Benefits of Using Ontologies in the Management of High Speed Networks

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

Departamento de Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid,
E.T.S.I. de Telecomunicación, Av. Complutense, s/n, E-28040 Madrid, Spain
Tel: +34 91 549 57 00, Fax: +34 91 336 73 33
{jlopez, villagra, berrocal}@dit.upm.es
* Currently working at Universidad Autónoma de Madrid, Spain

Abstract. Network management is an area where many different technologies coexist. Several languages are used to define the information to be managed, which are specific of each management model. As a result, many specifications that describe similar resources are expressed separately. To solve this question, this paper takes advantage of the knowledge representation technique known as ontology to unify current heterogeneous information definitions from a semantic viewpoint. With this approach, management information is specified using ontology languages, including behavior constraints, and methods used to combine ontologies are applied to merge and map the concepts contained in existing management models.

Keywords: Ontology, Management Specification, Information Integration, Behavior Description.

1 Introduction

Many different frameworks coexist in the management of high speed networks. They include SNMP (Simple Network Management Protocol), OSI-SM (Open Systems Interconnection - Systems Management), and WBEM (Web Based Enterprise Management). Even CORBA (Common Object Request Broker Architecture) can be applied to manage these networks and related resources [1]. For their operation, these standards also define some information, usually grouped in Management Information Bases (MIBs) to describe the managed resources, so that a common knowledge is shared between managers and agents. However, the management information can be defined in several languages which are specific of each management model.

This model diversity is a problem when two or more of these technologies have to be used to access different network resources belonging to different management domains. In these cases interoperability mechanisms must be applied. However, existing approaches such as IIMC (ISO-Internet Management Coexistence), JIDM (Joint Inter-

This work has been partially funded by the Spanish Ministry of Science and Technology under the project GESEMAN (TIC2002-00934).

Domain Management), or even CIM (Common Information Model) only provide syntactic translations that recast information models [2]. With these solutions, if one resource is described in two or more information models, the structures composing the specification can directly be mapped, but not the meaning they contain, which is a problem to achieve an integrated enterprise management. To solve it, the semantics contained in the information should be taken.

At the same time, ontologies have been successfully used to solve similar problems in other domains such as the Semantic Web, in which this knowledge representation technique provides semantics to web pages and web services. This paper studies how ontologies can also be useful for network and system management, unifying from a semantic viewpoint current heterogeneous information definitions. For this, first of all ontologies will be presented and compared to current management information models. Then, three steps to obtain models with enhanced semantics and integrated in a management ontology will be detailed. Finally, conclusions obtained in this study will be provided.

2 Applying Ontologies to Network and System Management

Ontologies are one of the main approaches used in the scope of Knowledge Management and Artificial Intelligence to solve questions related to semantics. To understand how they can be applied to management information models, an explanation about them is given. An ontology can be defined as "an explicit and formal specification of a shared conceptualization" [3]:

- It is explicit because it defines the concepts, properties, relationships, functions, axioms and restrictions that compose it.
- It is formal because it can be read and interpreted by machines.
- It is a conceptualization because it is an abstract model and a simplified view of the entities it represents.
- Finally, it is shared because it has been agreed by a group of experts.

Briefly, an ontology is the definition of a set of concepts, their taxonomy, interrelation and the rules that govern these concepts.

Ontologies can be classified into two main groups: lightweight and heavyweight. First group include those modeling the information related to a domain but without axioms or restrictions, so that it is difficult to reason with them. Second group do include all elements that allow the inference of knowledge from the defined information.

Then, management information models could be understood as lightweight ontologies. Internet MIBs or CIM schemas define the information of the management domain in a partially formal way, and have been agreed in working groups. Nevertheless, their semantics is limited because they do not include restrictions [2].

Another point that differentiates ontologies from management information models is the way in which the interoperability problem has been addressed. Solutions applied to integrate different ontologies are not only about syntactic translations of different languages, but also dealing with the semantics of the information. Proposals from different research groups working in this area include merging models to obtain a common ontology and mapping among all models.

