

# Application of the Web Ontology Language to define management information specifications

Jorge E. López de Vergara, Víctor A. Villagrà, and Julio Berrocal

Departamento de Ingeniería de Sistemas Telemáticos

Universidad Politécnica de Madrid (DIT-UPM)

E.T.S.I. de Telecomunicación, Av. Complutense, s/n, 28040 Madrid, Spain.

{jlopez, villagra, berrocal}@dit.upm.es

**Abstract:** The eXtended Markup Language (XML) has emerged in the Internet world as a standard representation format, which can be useful to describe and transmit management information. However, XML formats alone do not give formal semantics to it. To solve this question, ontology languages based on the Resource Description Framework can be used to improve the expressiveness of the management information specifications. This paper presents an approach that uses an XML-based ontology language to define network and system management information. For this, the structures of the Web Ontology Language known as OWL are analyzed and compared to those used in management definitions, studying also the advantages that ontology languages can provide in this area.

**Keywords:** OWL, Network Management Information, Ontology, XML.

## 1 Introduction

Information definition is a key aspect in an Integrated Network Management model. Many languages have been specified with this purpose to date: Structure of Management Information (SMI), Guidelines for the Definition of Managed Objects (GDMO), Managed Object Format (MOF), etc. Meanwhile, the eXtended Markup Language (XML) has emerged in the Internet world as a standard representation format, enabling the definition, transmission, validation, and interpretation of information. Then, several approaches [1] have appeared that embrace this format to describe management information, including those proposed by the DMTF or the IETF Network Configuration Work Group.

Using XML in network management has some advantages. Many general tools and libraries that handle this format can be leveraged to develop management applications. At the same time, XML style sheets can be used to represent the information in a different way, which is useful in a scenario with multiple management technologies. However, Document Type Definitions (DTD) or XML Schemas only specify the information format, which does not give semantics to it. A machine that validates the syntax of a set of tags cannot deduce their meaning to infer the state of the network. Even, style sheet transformations are based on the syntax, and do not process the semantics contained in the XML documents.

Another step in this technology is the use of the Resource Description Framework (RDF) and its Schema (RDFS). They are XML-based languages that provide a limited way of defining structured sets of terms, with class hierarchies, and domain and range constraints. In fact, some ontology definition languages used in the Semantic Web are based on RDFS, but they go beyond so that machines can perform useful reasoning tasks. Then, these ontology languages can be used to improve the semantic expressiveness of the management information specifications. With them it is even possible to reason with the knowledge handled in the management tasks. This paper addresses these issues and proposes the use of OWL, the Web Ontology Language, as the management information language used by a manager. For this, current management information definitions have to be mapped to OWL allowing also an integration of different specifications. However, OWL has been defined for general purposes and lacks some common constructions of the management domain. So,

besides the mapping specification, it is proposed the needed OWL constructions to include all the components of the management information models.

The following section introduces ontologies as a way to specify management information. Next, OWL is studied showing its main characteristics. Then, a way to add facets that are common in management information languages is analyzed. After that, current works about the application of ontologies in the management area are presented. Finally, some conclusions are given.

## 2 Ontologies for management information specifications

Ontologies are one of the main approaches used in the scope of Knowledge Management and Artificial Intelligence to solve questions related to semantics, with current relevance in the Semantic Web. They describe an abstract model of a domain by defining a set of concepts, their taxonomy, interrelation and the rules that govern these concepts in a way that can be interpreted by machines.

Integrated management information models could be understood as a form of lightweight ontologies because they just define the information of the management domain, but they do not include axioms or constraints, present in heavyweight ontologies, which makes difficult to reason with them. Recently, some applications of ontologies have been presented to improve these information models:

- Management information languages can be compared in terms of their semantic expressiveness [2].
- Management information specifications can be translated into ontology languages, enhancing the expressiveness of the management languages with formal axioms and constraints so that a manager can reason with these definitions [2]. These translations are also useful to define a vocabulary for intelligent agents [3], [4].
- Management information defined in different languages can be integrated in a semantic way applying ontology techniques to merge and map different specifications taking into account their meaning [5].

Figure 1 shows how these applications can be useful in a multi-domain management scenario where different technologies are used to monitor and control the resources. In this case, all management specifications described with different information models can be merged in a single ontology, with sets of mapping rules to semantically translate the information, i.e. identifying the same concept from different definitions by extracting their meaning. At the same time, constraint constructions included in ontology languages can be used to enhance the semantics of the management information. They allow the inclusion of formal behavior rules in the information, so that an intelligent manager can check if the state of the system complies with those constraints.

