALI
- SRM_HDATUM_ADINDAN_MEAN_SLTN
- SRM_HDATUM_ADINDAN_SENEGAL
- SRM_HDATUM_ADINDAN_SUDAN
- SRM_HDATUM_AFGOOYE_SOMALIA
- SRM_HDATUM_AIN_EL_ABD_1970_BAHRAIN
- SRM_HDATUM_AIN_EL_ABD_1970_SAUDI
- SRM_HDATUM_ANNA_1_ASTRO_1965
- SRM_HDATUM_ANTIGUA_ISLAND_ASTRO_1943
- SRM_HDATUM_ARC_1950_BOTSWANA
- SRM_HDATUM_ARC_1950_BURUNDI
- SRM_HDATUM_ARC_1950_LESOTHO
- SRM_HDATUM_ARC_1950_MALAWI
- SRM_HDATUM_ARC_1950_MEAN_SLTN
- SRM_HDATUM_ARC_1950_SWAZILAND
- SRM_HDATUM_ARC_1950_ZAIRE
- SRM_HDATUM_ARC_1950_ZAMBIA
- SRM_HDATUM_ARC_1950_ZIMBABWE
- SRM_HDATUM_ARC_1960_MEAN_SLTN
- SRM_HDATUM_ASCENSION_ISLAND_1958

SRM_HDATUM_AIN_EL_ABD_1970_SAUDI... C

Name: SRM_HDATUM_AIN_EL_ABD_1970_SAUDI

rdfs:comment: Definition: Ain el Abd 1970, Extent: Saudi Arabia, Value: 11

Protege:valuesComputer:

HasShortInteg: 11

HasDenotation: 11