A CIM-based Ontology for Semantic IT-Management

Andreas Textor
RheinMain University of Applied Sciences, Wiesbaden, Germany

Motivation

In the domain of IT management, numerous models, protocols and tools have been developed. Notable models include the OSI network management model (CMIS/CMIP) and the still widely used simple network management protocol (SNMP). A more recent approach to specify a comprehensive IT management model is the Common Information Model (CIM [1]), a widely recognized Distributed Management Task Force (DMTF) standard.

The more complex an environment gets, the higher the need for comprehensive, highly automated IT management. To achieve this long-term goal, the various available sources of information need to be combined. However, syntactic translations from one model to another are often not sufficient, as the same concept can be expressed in different ways in two different domains [2]. For this reason, several researchers have proposed to use ontologies to unambiguously and comprehensively model IT environments. An ontology is a formal representation of a set of concepts within a domain and the relationships between those concepts. It can also contain behaviour rules and instance data that represents concrete entities. Thus, ontologies are a possible formal base for automated IT management.

Related Work

To allow the application of ontologies to IT management, a suitable domain model is required. In [3], CIM has been proposed for this purpose, as it is an actively used, maintained and freely available standard. As CIM is also the foundation for other models, such as the storage-related standard SMI-S (Storage Management Initiative Specification) of the SNIA (Storage Networking Industry Association, [4]) and is used in the DMTF SMASH specification (Systems Management Architecture for Server Hardware, [5]), the approach presented here is also applicable in environments where one of these is employed.

The authors in [6] point out that CIM is usable for inferring properties about distributed systems, but is a semi-formal ontology with limited support for knowledge interoperability and aggregation, as well as reasoning. This firstly means that support in CIM for the specification of formal rules is very limited. Secondly, the interrelation of the specified information with knowledge from other domains (which are usually not specified in CIM) is not easily possible in a way that supports automated management, e.g. business processes that refer to physical or logical IT systems. For the approach presented here, the first point is of primary importance, while in the long run, the connection to information models from different domains (or different views of the environment in a business) will become more relevant.
The idea of using semantic web technologies for IT management has been examined in several publications, as well as the conversion of CIM to OWL. One notable conversion approach is described in [7], where the authors introduce a meta-ontology that is used to model the CIM constructs that have no direct OWL correspondence. However, they do not describe how the meta-ontology is constructed and how certain elements are translated, such as CIM methods, data types of properties and constraints (e.g. MaxLen for strings).

Approach

In this work, we construct an IT-management domain ontology in the Web Ontology Language (OWL, [8]) by transforming the CIM schema into OWL. This way, the rich and extendable model of the CIM schema can be used, while simultaneously a formal knowledge base is created that not only contains the domain model, but can also express rules and incorporate instance data, i.e. all the real entities of the managed system. The ontology can be enriched with rules formulated in the Semantic Web Rule Language (SWRL, [9]). SWRL rules can be directly embedded in OWL ontologies. That way, a comprehensive and formal description is formed that can be used as the basis for automated management.

Details about the transformation of CIM to OWL are described in [10]. Basically, each element is translated into a corresponding OWL construct, or, if not possible, modelled using OWL constructs, or created as an element in the CIM Meta Ontology. As an example for the translation of elements, figure 1 shows the translation of CIM properties to OWL constructs. For each property, a subclass of the meta ontology class CIM_Value is created. Additionally, an OWL object property is created that connects the property class and the class which contains the property.

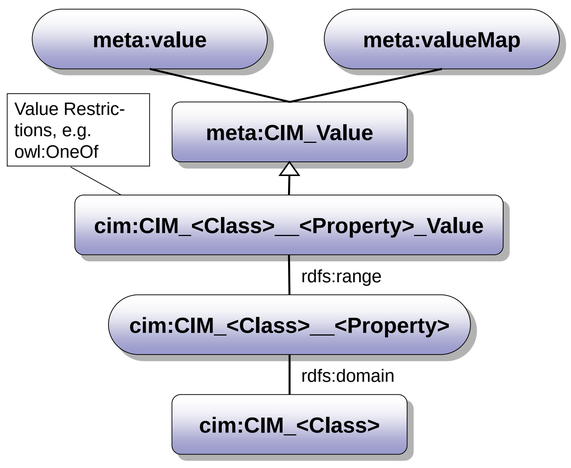

Fig. 1: Translation of CIM Properties

Based on the ontology that contains the domain model and rules, a management system is constructed. This management system, which is shown in figure 2, loads the CIM ontology and SWRL rules and then starts to read runtime information from the system under management by querying a CIM Object Monitor (CIMOM). Returned information is transformed into OWL instance data which matches the ontology entities of the previously loaded CIM ontology. As the ontology combines model and instance data, the SWRL rules can refer to both structural and instance data. A semantic reasoner component evaluates the ontology and the rules contained within, and can trigger reactions based on the evaluation results, thus creating a feedback loop to the managed system.

For automated management of the environment using the CIMOM, a rule engine is in any case necessary. To achieve the main goal of the approach presented here, it could be argued that an ontology to represent the domain model is not necessary. However, using the Semantic Web standard formats OWL and SWRL not only solves the problem of how to apply rules to CIM (which has no built-in rule format), but also allows the application of existing tools such as model editors and semantic reasoners. More importantly, it enables the connection of the technical domain model that represents the system under management to other domains. While CIM is approriate for a technical view, entities can be connected to other business views that are modelled as OWL ontologies, such as cost and accounting models, business processes etc. Possibilities of this approach will be investigated in the future.