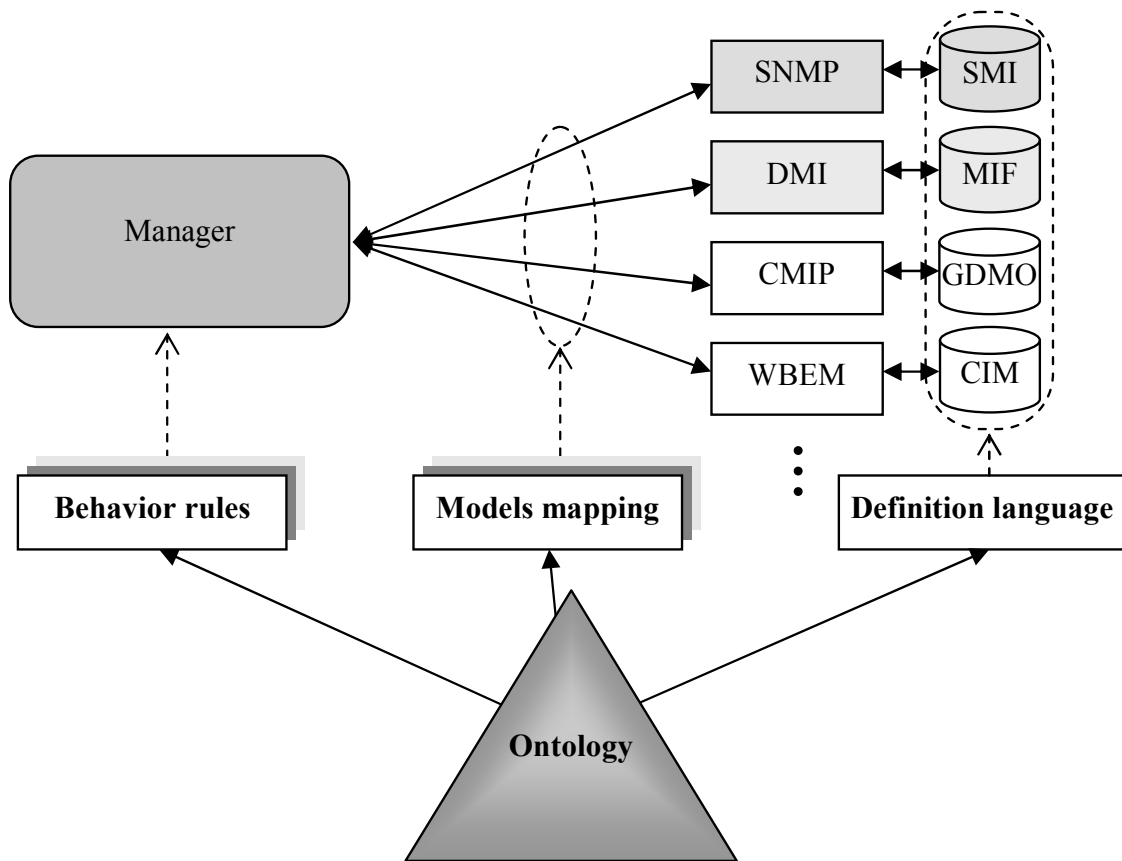


Figure 1. Ontology applications to network management information

Taking these applications as a starting point, next sections propose the use of a Semantic Web ontology language to define management information, which combines the advantages of XML and ontologies to improve the management of networks, systems, applications, and services. The main benefit of this approach is given by the fact that ontology languages related to the Semantic Web have a considerable amount of users and available tools. Another advantage compared to management information languages is that they have been formalized, so that its semantics is sound and complete and it can be used by intelligent systems. At the same time, some general common ontologies can be reused for the definition of information, such as those that address measurement units. On the other hand, they are not specific for management and do not have constructions to define all the constructions common in management. They also do not allow the definition of methods or operations, although management information bases do not include many of them.

### 3 OWL: the Web Ontology Language

OWL [6] (not an acronym) is the Web Ontology Language proposed from the World Wide Web Consortium for the definition of ontologies in the Semantic Web, being currently in the last specification stage. Three flavors have been defined based on RDF and RDFS to deal with different levels of expressiveness, as shown in Figure 2: OWL Lite, for simple constraints; OWL DL, based on Description Logics, with computational completeness and decidability; and OWL Full, with the maximum expressiveness, but without computational guarantees.

