Ontologies: Giving Semantics to Network Management Models

Jorge E. López de Vergara, Víctor A. Villagrá, Juan I. Asensio[†], and Julio Berrocal.

Departamento de Ingeniería de Sistemas Telemáticos,
Universidad Politécnica de Madrid (DIT-UPM), Spain.
[†]Visiting researcher from Universidad de Valladolid, Spain.

***Abstract* - The multiplicity of network management models may imply in some scenarios the use of multiple management information languages defining the resources to be managed. Each language has a different level of semantic expressiveness which is not easily measurable. And these management information models cannot be easily integrated due to a difficult translation of the semantics they contain.**

This paper proposes the use of ontologies as a new approach to improve the semantic expressiveness of management information languages. Ontologies are being currently used, for instance, to provide web pages and web services the semantics they usually lack (what is known today as "Semantic Web"). Applying ontologies to management information languages can also be useful for the integration of information definitions specified by different management languages and for the addition of behavior information to them.

***Keywords* -** Ontology, Network Management, Management Information Language, Information Models Integration, Behavior Information.

1. INTRODUCTION

Currently, there are several integrated network management models using different technologies for resource management, such as SNMP (Simple Network Management Protocol), CMIP (Common Management Information Protocol), DMI (Desktop Management Interface) and WBEM (Web Based Enterprise Management). Even distributed processing technologies such as CORBA (Common Object Request Broker Architecture) have also been applied for network management [1].

Every integrated management model has identified the need of a management information definition language to describe the resources to be managed (its management domain), in order to ensure the cooperation between managers and agents. Therefore, each model has defined its own management information definition language, with different capabilities and expressiveness:

- SMI (Structure of Management Information), with its different versions, for SNMP.
- GDMO (Guidelines for the Definition of Managed Objects) for CMIP.
- MIF (Management Information Format) for DMI.
- MOF/CIM (Managed Object Format/Common Information Model) for WBEM.
- IDL (Interface Definition Language) for CORBA.

The problem arises when different management technologies are used for the same networked system: Interoperability among all different management models involved is necessary to provide a unified view of the whole managed system. However, only syntactic translations among management languages have been applied to date. This implies that when the same resource is described with two different management information definition languages, it is possible to apply a direct translation between the defined structures of the descriptions, but not between their meanings. This is a problem to achieve an integrated management using different models simultaneously: Semantic interoperability becomes necessary to solve this question.

Meanwhile, ontologies have been successfully used to solve similar semantic problems in other domains such as the Semantic Web, where these knowledge-based techniques provide web pages and web services the semantics they usually lack.

This paper explores how ontologies can also be useful for network and system management as a way to unify heterogeneous definitions of management information including aspects not addressed before, helping to reach a semantic interoperability of different management models and languages:

- A framework can be established to compare different management information languages from an ontology viewpoint. All these languages have different levels of expressiveness, so the same resource can be defined in multiple ways depending on the used model. As ontologies provide all the necessary constructs to represent the semantics of specified information, these constructs can be used to compare the semantic expressiveness of management information languages in a neutral way.
- Semantic interoperability can be achieved improving existing syntactic translations by integrating in a semantic way the management information from different models, mapping and merging ontology-based information definitions. For this, a methodology is proposed that does not only apply recast translations [2] but also merges different definitions in a common one and declares the necessary mapping rules.

Moreover, behavior characteristics could also be included to this merged management information by using rules, axioms and constraints usually contained in ontologies. Currently, GDMO includes a `BEHAVIOR` template and SNMP SMI was redefined in its second version to improve its semantics. However, these approaches are only human-oriented: it is difficult for a computer to understand these `BEHAVIOR` or `DESCRIPTION` statements. Ontology-based behavior information could be understandable by intelligent network managers.

2. ONTOLOGIES AS A KNOWLEDGE REPRESENTATION PARADIGM

Ontologies are one of the main approaches being used in the Knowledge Management and Artificial Intelligence fields to solve semantic issues.

To understand how ontologies can be applied to improve the information management modeling it is necessary to explain previously what they are. There are many definitions of Ontology, but one that is both general and complete is "an explicit and formal specification of a shared conceptualization" [3]:

- It is explicit because it defines the concepts, properties, relationships, functions, axioms and constraints that compose it.
- It is formal because it is machine readable and interpreted.
- It is a conceptualization because it is an abstract model and a simplified view of the existing things it represents.
- Finally, it is shared because there has been previously a consensus about the information and it is accepted by a group of experts.