Three steps have been studied to improve the semantic interoperability of different management specifications and integrate these information models from an ontology viewpoint. These steps, depicted in Fig. 1 and detailed in following sections, are the following ones:

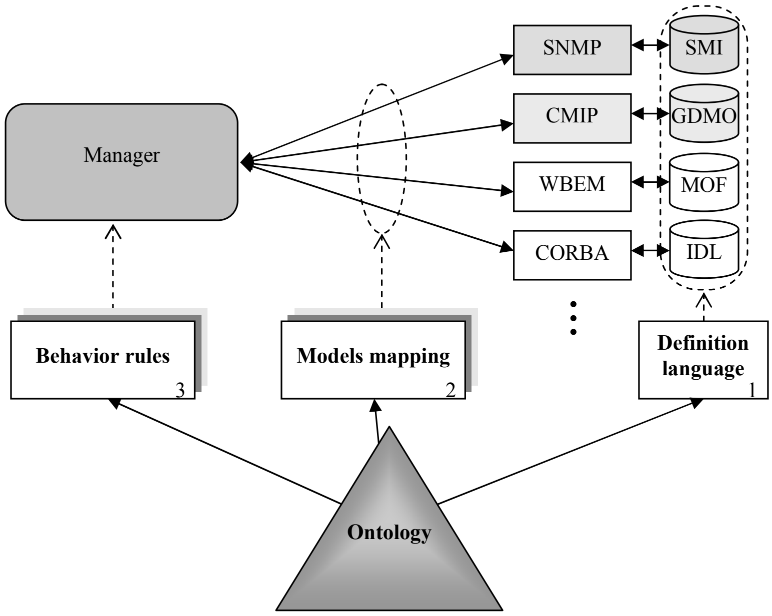

Fig. 1. Ontology applications to network and system management

1. First of all, the possibilities of specifying management information using an ontology definition language are studied, adapting such language to express common management constructions.
2. These management information specifications expressed with ontology languages can be merged and mapped using the techniques used for ontologies. For this, a method is defined that combines both tasks and is applied specifically to management information. Then, a new model can be obtained that integrates existing ones, bearing in mind the semantics contained on them, and declaring at the same time the mapping rules with initial specifications. In this way, the interoperability problem that was previously identified can be solved.
3. Finally, a way of adding behavior to management information specifications is analyzed. A set of constraints can be added to complete the common model generated in preceding step by taking advantage of the characteristics included in ontology languages.

3 Describing the Management Information with Ontology Languages

A definition language with enough semantic expressiveness would be necessary to describe the management information taking into account its meaning. This section studies the possibility of using an ontology definition language to specify the management information. This language should be adapted so that it can also express some common constructions in management information languages.

There are also other works in the same line. In [4] a translation between SMIng (Structure of Management Information, next generation) and RDF (Resource Description Framework) is proposed. The ontology-based generated vocabulary would be used for the communication among intelligent agents. Another approach [5] integrates CIM meta-model in an ontology-tool. Then, CIM schemas would be used to exchange information among intelligent agents using OKBC (Open Knowledge Base Connectivity). Nevertheless, these approaches do not use any concrete ontology language to define the management information.

Those ontology languages related to the Semantic Web are the most used and with a high number of available tools. DAML+OIL (DARPA Agent Markup Language + Ontology Inference Layer) [6] is one of the most remarkable. Based on DAML+OIL, the W3C is currently defining OWL (not an acronym) [7], a web ontology language that is in the final stage of its definition process and quite similar to DAML+OIL. Another advantage with respect to management information languages is that ontology languages have been formalized, so that their semantics is sound and complete and they can be used by intelligent systems.

Although the study shown in this section is referred to DAML+OIL, it can be generalized to other ontology definition languages with similar characteristics, such as OWL. All those languages that allow the definition of classes and properties can be valid to define management information, but it is possible that some information get lost if these languages do not have the suitable facets or a mechanism to define them.

