Application of ontologies for the integration of network monitoring platforms

Jorge E. López de Vergara, Javier Aracil, Jesús Martínez, Alfredo Salvador, José Alberto Hernández

Networking Research Group, Universidad Autónoma de Madrid
Francisco Tomás y Valiente, 11, E 28049 Madrid, Spain
{jorge.lopez_vergara, javier.aracil}@uam.es

Abstract. This paper presents an ontology-based approach to integrate the measurements provided by different network monitoring tools and platforms. The combination of such measurements is valuable to network operators, enabling the development of new management applications. The use of ontologies provides some advantages over current syntactic solutions: classification, inference and querying capabilities are some of them. Moreover, they can reduce the complexity of information integration, providing solutions that can be applied to existing network monitoring infrastructures.

Keywords: Ontology, measurement information integration, network monitoring platform.

1 Introduction

Network management and monitoring is a key task in current telecommunication networks. Different monitoring tools and platforms have been developed over the years to obtain active and passive measurements of delay or bandwidth. The integration of such measurements can be valuable to network operators to obtain network weathermaps or network tomographies. However, this integration in a single view is difficult because each measurement platform uses its own data structures and its own interaction interfaces. The aim of the MOMENT (Monitoring and Measurement in the Next Generation Technologies) project [1] is precisely to find ways to solve this integration problem.

A possible solution is to use the works of the Open Grid Forum Network Measurements Working Group (NMWG) [2] and PerfSONAR [3]. These works provide a normalized XML syntax for measurement data and a set of normalized web service interfaces. Nevertheless, these technologies are simplistic in the following terms:

- They are based on XML Schema, which just provide a common syntax. In this way, it is not possible to infer any information directly from measurement data. This inference is only possible if a given application analyzes the data.

- Their interfaces are limited to a reduced set of operations, which cannot cover the full variety of network measurements. For instance, the SQL Measurement Archive interface just provides measurements of link utilization, link capacity, input errors, output drops and circuit status.

A more powerful solution is to deal with the information at a semantic level, enabling some degree of inference and automatic reasoning over the retrieved measurement data. At the same time, it is possible to define the information at different abstraction levels, which allows the definition of specific class of measurements that are derived from generic ones.

This paper focuses on this approach, applying the concepts provided by ontologies to address the integration of measurement information from a semantic viewpoint. For this, next section presents ontologies, providing general ideas about them. Then, it is explained how they can be applied to the MOMENT project, including the use of ths technique for classification of tools and information mappings. Finally, some conclusions are provided.

2 Ontologies

An ontology is defined in [4] as “a formal specification of a shared conceptualization”. In practical terms, an ontology is a hierarchy of concepts with attributes and relations that brings a terminology to define in consensus semantic networks of interrelated information units. An ontology provides a vocabulary of classes and relationships to describe a domain, stressing knowledge sharing and knowledge representation.

Ontologies can be useful in several aspects: their semantic definition of information enables a classification of knowledge (e.g. a tool that performs active measurement is an active tool) and inference (e.g. if a measurement is over a threshold then the network is overloaded). At the same time it is possible to query this knowledge (e.g. find all measurements whose destination address is W.X.Y.Z).

Several languages can be used to describe ontologies. Among them, the Web Ontology Language (OWL, not an acronym) [5] is especially relevant. It is a language based on XML with classification capabilities that can also be combined with other languages for inferencing and querying the knowledge base. Moreover, its distribution characteristics are very interesting to solve the integration of information from disperse sources.

Ontologies have been used in other information integration problems, including network management [6]. In such previous work, ontologies are proposed as a way to solve the heterogeneity of network management information models, following a methodology that merges all information into a single model, providing mappings from that new model to the old ones. We believe that the same solution can be applied to the MOMENT scenario.

3 Applications of ontologies in MOMENT

Once the ontologies and their utility have been presented, this section provides two examples of their application to MOMENT. The first one has been used to obtain a taxonomy of current monitoring tools and platforms. The second one is a proposal for the integration of the measurement information provided by monitoring tools.

3.1 Tools classification

It was necessary in the early stages of the MOMENT project to have a state of the art with existing monitoring tools. For this, an ontology has been created. This approach is different from the one used in MOME project, where a relational database was used [7]. The advantages of using ontologies are manyfold: the ontology can be downloaded from the web and read by anyone freely, the information is modeled in a more flexible way than using tables, this information can be later classified based on the properties of the instances, the knowledge base can be queried including semantic information (e.g. is there any tool whose input is a kind of file?), etc.