In brief, it can be said that an ontology is the definition of a set of concepts, its taxonomy, interrelation and the rules that govern such concepts.

Ontologies can be classified in two different groups: lightweight and heavyweight. The former includes those ontologies capable of modeling the information referred to a domain, but it is difficult to reason with them because they do not contain axioms or constraints. Ontologies of the latter group include all elements that enable the inferences about the knowledge they contain.

In this way, existing management information models could be understood as lightweight ontologies: Models like Internet MIBs or CIM schemas define the information of the management domain in a formal way and they have been agreed in working groups. However, their semantics is limited. Its improvement would eventually enable the inference of knowledge based on existing one.

There are many terminologies when dealing with ontologies depending on the paradigm used for knowledge representation. Although the terminology is different, direct mappings can be done with most of them, and similarly with information management languages. Some paradigms for knowledge representations in ontologies are:

- <u>Semantic Networks</u>. They usually deal with concepts, instances, relations and properties. Concepts express any kind of static and cognitively autonomous semantic phenomena. Entities belonging to the interpretation of the concept are instances. Properties are characteristics of the entities.
- <u>Frame-based representation</u> uses classes, instances, slots, and facets. Comparing to Semantic Networks, classes map into prior concepts and instances remain the same. Slots are similar to properties. Finally, facets are properties of the slots.
- <u>Description logic</u> uses concepts, roles and individuals. In this case, roles map into properties and individuals, into instances.
- Finally, <u>object-oriented paradigm</u> uses classes, objects and attributes. Consequently, objects are instances of classes, and attributes are their properties.

The comparison of management information languages from the ontology viewpoint will be done taking into account several proposals about comparing different ontology languages. Next sections will show how these approaches can be applied for comparing and integrating management information.

3. MANAGEMENT INFORMATION LANGUAGES COMPARISON

A first step to reach a semantic integration of management information definitions is to analyze the implicit semantics included in those definitions. For this, this section uses an ontology-based approach to study and compare different languages usually applied for the definition of management information for networks and systems. Ontology languages have semantic expressiveness, and so, analyzing expressive capacity in these terms should be as neutral as possible. This proposal is based on some existing approaches to compare ontology languages [4, 5], which evaluate the elements these languages can express.

There are also some proposals about comparing management information languages from other viewpoints different from ontology-based representations. For instance, [6] compared GDMO, SMI and IDL in terms of their metamodel. CIM and GDMO characteristics have been compared in [7]. The list of languages compared in this section includes GDMO, SMIv2, MIF, IDL, MOF/CIM and SMIng.

To present this analysis, first of all, languages being compared will be presented in chronological order, with a brief review of their characteristics. Then, a comparison of their expressiveness will be done matching them with the constructs commonly included in ontology languages.

3.1. Languages presentation

3.1.1. GDMO (Guidelines for the Definition of Managed Objects)

GDMO was specified as part of the OSI Systems Management - Structure of Management Information, and was adopted in TMN to define the Generic Network Information Model.

This language uses the object oriented paradigm, but it adds some other characteristics to allow a better reuse of the defined management information. The reuse also adds a lot of complexity to this language.

Definitions using these Guidelines normally have classes that include some packages. Packages are composed of attributes, actions (name used for methods) and notifications, which usually have different properties or facets such as data type or behavior.

3.1.2. SMIv2 (Structure of Management Information version 2)

SMI was specified to define the management information for SNMP agents in Internet. Its improved second version is the current Internet management information language standard, keeping the simple philosophy of SNMP.

One of the reasons for its simplicity is the usage of a reduced set of object-orientation concepts as the basis for the management information models: managed objects are defined but they do not use concepts such as encapsulation, inheritance, polymorphism, etc. These objects can be presented as scalar variables or by defining tables. However, tables can also be seen as classes, where attributes are the columns of the table, and each row is an instance of the class. Another proposal [8] has gone further, identifying some kinds of relationships, including inheritance among different tables.

Facets such as syntax or description are also included in definitions using this Structure.