OWL Full
OWL DL
OWL Lite
RDFS
RDF
XML

**Figure 2. Levels of semantic expressiveness**

Using the analysis framework of [7], it can be said that OWL is a very complete ontology language, because it includes the following characteristics:

- Concepts: it can define:
  - Partitions, as well as class documentation.
  - Attributes, although it is not possible to distinguish between class and instance attributes. They can be defined both for a local and a global scope by specifying their domain.
  - Facets, or attribute properties, such as type constraint or cardinality, as well as documentation. However, it does not include a default value or other management facets that could be added as an extension.
- Taxonomies: subclasses can be defined, with multiple inheritance, as well as exhaustive and disjoint decompositions. At the same time, a class can be defined as complement of another.
- Axioms: OWL allows the definition of first order logic axioms with respect to algebraic property relationships (symmetry, transitivity and uniqueness). Universality and existentiality can also be defined for classes and properties.
- Instances of concepts, relationships (or facts) and claims can be also defined.

Then, this ontology language can be directly used to specify management information, because it has most of the constructions included in management information languages:

- Classes can be defined using the tag `owl:Class`, and its inheritance with `rdfs:subClassOf`.
- Then, attributes are specified using `owl:DatatypeProperty` for literal values or `owl:ObjectProperty` as references to class instances. Containment can also be expressed with this construction. One peculiarity of ontology languages is that properties do not directly belong to a class. Then, to attach a property to a class, the tag `rdfs:domain` has to be included. Other peculiarities about attributes are explained in next section.
- Finally, other constructions can also be defined: associations are classes with two or more `owl:ObjectProperty`, and events and notifications would be a specialization of `owl:Class`.

The next lines show an example about how to translate to OWL the class `CIM_ManagedSystemElement`, defined in the CIM Core Model as depicted in Figure 3. This translation would be similar in other object oriented languages such as GDMO. A prior

transformation has to be done in table-based languages such as SMI to consider those tables as classes.

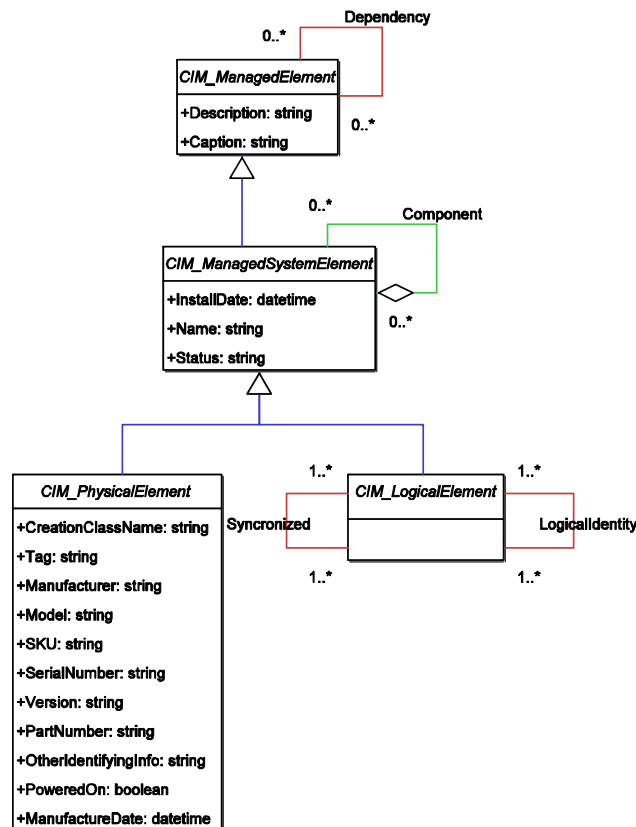


Figure 3. Subset of the CIM Core Model.