DAML+OIL is a very complete ontology language, because it allows the definition of classes and properties, which can belong to a class domain. At the same time, properties can have different facets, such as the type or cardinality constraint, as well as the documentation. Class specialization can also be defined, with simple and multiple inheritances, as well as other relationships with range constraints. Finally, instances of these classes can also be defined. To illustrate it, next lines show the class `CIM_ManagedSystemElement` taken from CIM core model described in DAML+OIL. Data types are those used in XSD (XML Schema Data types).

```
<daml: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"/>
</daml:Class>
<daml: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>
   <daml:domain
      rdf:resource="#CIM_ManagedSystemElement"/>
   <daml:range rdf:resource=
      "http://www.w3.org/2000/10/XMLSchema#dateTime"/>
</daml:DatatypeProperty>
<daml: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>
   <daml:domain
      rdf:resource="#CIM_ManagedSystemElement"/>
   <daml:range rdf:resource=
      "http://www.w3.org/2000/10/XMLSchema#string"/>
</daml:DatatypeProperty>
<daml: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>
   <daml:domain rdf:resource="#CIM_ManagedSystemElement"/>
   <daml:range rdf:resource=
      "http://www.w3.org/2000/10/XMLSchema#string"/>
</daml:DatatypeProperty>
```

However, ontology languages do not have constructions to define all the facets common in management information languages. For instance, DAML+OIL does not include facets such as the default value or the access level, which are common in management languages. To solve this, and taking into account that DAML+OIL has been defined on top of RDF, these management facets can also be defined in RDF, extending DAML+OIL vocabulary. Next lines show the definition of such facets.

```
<!-- Default value -->
<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/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Property>
<!-- Access level-->
<rdf:Property rdf:ID="access">
   <rdfs:label>access</rdfs:label>
   <rdfs:comment>
It defines the access of a property, which can be readable
and/or writeable.
   </rdfs:comment>
   <rdfs:domain rdf:resource=
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
   <rdfs:range rdf:resource="#accessString"/>
</rdf:Property>
```

```
<xsd:simpleType name="accessString">
   <xsd:restriction base="string">
      <xsd:enumeration value="read-only"/>
      <xsd:enumeration value="read-write"/>
      <xsd:enumeration value="read-create"/>
   </xsd:restriction>
</xsd:simpleType>
```

Even if new facets are defined in RDF, management methods or operations cannot be expressed in this language. If necessary, other ontology languages with operation support should be used instead. However, not all management information specifications include them frequently. Then, from basic DAML+OIL structures and RDF-based management facets, current management information can be translated into an ontology language. To carry out this task, languages constructions should be compared as done in [2].

4 M&M method: Merging and Mapping the Management Information

Expressing all management information models in the same language is not enough to integrate them semantically, even if it is an ontology language. It is necessary to carry out other procedures that identify the meaning contained in these specifications. This can be done by leveraging the merging and mapping techniques used in the ontology field. This section shows a method that combines different definitions in a common model extracting their semantics, declaring different mapping rules to the initial specifications, as illustrated in Fig. 2.

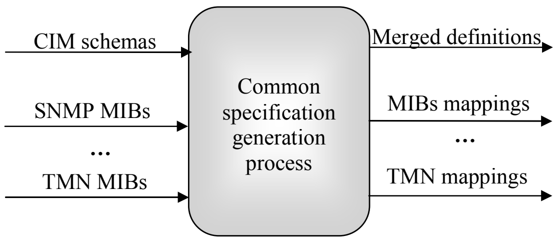

Fig. 2. Merging and mapping management information process

To assist in this process this work has defined a method called M&M (Merge and Map). It proposes a set of steps to help in the procurement of both the common model and the mapping rules. It is based in the merging method detailed in [8], adapting it to the particular case of the network and system management, and adding the mapping definition, so that for every merged element, a new rule is set in the mapping ontology to express the relationship between these elements.