3.1.3. MIF (Managed Information Format)

MIF was specified by DMTF, and used to define desktop related information. Its paradigm is somehow similar to SMI: Only groups of simple variables (here called attributes) and tables can be defined. Nevertheless, it is even simpler, as table keys are always internal to that table, and thus, associations cannot be defined in this way.

Once again, some different facets are defined for every attribute.

3.1.4. IDL (Interface Definition Language)

IDL is the language used in CORBA to define distributed object interfaces. It is an object oriented model in which classes are interfaces that have attributes and methods.

This language has been included in the set of management information languages because there are many proposals for the use of CORBA for network management. Moreover, ITU-T M.3120 Recommendation uses it to define TMN (Telecommunication Management Network) management information, and also applies it in UTRAD (Unified TMN Requirements, Analysis and Design).

3.1.5. MOF/CIM (Managed Object Format / Common Information Model)

MOF is the syntax used for CIM, which is the language used in WBEM. It has been defined again by DMTF, but this model is object-oriented and much more powerful than MIF. However, its complexity is lower than GDMO. This language can also be translated to XML (Extended Markup Language) to exchange the information, which is sometimes known as XML-based management.

With this Format, classes can have properties (the name they use for attributes) and methods. Other facets can be defined, thanks to the possibility of specifying new qualifiers.

3.1.6. SMIng (Structure of Management Information, next generation)

The next generation of SMI has been proposed by the IRTF Network Management Research Group to define management information and to integrate SMI and SPPI (Structure of Policy Provisioning Information), avoiding the use of ASN.1.

Even though it is not a standard and it is still being defined, SMIng expressiveness is bigger than SMI, because it is an object-oriented language. In addition to classes with attributes and events, it is also possible the definition of extensions, which specify new structures by providing the syntax they must comply with. Integration with XML is also being studied.

3.2. Comparison

The structure of this comparison is mainly based on [4] and [5]. The former work provides a framework with a set of criteria to compare ontology languages, studying their semantic expressiveness. The latter specializes this framework to analyze the integration of information based on ontologies. The criteria explained in these proposals has been applied to network management languages as ground of comparison.

Main kinds of components used to describe the domain knowledge are concepts, taxonomies, relations, functions, instances and axioms. This section will detail each one, comparing all presented languages in these terms.

3.2.1. Concepts

Concepts or classes are the most important elements to define information. The following items identify the expressiveness of a language when defining classes.

- Metaclasses: This item deals with the possibility of defining classes as instances of other ones. Most languages do not allow this construction. Just in MOF/CIM and SMIng it is possible to define new statements with qualifiers and extensions respectively, which indirectly makes feasible the redefinition of classes.
- Partitions: This concept, very common in ontology languages, related to the definition of sets of disjoints classes, does not exist in management information languages.
- Attributes: Concepts usually have attributes, and all management information languages allow their definition.
  - Local scope: Attributes can be defined inside or outside the class. Only GDMO, with attributes that belong to a package, defines attributes outside the scope of a class.
  - Instance attributes or templates: They are attributes whose value may be different for each instance of the concept. All management information languages allow instance attributes.
  - Class or own attributes: They are attributes whose value, which can be eventually modified, must be the same for all instances or concepts. In this case, just MOF/CIM can explicitly define such attributes, using the `static` qualifier.
  - Polymorph attributes: They are attributes with the same name and different behavior for different concepts. In this case, those languages that have an attribute namespace inside the class scope can have this kind of attributes. That is MIF, IDL, MOF/CIM and SMIng. SMIv2 has a single namespace for all attributes and so, it does not allow these polymorph attributes. GDMO has the allomorphism, in which an attribute can also behave as an attribute of the parent class, but this does not exactly match with the definition above.
- Facets: Attributes usually have a set of predefined properties or facets.
  - Default value: All languages except IDL can define a default value for their attributes.
  - Data type constraint: All languages use data types to define their attributes. They are usually simple types except in GDMO, where all ASN.1 types can be used, and IDL that can also have structures and sequences. MOF/CIM allows arrays of simple types.
  - Cardinality constraint: It constraints the maximum and minimum number of values of an attribute. In this case, only MOF/CIM and IDL can define such constraints, and only for arrays and sequences respectively. The former language also allows the definition of maximum and minimum numbers for association, by using `max` and `min` qualifiers.
  - Documentation: It gives a natural language definition of the attribute. Most languages have a description statement. In GDMO, behavior template is usually taken for this. IDL does not have any kind of documentation, although comments can be used in this case.
  - Operational definition: It could include the definition of a formula or rule to be used when obtaining a value for that attribute. None of the studied languages included directly such facet.
  - Addition of new facets: As stated with respect to metaclasses, only MOF/CIM and SMIng can define new facets, by using qualifiers and extensions respectively. `Pragma` statement could be used in MIF to define particular facets.
  - Other facets: All analyzed languages have other facets, such as the access, the key or index, and the identifier, which seem to be common facets in these management information languages.