The defined ontology contains about one hundred classes. However, most of them are used as a taxonomy of measurement, input/output and control data, communication paradigm, platform, license or filter. This taxonomy can be later used for the classification of tools. Apart from these taxonomy classes, Monitoring Tool (see **Fig. 1**) is the most important of all classes, as it describes the information that characterizes such tools. For this, the following properties were specified:

- Communication Paradigm: This property indicates the communication paradigm used in the tool. Its range is the Communication Paradigm class and its subclasses.
- Contact URL: This property indicates a contact URL of the tool developers.
- Control: this property indicates how the tool is controlled. Its range is the Control IO class and its subclasses.
- Description: This property contains a description of the tool in natural language.
- Evaluator: This property provides the person that has entered the tool information in the ontology.
- Filter: This property provides information about the filtering capabilities of the tool. Its range is the Filter class and its subclasses.
- Homepage URL: This property contains the main URL of the tool in the web.
- Input: This property is about the input of the tool. Its range is the Data IO class and its subclasses.
- License: This property indicates the license of the software. Its range is the License class and its subclasses.
- Measurement: This property indicates the type of measurements that this tool can deal with. Its range is the Measurement class and its subclasses.
- Name: This property contains the name of the tool.
- Output: This property is about the output of the tool. Its range is the Data IO class and its subclasses.
- Platform: This property indicates in which platform the software runs. Its range is the Platform class and its subclasses.

- Tool Dependency: This property indicates if this tool depends on other tools in the ontology. Its range is also a Monitoring Tool.
- Version: This property contains the version of the tool.

This class was also classified in subclasses: Analysis Tools, which process data already captured and provide new data to be later visualized; Capture Tools, which acquire data from the network; and Visualization Tools, which show the monitored data to the user.

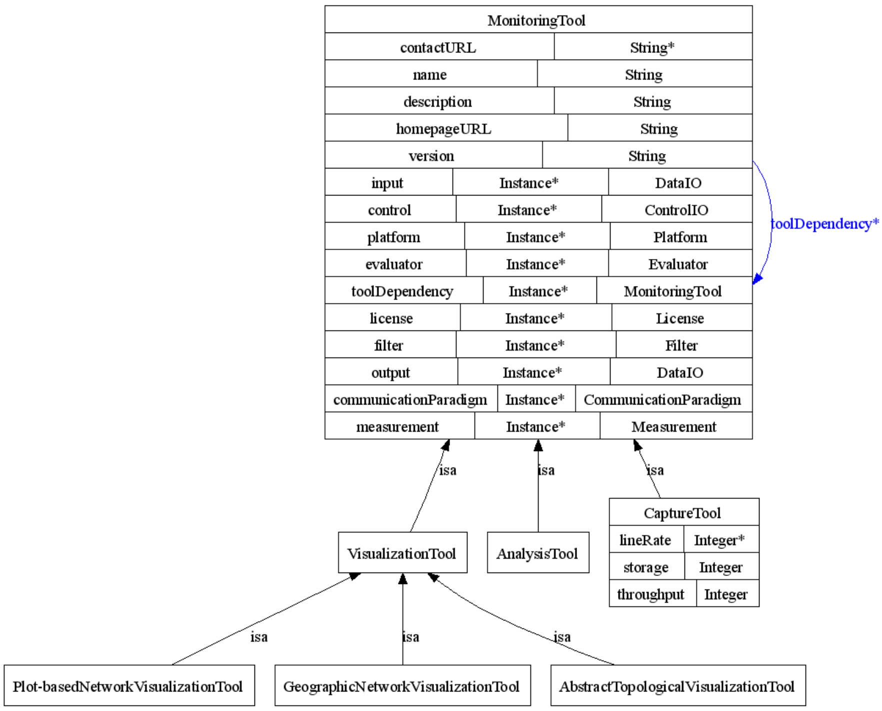

Fig. 1. Monitoring Tool classes in the ontology.

Once the knowledge base has been generated from this ontology, it is possible to query it. From these queries, it was concluded that there is a huge heterogeneity of tools, with many different data formats, modes of operation and interfaces. As a result, the design of the MOMENT integration system must be flexible enough to incorporate such differences between tools, with diverse data formats and control interfaces.

3.2 Information mapping

As stated above, it is necessary to integrate the measurement information provided by monitoring tools in a flexible way. For this, we propose the solution depicted in **Fig. 2**, which is based on the same ideas already stated in [6].