First of all, the XML document with RDF syntax has to be defined, including a set of namespaces for different tags:

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<rdf:RDF
  xmlns="http://www.dmtf.org/standards/cim/cim_schema_v28#"
  xmlns:cim="http://www.dmtf.org/standards/cim/cim_schema_v28#"
  xml:base="http://www.dmtf.org/standards/cim/cim_schema_v28#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#" >
  
```

Then, a header with general ontology information is included:

```

<owl:Ontology rdf:ID="CIM_Core">
  <rdfs:comment>The version 2.8 of CIM Core model</rdfs:comment>
  <rdfs:label>CIM Core Model</rdfs:label>
  <owl:versionInfo>2.8</owl:versionInfo>
  <owl:priorVersion rdf:resource=
    "http://www.dmtf.org/standards/cim/cim_schema_v27">
</owl:Ontology>
  
```

Next, the class is defined as a subclass of CIM\_ManagedElement, also specified in this document:

```

<owl:Class rdf:ID="CIM_ManagedSystemElement">
  <rdfs:comment>
  
```

```
CIM_ManagedSystemElement is the base class for the System Element
hierarchy. [...]
  </rdfs:comment>
  <rdfs:subClassOf rdf:resource="#CIM_ManagedElement"/>
</owl:Class>
```

Class properties are also defined, with a mapping of their data types to XML Schema Datatypes (XSD).

```
<owl:DatatypeProperty rdf:ID="InstallDate">
  <rdfs:comment>
A datetime value indicating when the object was installed. A lack of a
value does not indicate that the object is not installed.
  </rdfs:comment>
  <rdfs:domain rdf:resource="#CIM_ManagedSystemElement"/>
  <rdfs:range rdf:resource=
    "http://www.w3.org/2001/XMLSchema#dateTime"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="Name">
  <rdfs:comment>
The Name property defines the label by which the object is known. When
subclassed, the Name property can be overridden to be a Key property.
  </rdfs:comment>
  <rdfs:domain rdf:resource="#CIM_ManagedSystemElement"/>
  <rdfs:range rdf:resource=
    "http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="Status">
  <rdfs:comment>
A string indicating the current status of the object. Various operational
and non-operational statuses are defined. [...]
  </rdfs:comment>
  <rdfs:domain rdf:resource="#CIM_ManagedSystemElement"/>
  <rdfs:range rdf:resource=
    "http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
```

Finally, the RDF document tag is closed.

```
</rdf:RDF>
```

This translation can be done with all classes defined in the CIM core model. However, not all their qualifiers can be directly expressed in OWL. For instance, attributes can be specified with a maximum length or an access level, which are not defined above. Next section explains how to express in OWL these common management facets.

## 4 A proposal for expressing management facets in OWL

Attributes of any information model usually have a set of properties called facets, such as the data type or the description. Most of the common management facets are included in OWL. Other ones can be expressed as an extension to this language, specifying them in RDFS to include those cases where they do not exist in OWL. In this way, every system can understand most of the ontology if it understands OWL. At the same time, the defined information is as complete as possible.

The next list shows relevant common facets in management information that can be represented using OWL:

- Data type restriction. All management information languages use data types to define attribute syntax. Usually, value ranges can also be included. In OWL, `rdfs:range` specifies the set of values a property can have. `owl:allValuesFrom` can also be used to define a concrete class whose property has this range of values. If an attribute is defined with `owl:DatatypeProperty`, data types are taken from those defined in the XML Schema, as seen previously. This kind of data types can be restricted to a specific range of values. For instance, a string with maximum length of 256 can be defined as:

```
<xsd:simpleType name="string256">
  <xsd:restriction base="xsd:string">
    <xsd:maxLength value="256" />
  </xsd:restriction>
</xsd:simpleType>
```

- Documentation. Provides a description of the attribute in natural language. Most management languages have such a clause. In OWL, the constructions `rdfs:comment`, defined as `owl:AnnotationProperty`, can be used to describe the property, as seen in the example above.
- Unique identifier. Gives an identifier, usually numeric in management languages, unique for each attribute, and linked to its name. In OWL, it can be given by the tag `rdf:ID`, that provides the reference to a resource in a document, which has a Uniform Resource Identifier or URI.
- Implementation status. Informs about the current state of the attribute in the specifications. Also, the version definition. In OWL, many tags have been defined with this purpose: `owl:versionInfo`, `owl:priorVersion`, `owl:backwardCompatibleWith`, and `owl:incompatibleWith`, to be included in the header of the ontology, as seen in the example with first two tags, and `owl:DeprecatedClass`, `owl:DeprecatedProperty` for classes and properties inside the document.
- Distinction. This kind of attribute distinguishes every instances of a class. Although `rdf:ID` tag can be used to identify instances, OWL also provides the `owl:FunctionalProperty` construction, that restricts the value of a property to be unique for each instance.
- Cardinality. This facet restricts the number of values for every attribute. OWL defines `owl:cardinality`, `owl:maxCardinality`, `owl:minCardinality` to specify respectively the exact, maximum and minimum cardinality of a property.
- Reference. Gives a cross reference to other related attribute. With this purpose, `rdfs:seeAlso` can be used to specify a resource with additional information.
- Redefinition. Informs if the attribute is redefined with respect to one contained in a parent class. A property can specialize another one using the tag `rdfs:subPropertyOf`.

Table 1 summarizes these facets, and compares them with the constructions used in three main management information languages (SMIv2, GDMO and CIM). This table shows that three facets do not have a correspondence in OWL. A way of expressing these facets is explained below.

**Table 1. Common facets in management information languages**

Language	SMIv2	GDMO	MOF/CIM	OWL
Data type restriction	SYNTAX	WITH ATTRIBUTE SYNTAX	Attached	rdfs:range, owl:allValuesFrom
Documentation	DESCRIPTION	BEHAVIOUR	Description	rdfs:comment
Unique identifier	::=	REGISTERED AS	MappingString	rdf:ID
Implementation status	STATUS	n/a	Version, Revision	owl:versionInfo, owl:priorVersion, owl:backwardCompatibleWith, owl:incompatibleWith, owl:DeprecatedClass, owl:DeprecatedProperty
Distinction	INDEX, AUGMENTS	WITH ATTRIBUTE	Key, Propagated, Weak	owl:FunctionalProperty
Cardinality	n/a	n/a	[n], Max, Min	owl:maxCardinality, owl:minCardinality, owl:cardinality
Reference	REFERENCE	n/a	Model Correspondence	rdfs:seeAlso
Redefinition	n/a	DERIVED FROM	Override	rdfs:subPropertyOf
Access	MAX-ACCESS	Attached	Read, Write	n/a
Units	UNITS	n/a	Units	n/a
Default value	DEFVAL	DEFAULT VALUE	=	n/a

- Access. Shows the way an attribute can be accessed: read, write, create, etc. To define the access, a set of access classes can be defined in OWL.

```
<owl:Class rdf:ID="accessClass">
  <owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:about="#read-only"/>
    <owl:Thing rdf:about="#read-write"/>
    <owl:Thing rdf:about="#read-create"/>
  </owl:oneOf>
</owl:Class>
```

Then, an “access” RDF property is defined, whose domain can be both an OWL object property and an OWL data type property:

```
<rdf:Property rdf:ID="access">
  <rdfs:label>access</rdfs:label>
  <rdfs:comment>
It defines the access of a property, which can be read or write.
  </rdfs:comment>
  <rdfs:domain rdf:resource=
    http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:domain rdf:resource=
    "http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:range rdf:resource="#accessClass"/>
</rdf:Property>
```



- Units. Provides the measurement unit used in the attribute. A “units” property can be defined in RDFS for this. In this case, the domain can only be related to data type properties. The range can be any ontology about measurement units, such as the GNU Units Ontology developed in the DAML (DARPA Agent Markup Language) Program. Then, values of these facets would be instances of this ontology.

```

<rdf:Property rdf:ID="units">
  <rdfs:label>units</rdfs:label>
  <rdfs:comment>
It defines the measure units of a property.
  </rdfs:comment>
  <rdfs:domain rdf:resource=
    "http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:range rdf:resource=
    "http://www.daml.org/2002/10/units/units-ont.owl#Unit"/>
</rdf:Property>

```

- Default value. All management languages allow the definition of a default value for their attributes. To define this facet in RDFS, the domain will again be OWL data type properties, and range will not be restricted, so any data type can be expressed.