3.2.2. Taxonomies

Concepts are usually organized in taxonomies, with generalization/specialization relationships among them. There are some taxonomic characteristics that can be analyzed:

- Subclass of: It specializes general concepts in other ones more specific. Object oriented management languages allow the definition of subclasses, being GDMO and IDL the only ones in which multiple inheritance can be defined. In SMIv2, the use of an external index or the augments clause can be considered as a kind of inheritance.
- Disjoint decomposition: It defines explicitly a partition as subclass of a class. Two classes are disjoint if they are sibling classes and there cannot exist a child class of both of them at the same time. None of the studied languages can define such kind of decomposition.
- Exhaustive subclass decomposition: It defines a partition but in this case, the parent class is the union of all subclasses. As in disjoint decomposition, no language is prepared to define this decomposition.
- Not subclass of: It states that a class is not a specialization of another one. Once again, none of the languages had this construction. MOF/CIM defines the `final` qualifier to prevent a class from being specialized, but it is not the same concept.

3.2.3. Relations and functions

Relations represent a type of interaction between concepts. Functions provide a unique value from a list of valued arguments.

- Definition of relations/functions: Object oriented languages allow the definition of relations. GDMO relationships are aggregations, although GRM (Generic Relationship Model) can be used for other type of associations. Relations can also be defined in IDL and SMIng using the MOF/CIM association approach (classes with two or more references). In SMIv2, an association can be defined between classes using external indexes. With respect to the functions, only GDMO, IDL and MOF/CIM allow the definition of functions.

- Data type constraint: It defines that the type of the arguments is constrained. As all management information languages are typed, all of them use those types in all constructs, and so, the type of the arguments is constrained.
- Operational definitions: It defines the way to obtain or infer values of arguments with procedures and formulas, or defining their semantics using axioms or rules. This is one of the main lacks of management information languages. GDMO has the behavior construct, but it is usually used as a description. Some proposals, which are not part of the standard, have been done to formalize it creating a specific behavior language or using other specification languages like SDL (Specification and Description Language).

3.2.4. Instances

They represent elements of a given concept, a relation or an assertion.

- Instances of concepts: For every concept, it is possible to have instances in any of the network management models included here.
- Facts: They are instances of relations. In this case, just those models that can define relations can have facts. This means that all management models with object oriented languages can have them.
- Claims: They represent assertions of a fact by an instance. This concept, common in the semantic web, is not included in any of the management models being studied.

3.2.5. Axioms, production rules and reasoning

Axioms model expressions that are always true, and are usually used to define constraints. Production rules follow the if-then structure, and are used to express sets of actions. Reasoning processes can be carried out by following the different relations defined in the represented knowledge.

Management information languages do not include any of these elements. This means that it is not possible to define constraints or procedures with them, which would be very useful to define behavior. As stated before, GDMO includes a behavior template, but it is open to any definition. Other languages do not even include such construct, although extension mechanisms existing in MOF/CIM and SMIng could be used for this.

3.2.6. Comparison table

Table 1 shows a summary of all compared languages in terms of ontology elements. A plus sign (+) has been placed when the language has that element and a minus (-) when the language has not. An asterisk (*) is used when the language has a similar functionality but it is not directly applicable.