This method does not generate a result automatically. Its aim is to help the person performing the merging and mapping task. For this, some heuristics have been proposed to identify possible candidates to merge with high probability. These heuristics can be:

- Candidates by similar character strings. Two classes or properties are candidates to be merged with high probability if they have a similar substring included in their identifiers or in their descriptions. Synonyms can also be used if they are available.
- Candidates by similar inheritance hierarchy. Two classes are candidates to be merged with high probability if their parent classes are similar, because child classes of a class are usually similar to the child classes of another class that has been merged with the first one.
- Candidates by property domain. Two properties are candidates to be merged with high probability if the classes containing them are also similar. At the same time, two classes are candidates to be merged with high probability if the properties they contain are also similar.

To make the merging process easy, an existing specification, CIM, is used as a base model which will be merged with other information definitions such as MIB-II, HOST-RESOURCES-MIB or the M.3100 recommendation [2].

Usual mappings among management information models have been identified. They can be the following ones or a combination of them:

- Direct, if it is a 1:1 relationship in which no transformation is needed. In this case, the value contained in an element in both models is exactly the same. Direct mappings are the most common ones and are proposed by default.
- Value set, if it is a 1:1 relationship in which for each value of an element there is another value for the element of the other model. This kind of mapping is proposed if at least one of two merged properties has an enumeration of possible values.
- Data types, if it is a 1:1 relationship in which the elements of both models have different data types that have to be converted.
- Arithmetic operation on one element, if it is a 1:1 relationship in which the value of one element is obtained calculating it with the value of the element of the other model. This kind of mapping is useful if the measurement units of both elements are different.
- Arithmetic operations on some elements, if it is a 1:n relationship in which the value of one element is obtained by arithmetic combination of the values of some elements of the other domain. In this case, the user should define the mapping expression.
- Character strings, if it is a 1:n relationship in which the value of one element is composed by concatenating different strings, which are the values of the elements in the other domain. Once again, the user has to define the mapping rule.

To describe the mapping rules that translate the information instances from a concrete model to the common model a simple mapping ontology has been defined. Fig. 3 shows with a representation of this ontology a class diagram.

This ontology has two concepts: each `Element` composing an ontology (classes, properties, etc.) has a translation `Formula`. Every `Element` has some properties such as the `type` or the `reference` to its specification. Every `Formula` has an `expression` written in a concrete `language` to translate the set of `source` and `target` elements. Relationships among mapped elements and inverse formula are also included. With this ontology any mapping of those identified before can be addressed. Other approaches, such as the MappingStrings qualifier used in CIM, can only represent direct mappings.

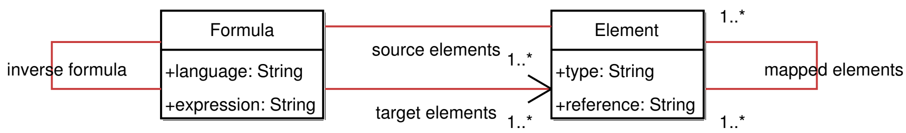

Fig. 3. Mapping ontology

Fig. 4 shows the activity diagram that describes the M&M method. Gray activities are those performed by a user and those in white are automatically completed by the system. In brief, this method consists on the identification of similar classes and properties, using commented heuristics, to merge them. At the same time, every element of the mapping ontology is automatically defined at the beginning, and later, associated formulas are created when merging the elements. The person in charge of this process must validate every operation proposed by the method, and can also define different ones. The final result is a common model and a set of instances of the mapping ontology representing the rules to translate the merged models, as stated before. An application of this method is presented in [9].

A management system based on this proposal would work in the following way. If it needs to obtain all instances of a concrete element of the common model, it will look for it in the mapping ontology, finding at the same time the formula and mapped elements of the merged models. It will access each management domain containing this information to get it. Applying the expression contained in the formula it will translate the value of the instances obtained in each domain to the common model.