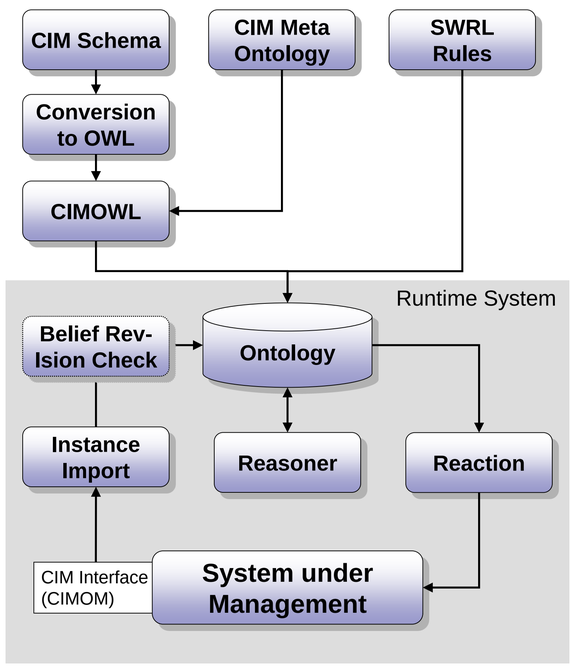

Fig. 2: Architecture of runtime system

Status

A proof of concept was implemented that manages a Linux file system. File system information is acquired via the OpenPegasus CIMOM, and rules are evaluated that monitor file sizes. Rules include checks if sizes of certain files or directories excess a given threshold and are older than a given reference file. If that is the case, actions can be taken, such as moving or compressing folders.
In addition to the evaluation of rules at runtime, the ontology can be queried for runtime information using the SPARQL Query language [11]. In contrast to the CIM Query Language, this also allows querying facts that were reasoned by a semantic reasoner, and which were not explicitly given in the original model, for example, enumerate all files that are older than a given reference file. Both the translation tool for the CIM to OWL translation and the runtime management system were implemented, and performance tests were carried out. The CIM schema with approximately 1400 classes is converted to about 100,000 OWL axioms. The not yet optimized runtime system takes between 8 and 80 seconds on an Intel Core 2 Duo with 2 GHz and 2 GB RAM for one reasoning cycle, when the ontology contains the full CIM schema and up to 100,000 instances.

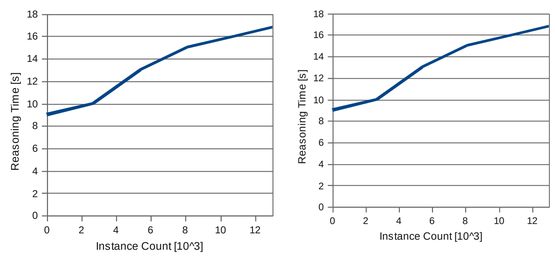

Fig. 3: Reasoning Performance

Figure 3 shows the performance of the runtime system and the time required for reasoning. For this diagram, the computer described above was used and the ontology used for reasoning consisted of two parts: The first part is the CIMOWL ontology with 70,000 axioms, the second part was a varying amount of OWL instances representing CIM_DataFiles of a server file system. Each CIM_DataFile is represented as a set of 27 OWL instances, which include the instance for the file and one instance for each of the file attributes (file size, modification time, etc.). SWRL rules to categorize files by size were loaded into the ontology.

The translation tool for the CIM to OWL translation was not only used on the CIM schema, but also on the SMI-S 1.5.0 schema files, as well as the VMware CIM SMASH/Server Management API for ESX.
Based on these results, next steps include performance optimizations and the application of the management system to automated storage management and/or automated virtual machine management together with a partner.

References

[1] Distributed Management Task Force, “Common Information Model (CIM),” http://www.dmtf.org/standards/cim/.
[2] J. E. L. De Vergara, V. A. Villagrá, and J. Berrocal, “Semantic Management: advantages of using an ontology-based management information meta-model,” in Proceedings of the HP OpenView University Association Ninth Plenary Workshop, 2002.
[3] J. E. L. De Vergara, V. A. Villagrá, J. I. Asensio, and J. Berrocal, “Ontologies: Giving Semantics to Network Management Models,” IEEE Network, vol. 17, May/June, 2003.
[4] Storage Networking Industry Association, http://www.snia.org/.
[5] Distributed Management Task Force, “Systems Management Architecture for Server Hardware (SMASH),” http://dmtf.org/standards/smash.
[6] S. Quirolgico, P. Assis, A. Westerinen, M. Baskey, and E. Stokes, “Toward a Formal Common Information Model Ontology,” pp. 11–21, 2004.
[7] Majewska, M., Kryza, B., Kitowski, J.: Translation of Common Information Model to Web Ontology Language. Lecture Notes in Computer Science, vol. 4487. Springer Berlin Heidelberg, 2007, pp. 414-417
[8] World Wide Web Consortium, “Web Ontology Language (OWL)”, http://www.w3.org/2004/OWL/.
[9] World Wide Web Consortium, “Semantic Web Rule Language (SWRL),” http://www.w3.org/Submission/SWRL/.
[10] A. Textor, J. Stynes, and R. Kroeger, “Transformation of the Common Information Model to OWL,” in 10th International Conference on Web Engineering - ICWE 2010 Workshops, ser. Lecture Notes in Computer Science, vol. 6385. Springer Verlag, July 2010, pp. 163–174.
[11] World Wide Web Consortium, “SPARQL Query Language for RDF,” http://www.w3.org/TR/rdf-sparql-query/.

Contact information

RheinMain University of Applied Sciences
Distributed Systems Lab

Unter den Eichen 5
D-65195 Wiesbaden
Germany

andreas.textor@hs-rm.de
+49 611 9495 1233

TEMPLATE DESIGN © 2008
www.PosterPresentations.com