Table 1. Summary of the compared characteristics

LANGUAGE	GDMO	SMIv2	MIF	IDL	CIM	SMIng
CONCEPTS						
Metaclasses	-	-	-	-	*	*
Partitions	-	-	-	-	-	-
Attributes						
Local scope	-	+	+	+	+	+
Instance attributes	+	+	+	+	+	+
Class attributes	-	-	-	-	+	-
Polymorph attributes	-	-	+	+	+	+
Facets						
Default value	+	+	+	-	+	+
Type constraint	+	+	+	+	+	+
Cardinality	-	-	-	*	+	-
Documentation	*	+	+	-	+	+
Operational definition	-	-	-	-	-	-
New facets	-	-	*	-	+	+
Other facets	+	+	+	+	+	+
TAXONOMIES						
Subclass of	+	*	-	+	+	+
Disjoint decomposition	-	-	-	-	-	-
Exhaustive decomposition	-	-	-	-	-	-
Not subclass	-	-	-	-	-	-
RELATIONS/FUNCTIONS						
N-ary relations	+	*	-	+	+	+
Functions	+	-	-	+	+	-
Type constraint	+	*	-	+	+	+
Operational definitions	*	-	-	-	-	-
INSTANCES						
Instances of concepts	+	+	+	+	+	+
Facts	+	*	-	+	+	+
Claims	-	-	-	-	-	-
AXIOMS, RULES	*	-	-	-	-	-

As can be seen, MOF/CIM is the language with best semantic expressiveness, with 16 (+1*) elements. SMIng and GDMO are near MOF/CIM expressiveness, with 13 (+1*) and 10 (+3*) elements respectively.

There are some rows in which all or nearly all languages have a minus. They are mainly related to taxonomic relationships and definition of procedures or rules that describe certain behavior. In order to add more expressiveness to management information languages these elements should be included to them.

4. APPLYING ONTOLOGIES TO NETWORK MANAGEMENT MODELS

Previous section showed and compared the expressiveness of the different management information languages, which can be useful when making a direct translation from one language to another one. However, this translation, usually called recast, only produces an information module syntactically equivalent to other one, whereas implicit semantics contained in the original module is not considered for the translation. This question arises because management information languages lack formal behavior definitions, only including natural language descriptions of defined classes and attributes.

Recast translations have been defined between SMI and GDMO (ISO-Internet Management Coexistence proposal), and between SMI, GDMO and IDL (Joint Inter-Domain Management proposal). Other proposals have defined other recasts from SMI, GDMO or IDL to MOF/CIM [2].

Even if future management models use a unified information model such as CIM, legacy agents will exist with information defined in other language. Usually, current management applications parse this information and rewrite it in MOF/CIM, reaching a syntactic integration. If translations are mainly recasts, the CIM approach defining a common model integrating all existing management information modules will not work: This model, although written using the same language, will be just a set of unconnected classes, and even reaching a syntactic integration, their implicit semantics will not be integrated. As stated in prior section, taxonomies and relationships are important elements of an ontology: If all concepts are not under the same taxonomy and do not include enriching relationships, the ontology is not complete.

For instance, if an SNMP MIB such as the HOST-RESOURCES-MIB is translated to MOF/CIM, a set of classes such as `hrDevice` or `hrPrinter` will be derived, but if they are not related to CIM classes such as `CIM_LogicalDevice` or `CIM_Printer` respectively, then this information will not be seen as part of a unified view.

Moreover, there is a problem when different management domains have overlapped concepts in their respective information models. The ideal solution would consist in reflecting those concepts into CIM concepts, by using the "scratch pad" mechanism, defining `MappingStrings` qualifiers. The problem of this qualifier is that it only refers to the definition of the concept in the other model, and it does not explain how to translate their instances. Other qualifiers could be defined for this, but they have not been specified by the DMTF.

Once again, ontologies can help in this problem, as there are many proposals about merging and mapping them. This section explains two advantages of applying ontologies to current network management scenarios: the definition of an integrated information model, which could be an improved version of CIM, and how to add formal behavior information to the information models, which would enable a potential manager to reason with it. Applying ontologies might be complex, but it also improves the operation of integrated and intelligent management systems, which thanks to this approach can use a single information model for all management tasks.

4.1. Integrating management information models

Translations among several management models are normally needed in those scenarios where they are used for the same domain, in order to apply a coordinated management policy to the domain. There are two possibilities for this: The first one is to define translations between every two models. The second one is to define an information model containing existing ones. The second approach is better if the number of models is high, being proved that $2 \cdot n$ translations are just needed, instead of $n^2 - n$.