5 Adding Behavior Definitions to the Management Information

A step further in the information integration is to add a set of constraints to the obtained common management model. This will allow the description of the behavior related to the information contained in this model, which can be checked by the 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 can be applied again. They provide axioms and constraints to specify their behavior, which can be interpreted automatically because it is defined formally.

Two types of constraints can be included in the information. The first type has been named implicit constraint, and it is referred to the information that must be true in a correct operation state, because it has been defined in this way. The second type has been named explicit constraint, and it is defined by a concrete manager about already defined information to specify the behavior of the managed resources.

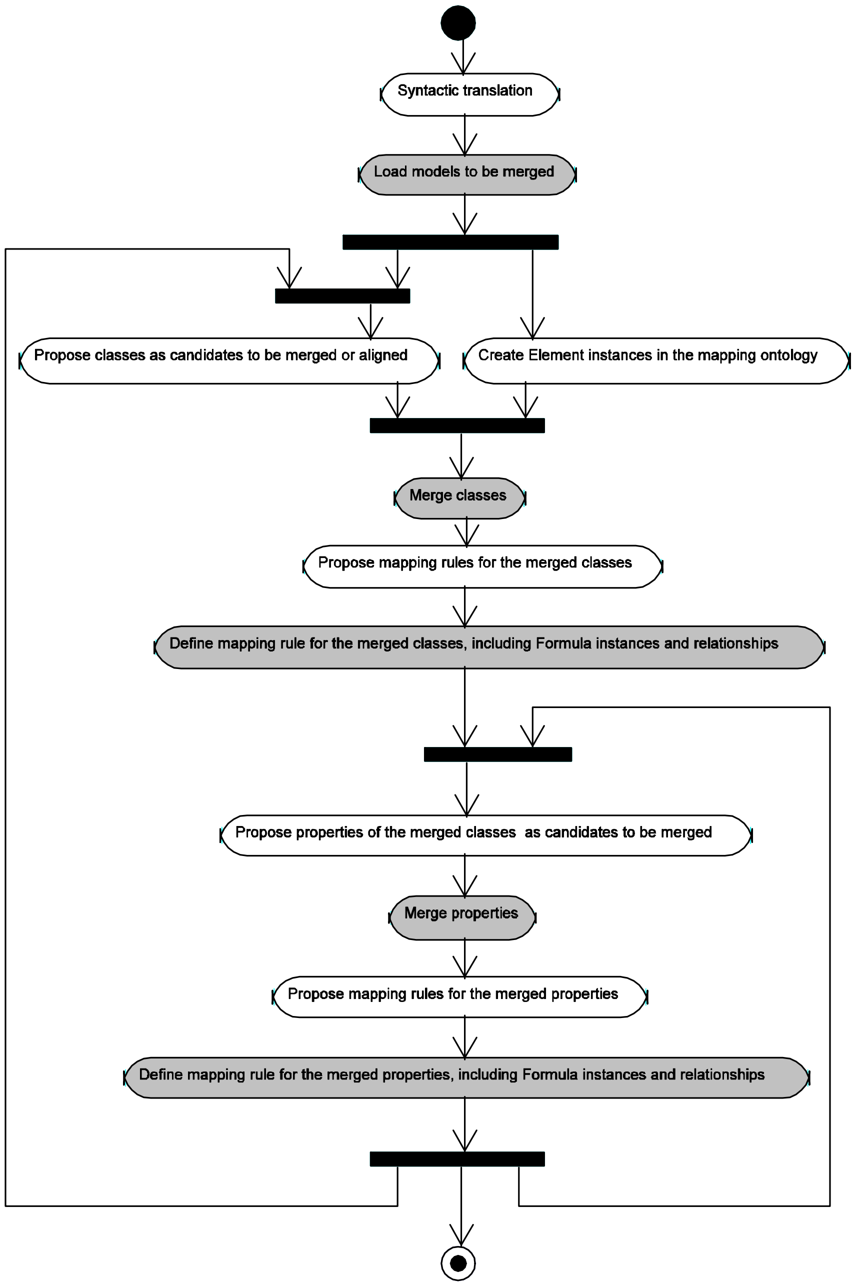

Fig. 4. M&M method activity diagram

Implicit constraints are usually defined in natural language in the class or attribute descriptions of the management information. An example of them could be different constraints contained in the `CIM_Printer` class of the CIM schema, such as "`a language that is used as a default by the Printer should also be listed in LanguagesSupported`", that limits the default language that a printer can use to those contained in a list.