```

<rdf:Property rdf:ID="defaultValue">
  <rdfs:label>defaultValue</rdfs:label>
  <rdfs:comment>
It defines the default value of a property.
  </rdfs:comment>
  <rdfs:domain rdf:resource=
    "http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</rdf:Property>

```

Other qualifiers can be defined if necessary in a similar way.

## 5 Application: ontology-based management

The analysis shown in prior sections has been performed in the scope of a research project whose aim is the integration and improvement of management information from a semantic viewpoint. Several tasks have been identified to achieve this goal, integrating OWL in the architecture depicted in Figure 1.

The first task is related to the translation of management specifications into ontology languages, to improve their semantic expressiveness. For this point, a Protégé 2000 [8] plug-in has been developed that loads Internet MIBs and CIM schemas (represented in MOF) as ontologies. To load Internet MIBs, tables and groups are processed as classes, and some inheritance associations are also obtained from the algorithm defined in [9] (see Figure 4 and Figure 5). The load of GDMO definitions is also an ongoing work. Then, these management specifications can be exported to different ontology languages supported in Protégé, including OWL among other XML-based ontology representations. The representation of this information in OWL will be done according to the methodology presented before.

A second task deals with the semantic integration of management specifications. A method has been defined to merge and map different MIBs [5], which is being developed as another plug-in for Protégé. OWL also provides some constructions to express simple mapping relationships, such as `owl:equivalentClass`, `owl:equivalentProperty` or `owl:sameAs`. However, a mapping ontology [2] is necessary to describe more complex mappings. This ontology can be represented in OWL.

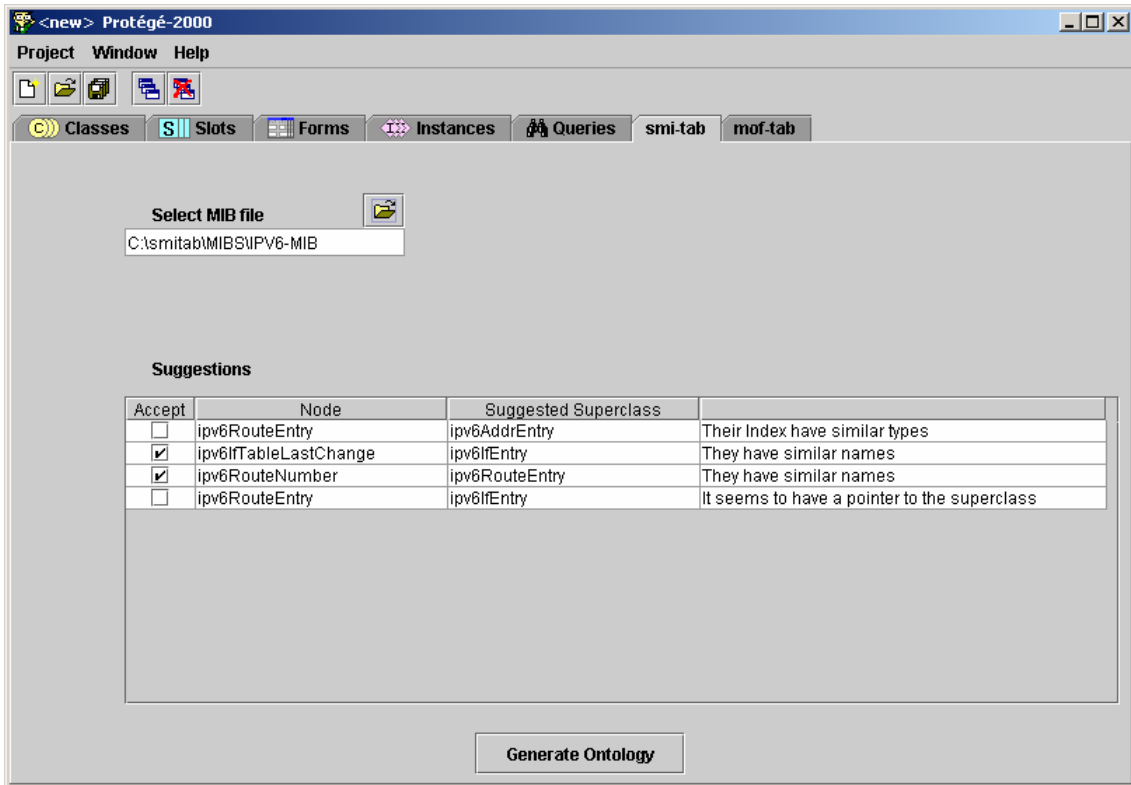


Figure 4. Protégé 2000 SMI-Tab plugin

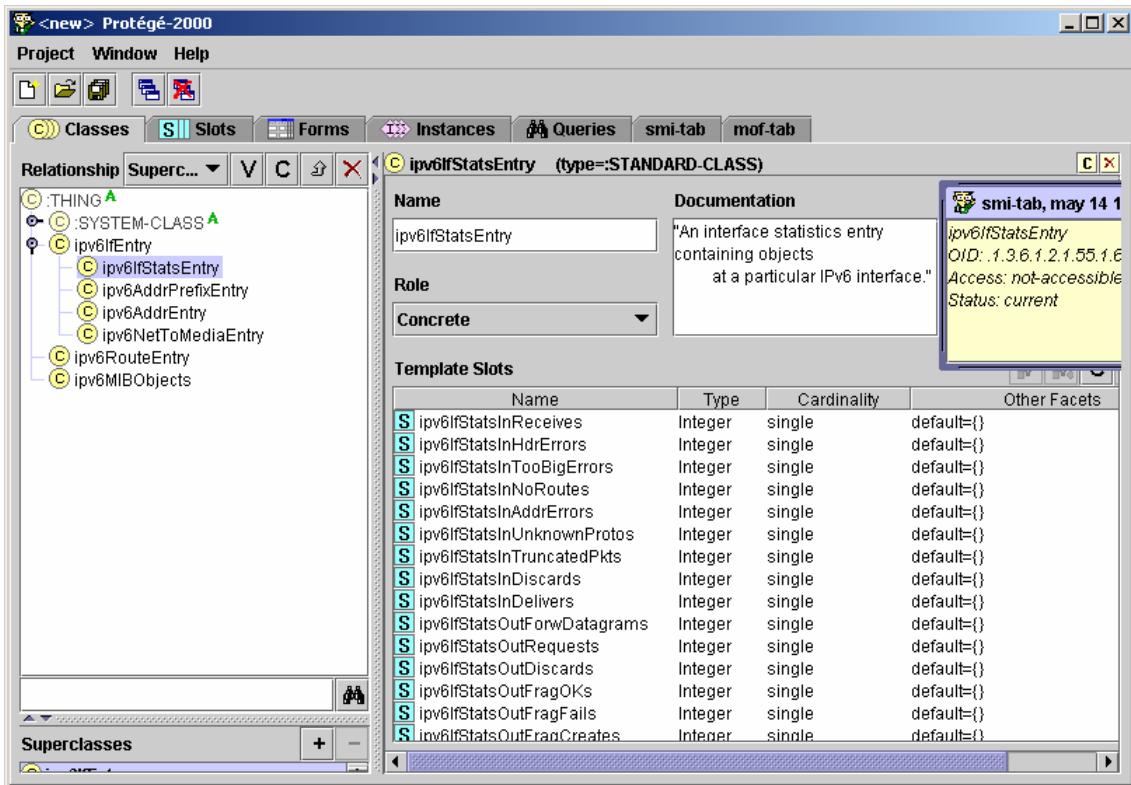


Figure 5. Generated Ontology from SMI-Tab plugin

A third task is related to the definition of behavior inside the ontology for its use by a manager. Current management information definitions include some rules about the behavior, but they are written in natural language in BEHAVIOUR or DESCRIPTION fields, and are not

machine-readable. Ontologies provide axioms and constraints, which can be interpreted and automatically checked by the management system because they are formally defined.

Two types of constraints can be included in the management information. The first type has been named implicit constraint, and it is usually described in natural language in the class or attribute description of the management information. Examples are those descriptions in current management information definitions that include words such as “*should*”, “*have to*” or “*must*”. They must be true in a correct operational state. For instance, the class `CIM_IPSubnet` of the CIM Network schema includes in the description of its property `SubnetNumber` that “*IPv4-compatible addresses should be used instead of IPv6*”, which limits the format of these addresses.