Second solution would define an information model containing all defined information. In an ideal case, shown in Figure 1, all merging models have the same weight and the same overlapped areas. In this case, final model should be a new one, which would be a union of all them. Mappings should be defined between the final model and each of the merged models.

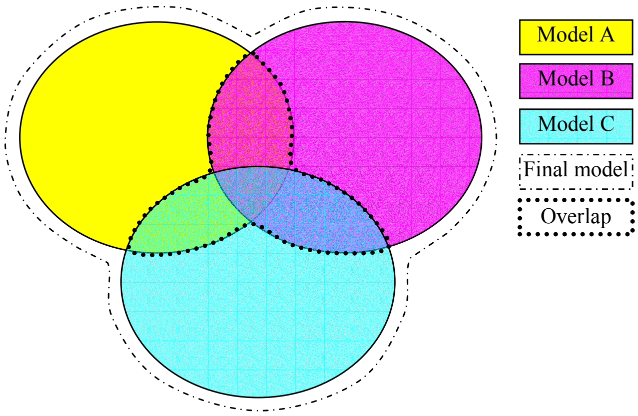

Figure 1. Ideal case, with all merging models of the same weight.

However, as seen in Figure 2, such a symmetric ideal case does not exist. It would be more appropriate to create the final model by taking the biggest one, and adding the information that is not overlapped. With this approach, the mapping task is reduced to define it only for those models merged to the bigger one.

In this way, the definition of CIM in WBEM has been a good approach as a common information model: It defines information for most management domains, including a subset of other management models such as many Internet MIBs and the Master MIF. In addition, it uses the most expressive management information language of those analyzed in section 3, being able to contain the information defined in other ones that are less expressive.

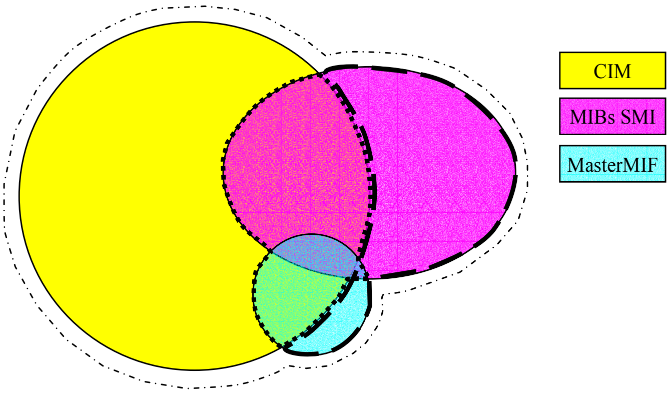

Figure 2. Real case: CIM, SMI MIBs and Master MIF have different weights and different overlaps.

Nevertheless, semantic interoperability is not completely achieved in CIM as shown previously and thus, this model should be extended, using ontology characteristics. This CIM-based ontology should contain all information defined for other management models: Internet MIBs, Master MIF or TMN M.3100 should be merged with CIM, including all necessary mapping rules. Translations should be done including semantics of all models.

There are several available tools for merging and mapping ontological definitions which can help in the creation of this extended CIM model (e.g., [9] and [10] respectively). They do not provide a completely automated algorithm, but give some heuristics based on lexical coincidences or hierarchy relationships that can be used for semiautomatic merging and mapping, which facilitates the task.

To obtain the desired general common management model, the following set of steps is proposed:

1. First of all it is necessary to translate every model to the same definition language. Reusing an existing management information language is better than specifying a new one to avoid more heterogeneity and leverage existing tools. As seen before, best approach is to use MOF/CIM as such language. Translations would be done by applying recast techniques. This translation is necessary to have all information models defined in the same way, making it possible their merge.
2. Once all models are defined in the same language, two steps are done in parallel:
    a. Merging: All models are merged in a global ontology. Those concepts that are overlapped have to be redefined to a common and unique concept. Same thing should happen with attributes. Other concepts that are not overlapped just have to be included in the model, but making relationships (associations or specializations) with the global ontology, to be able to reason with them.
    b. Mapping: At the same time, mapping rules have to be defined to translate instances from one model to the global ontology. For this, a simple string such as the `MappingString` qualifier used in CIM is not enough. Once again, there are some proposals about defining mapping ontologies that can be reused.