Explicit constraints follow a concrete policy. For instance, a policy can be defined to guarantee that the free space in the file systems of a computer will never be less than a 10% of the total capacity.

DAML+OIL allows the definition of constraints with first order logic with respect to relationship algebraic properties (symmetry, transitivity, uniqueness). Moreover, universality and existence constraints can be defined for classes and properties. However, DAML+OIL constraints are related to properties about objects, being less expressive with properties about data types. Because of this, other ontology constrain languages, such as PAL (Protégé Axiom Language) [10], are more suitable to define this kind of information.

The example proposed for the implicit constraint could be defined in DAML+OIL in the following way:

```
<daml:Class rdf:about"#CIM_Printer">
   <daml:Restriction>
      <daml:onProperty rdf:resource="#LanguagesSupported"/>
      <daml:hasClass rdf:resource="#DefaultLanguage"/>
   </daml:Restriction>
</daml:Class>
```

At the same time, it could be expressed in KIF like this:

```
(defrange ?printer :FRAME CIM_Printer)
(forall ?printer
   (element-of
      (DefaultLanguage ?printer)
      (LanguagesSupported ?printer)))
```

Using again PAL for the second example, it would result in:

```
(defrange ?fs :FRAME CIM_FileSystem)
(forall ?fs
   (> (AvailableSpace ?fs)
      * 0.10 (FileSystemSize ?fs)))
```

However, DAML+OIL cannot express exactly this constraint. A value constraint can be defined using the XSD constructions, but a maximum value with an operation cannot be defined. For example, if the available space is measured in a percentage instead of absolute values, the constraint could be defined in the following way:

```
<xsd:simpleType name="over10">
   <xsd:restriction base="xsd:positiveInteger">
      <xsd:minInclusive value="10"/>
<!-- As it is a percentage, it can include a maximum
constraint -->
      <xsd:maxInclusive value="100"/>
   </xsd:restriction>
</xsd:simpleType>
<daml:Class rdf:about="#CIM_FileSystem">
   <daml:Restriction>
      <daml:onProperty rdf:resource="#AvailableSpace"/>
      <daml:toClass rdf:resource="#over10"/>
```

```
    </daml:Restriction>
</daml:Class>
```

6 Conclusions

This paper has presented an approach to apply the formal representation technique known as ontology to improve the definition and integration of network and system management information. If the management information is written in an ontology language such as DAML+OIL or OWL, its expressiveness is enhanced, and there are many tools to read and validate this information. Nevertheless, it is also necessary to extend this kind of languages to express all the information usually contained in the management models.

Existing works with respect to the management information integration were based until now in syntactic translations. This task can be done in a semantic way if the M&M method is applied. In this way, a manager can handle a single model, where the underlying management domains are transparent. This will get better management applications, which will be able of connecting data that did not have a direct association so far because they belonged to different management domains. The application of this method to big information models has taken very long during the experimentation, due to the human intervention that is necessary to check the correctness of the proposed rules. However, this time will be less than what a completely manual process would take.

Based on these ideas, a management system like the one depicted in Fig. 5 can be developed, which uses this ontology-based approach, integrating all management models in a smart way, bearing in mind the semantics of the defined information. At the same time, generic gateways can be implemented that use the mapping ontologies obtained when applying the M&M method to translate the information to each management domain.