The second type of constraint has been named explicit constraint, and it is defined by a manager about already defined information to specify the behavior of the managed resources. These constraints follow a concrete management policy. For instance, a manager can define a constraint to guarantee that the number of connections to a certain TCP port of a concrete machine is less than a threshold.

There are some policy languages specifically defined for this purpose, like Ponder [10], but they are defined in a language different from the management information model. The ontology approach combines both types of information in the same model, which allow just one language interpreter and an integrated definition of the elements and their behavior in the same place, making easier its maintenance.

In this scope, OWL currently has a limited way of defining restrictions, using the tag `owl:Restriction`. To complement this issue, the Joint US/EU ad hoc Agent Markup Language Committee is currently defining the Semantic Web Rule Language (SWRL) [11], combining OWL with RuleML to extend the set of OWL axioms including Horn-like rules.

The final scenario will be as depicted in Figure 1, where the manager can use existing OWL interpreters to deal with the management ontology. This ontology will then be mapped to specific management information concepts with the use of a mapping ontology. Finally, the manager will be able to reason with the set of implicit and explicit constraints expressed in SWRL by using generic tools, like inference engines, which process this rule language. The overload performance due to the use of this architecture is only assumed by the manager, who has to parse and interpret XML-based ontology definitions and map them to traditional management models.

## 6 Conclusion

This paper has presented an approach that uses an XML-based ontology language to define network and system management information. In addition to the advantages related to XML syntax, ontology languages enhance the semantic expressiveness of the definitions, and there are many tools to validate and reason with them.

OWL is a general purpose ontology language based on RDFS that contains all constructions necessary to formally describe most of the management information: classes and properties, with hierarchies, and domain and range restrictions. In fact, most management facets can be mapped with different tags defined in OWL. Nevertheless, as it happens with many language translations, some information can be lost because of the lack of other constructions. This problem can be solved by defining new facets using RDFS as proposed.

In addition to semantic improvements, ontology languages can also be useful for the integration of different specifications from diverse management domains, and to formally describe the behavior of a manager.