The structure of a very simple mapping ontology is depicted below in Figure 3: Each possible element of the ontology (concepts, attributes, relations) has a translation formula. Other attributes can be added to that element, such as a reference to its definition (it could be an OID, for instance). At the same time, each formula has the set of source and target elements that take part on it, and an expression used to translate from the set of source elements to the set of target elements. Associations to mapped elements and inverse formula are also included.

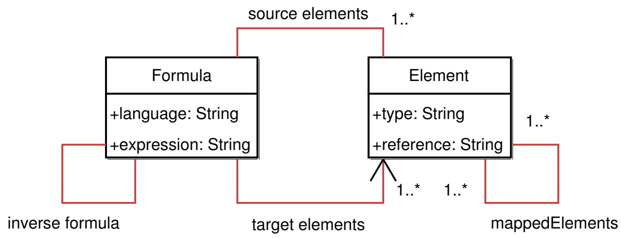

Figure 3. A simple mapping ontology.

A manager based on both the global ontology and the mapping ontology would work, for example, in the following way: If it needs to obtain all the instances of a certain element of the global ontology, it would search for it in the mapping ontology, finding also related formula and corresponding elements of the merged models. Expression contained in the formula would then be applied to translate elements of the merged models to fit in the global ontology, and the desired instances would be obtained.

4.2. Adding behavior information

A step forward integrating management information can be achieved by adding a set of formulas, axioms or constraints to the defined management ontology. This would make it heavyweight, as these elements would make it possible the inference of knowledge from the information contained in the ontology.

In this way, descriptions of the classes and attributes of the management information would include constraint definitions, such as: "a language that is used as a default by the Printer should also be listed in `LanguagesSupported`", contained in `CIM_Printer` class. Other times, a manager might want to comply with some policies. For instance: "The `AvailableSpace` of a `CIM_FileSystem` instance should be bigger than a 10% of the `FileSystemSize`".

As seen before, only GDMO has defined a very general template for the behavior of defined elements, and another more expressive language, MOF/CIM, does not include any expressiveness about declaring axioms or constraints. Nevertheless, MOF/CIM allows extensions by defining new qualifiers. Then, if this is the language used for the global ontology, a new qualifier, constraint, should be defined:

```
Qualifier Constraint : string = null,
Scope(any);
```

That is: a constraint is a qualifier that is defined as a string without any default contents that can be used for any element of the MOF/CIM language (mainly Classes, Properties and Methods). The language used for declaring constraints is not defined here. It could be the Object Constraint Language, defined as part of UML. Other option is the use of a language such as Prolog, LISP, or description logic languages as used in current Semantic Web proposals.

This behavior should be included in the ontology usually by the management information modelers, with basic and generic reasoning over the defined resources. It can be processed later automatically by intelligent management systems that reason with the provided ontology-based information definitions. Common managers can also read the information contained in the ontology, because MOF/CIM syntax remains the same. However, they will not be able to use the added behavior information.

5. Conclusions

This paper has applied ontologies concepts and principles to the definition of management information. This approach has been proved to be useful in many issues: first, when comparing languages, next, when integrating management information models, and also, when adding behavior information to complete this integrated model:

When comparing languages, it has been found that MOF/CIM is the language with better semantic expressiveness. However, it lacks the definition of taxonomic relationships and constraints for the description of behaviors.

When integrating management information models, the mapping and merging tasks can be done taking advantage of ontology tools, creating a global management ontology with an associated mapping ontology. This solution does not impose any restrictions, but due to the need of manual intervention to check performed tasks, it may take a long time when applied to huge information models. However, this time is much shorter than doing this work by hand.

Finally, different behavior rules can be added to that global ontology to improve it, allowing the definitions of constraints usually defined now in natural language in the definitions of the management information.

All these features could be used to develop a management system, a so called ontology-based manager that takes advantage of this approach, integrating all heterogeneous management models in a smart way, bearing in mind the semantics of the defined management information.

The ideas presented in this paper are currently being implemented, validating the proposed approach. Current developments include an adaptor to automatically load MOF/CIM and SMI specifications in an ontology tool [9], dealing also with SMI particularities. At the same time, this tool is being adapted to the peculiarities of the management information when merging and mapping it.