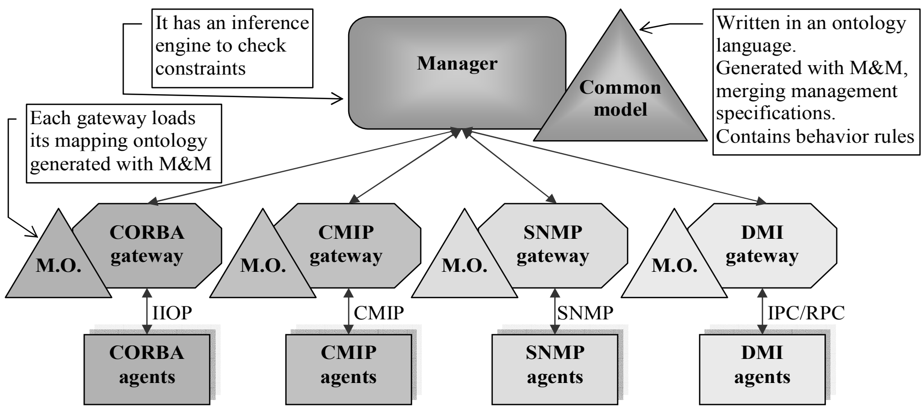

Fig. 5. Architecture of an ontology-based manager

Current works include automatic loading of MOF, SMI and GDMO files into an existing open source ontology tool [11] that generates DAML+OIL and OWL. At the same time, the M&M method is being adapted to that tool to automate the merging and mapping process.

References

1. Heinz-Gerd Hegering, Sebastian Abeck, Bernhard Neumair: Integrated Management of Networked Systems. Morgan Kaufmann (1999)
2. Jorge E. López de Vergara, Víctor A. Villagrá, Juan I. Asensio, Julio Berrocal: Ontologies: Giving Semantics to Network Management Models. IEEE Network, special issue on Network Management, Volume 17, Number 3 (May 2003)
3. R. Studer, V.R. Benjamins, D. Fensel: Knowledge Engineering: Principles and Methods. Data & Knowledge Engineering. 25. (1998) 161-197
4. Jun Shen, Yun Yang: RDF-Based Knowledge Models for Network Management. In: Proceedings of the Eighth IFIP/IEEE International Symposium on Integrated Network Management (IM'2003), Colorado Springs, Colorado, U.S.A. (March 2003)
5. Emmanuel Lavinal, Thierry Desprats, Yves Raynaud: A Conceptual Framework for Building CIM-Based Ontologies. In: Proceedings of the Eighth IFIP/IEEE International Symposium on Integrated Network Management (IM'2003), Colorado Springs, Colorado, U.S.A. (March 2003)
6. Dan Connolly, Frank van Harmelen, Ian Horrocks, Deborah L. McGuiness, Peter F. Patel-Schneider, Lynn Andrea Stein: DAML+OIL (March 2001) Reference Description. W3C Note. (18 December 2001)
7. Deborah L. McGuinness, Frank van Harmelen: OWL Web Ontology Language Overview. W3C Proposed Recommendation (15 December 2003)
8. Natalya Fridman Noy, Mark A. Musen: An Algorithm for Merging and Aligning Ontologies: Automation and Tool Support. In: Proceedings of the Workshop on Ontology Management, Sixteenth National Conference on Artificial Intelligence (AAAI-99), Orlando, Florida, U.S.A (July 1999)
9. Jorge E. López de Vergara, Víctor A. Villagrá, Julio Berrocal: An ontology-based method to merge and map management information models. In: Proceedings of the HP Openview University Association Tenth Plenary Workshop, Geneva, Switzerland. (July 2003)
10. Monica Crubézy: The Protégé Axiom Language and Toolset ("PAL"). Protégé Project, Stanford University. (April 2002) Available at http://protege.stanford.edu/plugins/paltabs/pal-documentation/index.html
11. N. F. Noy, M. Sintek, S. Decker, M. Crubezy, R. W. Fergerson, & M. A. Musen: Creating Semantic Web Contents with Protege-2000. IEEE Intelligent Systems, Vol. 16, No. 2 (2001)