Current works address these points, adapting ontology tools to load management specifications written in SMI, CIM or GDMO, and to automate the process of merging and mapping different MIBs, including also SWRL behavior constraints in the information.

## 7 References

1. F. Strauß and T. Klie, "Towards XML Oriented Internet Management," in G. Goldszmidt and J. Schönwälder (Eds.), *Integrated Network Management VIII, Proc. 8th IFIP/IEEE International Symposium on Integrated Network Management*, Colorado Springs, USA, March 2003, Kluwer Academic Publishers, pp. 505-518.
2. J. E. López de Vergara, V. A. Villagrà, J. I. Asensio, and J. Berrocal, "Ontologies: Giving Semantics to Network Management Models," *IEEE Network*, Vol. 17, No. 3, May/June 2003, pp. 15-21.
3. J. Shen and Y. Yang, "RDF-Based Knowledge Models for Network Management," in G. Goldszmidt and J. Schönwälder (Eds.), *Integrated Network Management VIII, Proc. 8th IFIP/IEEE International Symposium on Integrated Network Management*, Colorado Springs, USA, March 2003, Kluwer Academic Publishers, pp. 123-126.
4. E. Lavinal, T. Desprats, and Y. Raynaud, "A Conceptual Framework for Building CIM-Based Ontologies," in G. Goldszmidt and J. Schönwälder (Eds.), *Integrated Network Management VIII, Proc. 8th IFIP/IEEE International Symposium on Integrated Network Management*, Colorado Springs, USA, March 2003, Kluwer Academic Publishers, pp. 135-138.
5. J. E. López de Vergara, V. A. Villagrà, and J. Berrocal, "An ontology-based method to merge and map management information models", in *Proc. HP Openview University Association 10th Plenary Workshop*, Geneva, Switzerland, July 2003.
6. M. K. Smith, C. Welty, and D. L. McGuinness, "OWL Web Ontology Language Guide," W3C Candidate Recommendation, 10 February 2004.
7. A. Gómez-Pérez and O. Corcho, "Ontology Languages for the Semantic Web," *IEEE Intelligent Systems*, Vol. 17, Issue 1, January/February 2002, pp. 54-60.
8. N. F. Noy, M. Sintek, S. Decker, M. Crubézy, R. W. Ferguson, and M. A. Musen, "Creating Semantic Web Contents with Protégé-2000," *IEEE Intelligent Systems*, Vol. 16, Issue 2, March/April 2001, pp. 60-71.
9. J. Schönwälder and A. Müller, "Reverse Engineering Internet MIBs," in G. Pavlou, N. Anerousis, and A. Liotta (Eds.), *Integrated Network Management VII, Proc. 7th IFIP/IEEE International Symposium on Integrated Network Management*, Seattle, Washington, USA, May 2001, IEEE Press, pp. 423-436.
10. N. Damianou, N. Dulay, E. Lupu, and M. Sloman, "The Ponder Policy Specification Language," in M. Sloman, J. Lobo, E. Lupu (Eds.), *Proc. Policy 2001: Workshop on Policies for Distributed Systems and Networks*, Bristol, UK, 29-31 Jan. 2001, Springer-Verlag LNCS 1995, pp. 18-39.
11. I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Groszof, and M. Dean, "SWRL: A Semantic Web Rule Language Combining OWL and RuleML", *Joint US/EU ad hoc Agent Markup Language Committee Document*, Version 0.5, 19 November 2003.