6. References

1. Heinz-Gerd Hegering, Sebastian Abeck, and Bernhard Neumair, *Integrated Management of Networked Systems*, Morgan Kaufmann, 1999.
2. Jorge E. López de Vergara, Víctor A. Villagrá, and Julio Berrocal, "Semantic Management: advantages of using an ontology-based management information meta-model," *Proceedings of the HP Openview University Association Ninth Plenary Workshop (HP-OVUA'2002)*, Böblingen, Germany, 11-13 June 2002.
3. R. Studer, V.R. Benjamins, and D. Fensel, "Knowledge Engineering: Principles and Methods," *Data & Knowledge Engineering*. 25: 161-197. 1998.
4. Óscar Corcho and Asunción Gómez-Pérez, "A Roadmap to Ontology Specification Languages," *Proceedings of the Twelfth International Conference on Knowledge Engineering and Knowledge Management (EKAW'2000)*, Juan-les-Pins, France, October 2000.
5. H. Wache, T. Vgele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hübner, "Ontology-based integration of information - a survey of existing approaches," *Proceedings of the Workshop Ontologies and Information Sharing, IJCAI 2001*, Seattle, Washington, U.S.A., August 2001.
6. Bela Ban, "Towards an Object-Oriented Framework for Multi-Domain Management," *Proceedings of ECOOP '96 Workshop on Systems and Network Management*, Linz, Austria, July 1996..
7. O. Festor, P. Festor, Laurent Andrey, and N. Ben Youssef, "Integration of WBEM-based Management Agents in the OSI Framework," *Proceedings of the Sixth IFIP/IEEE International Symposium on Integrated Network Management (IM'99)*, Boston, Massachusetts, U.S.A., May 1999.
8. J. Schönwälder and A. Müller, "Reverse Engineering Internet MIBs," *Proceedings of the Seventh IFIP/IEEE International Symposium on Integrated Network Management*, Seattle, Washington, U.S.A., May 2001.
9. N. F. Noy and M. A. Musen, "PROMPT: Algorithm and tool for automated ontology merging and alignment," *Proceedings of the Seventeenth National Conference on Artificial Intelligence AAAI'00*, Austin, Texas, U.S.A., July 2000..
10. Alexander Maedche, Boris Motik, Nuno Silva, and Raphael Volz, "MAFRA – A MApping FRAmework for Distributed Ontologies," *Proceedings of the Thirteenth European Conference on Knowledge Engineering and Knowledge Management EKAW'02*, Madrid, Spain, October 2002.

7. Biographies

Jorge E. López de Vergara (jlopez@dit.upm.es) holds a research grant at Technical University of Madrid, Spain (DIT-UPM). He received his MSc degree in Telecommunications from that university in 1998 and is currently finishing his PhD in Telematics Engineering at DIT-UPM. He has participated in several Spanish and European Union Research Projects. His current research topics include network, service and distributed

application management, and ontologies for management information integration.

VÍCTOR A. VILLAGRÁ (villagra@dit.upm.es) is an Associate Professor of Telematics Engineering at the Technical University of Madrid, Spain (DIT-UPM). He received his MSc degree in 1989 and his PhD in Telematics Engineering in 1994, both from the Technical University of Madrid. He has participated in National and International Research Projects focused on service and network management and its application to the management of new distributed e-services. His current research interests include management of distributed environments and network security.

JUAN I. ASENSIO (juaase@tel.uva.es) is an Associate Professor of Telematics Engineering at the University of Valladolid, Spain. He received his MSc degree in 1995 and his PhD in Telematics Engineering in 2000, both from the University of Valladolid. He has participated in several Spanish and European Union Research Projects related to networked systems management. His current research interests include development and management of distributed, component-based software systems.

JULIO BERROCAL (berrocal@dit.upm.es) is a Professor at Technical University of Madrid, Spain (DIT-UPM). He received a Telecommunication Engineer degree in 1983 and a PhD in Telecommunication degree in 1986, both from UPM. He has participated in several R&D projects of the European Union Programmes and Spanish Programmes. His current research interests are in management of telecommunication networks and services, multimedia networking and broadband networks.