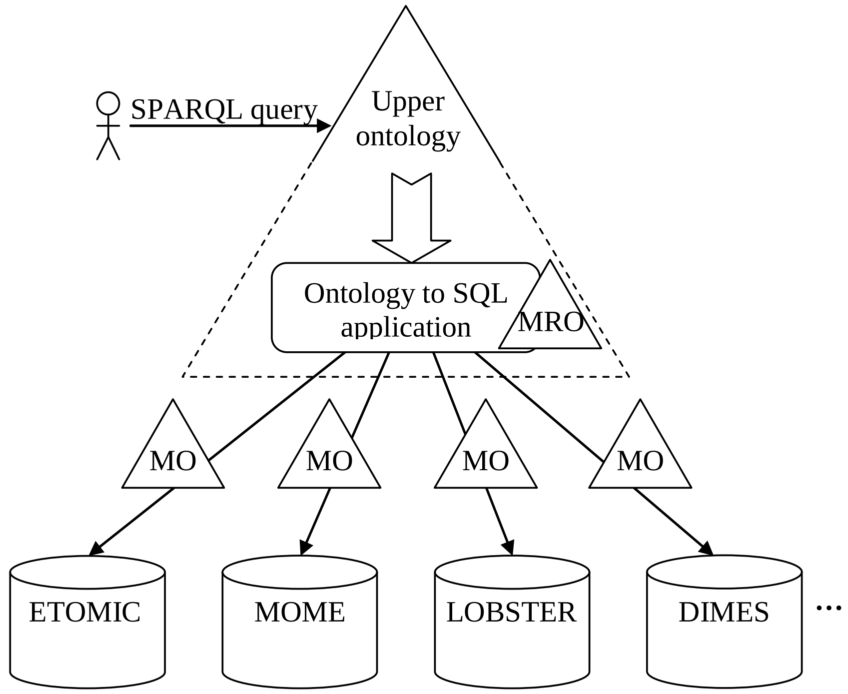

Fig. 2. Proposed Architecture.

In this solution, an upper ontology of measurements will be defined. This upper ontology contains the concepts of the different measurements. It can be based on the schemas defined by NMWF, as well as on the information provided by the monitoring tools, being a merged version of all of them. It should be noted that this upper ontology does not have any measurement instance. Those instances are contained in existing repositories (implemented as relational databases), simplifying the integration process.

It is worth mentioning that the separation between the upper ontology and the instances may be transparent to the users. When a user queries the ontology, using for instance SPARQL (a semantic web query language) [8], the query is delegated to an application that performs the following steps:

1. It looks up in the ontology of measurement repositories (MRO in **Fig. 2**) which of these repositories have measurement parameters included in the query. This ontology contains the needed information to access to these repositories: where they are and what they contain.
2. A mapping ontology (MO in **Fig. 2**) is used to translate the SPARQL query into an SQL query. This mapping ontology has to be previously defined for each repository, specifying the mapping between the relational database (tables and columns) and the upper ontology (classes and properties).
3. The results of each database query are translated back to the upper ontology concepts using again the mapping ontology.

4. The results are finally merged and provided to the user. These results can be used directly, or loaded in an inference engine to obtain more elaborated conclusions.

Given that this solution is also applicable to similar problems in other scopes, several applications exist that can provide the functionality described above, such as R2D [9].

4 Conclusions and further work

This paper has presented ontologies, and how they can be applied to the integration of network monitoring tools, which is an important issue to be addressed in MOMENT project. In this scope, an ontology-based taxonomy of current monitoring tools and platforms has been obtained. After this, a proposal has been shown for the semantic integration of the measurement information provided by monitoring tools.

Future works include the implementation and integration, as well as the validation of the ideas presented here in MOMENT project. Another important task is to find a method to homogenize the interaction interfaces with the monitoring platforms in the same way that it is done with the information.

Acknowledgements. This work has been partially funded by the European Union under the project FP7-MOMENT (INFSO-ICT-215225).

References

1. Monitoring and Measurement in the Next Generation Technologies (MOMENT). http://www.fp7-moment.eu/
2. J. Zurawski, M. Swany, D. Gunter: A Scalable Framework for Representation and Exchange of Network Measurements. In Proc. 2nd International IEEE/Create-Net Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, Tridentcom 2006, Barcelona, Spain (2006)
3. A. Hanemann, J. Boote, E. Boyd, J. Durand, L. Kudarimoti, R. Lapacz, M. Swany, S. Trocha, and J. Zurawski: Perfsonar: A service oriented architecture for multi-domain network monitoring. In Proc. Service-Oriented Computing, ICSOC 2005, Amsterdam, The Netherlands, December 2005, LNCS vol. 3826, Springer Verlag (2005) 241-254.
4. T. R. Gruber: A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, Vol. 5, No. 2 (1993) 199-220.
5. M. K. Smith, C. Welty, D. L. McGuinness: OWL Web Ontology Language Guide. W3C Recommendation, (February 2004)
6. J. E. López de Vergara, V. A. Villagrá, J. I. Asensio, J. Berrocal: Ontologies: Giving Semantics to Network Management Models. IEEE Network, vol. 17, no. 3 (2003) 15-21.
7. IST MOME: MOME Interoperability Database, Deliverable D12 (2005)
8. E. Prud'hommeaux, A. Seaborne: SPARQL Query Language for RDF. W3C Recommendation (January 2008)
9. C. Bizer, R. Cyganiak: D2R server – publishing relational databases on the semantic web. In Proc. 5th International Semantic Web Conference (November 2006)