Journal of Network and Systems Management, Vol. 15, No. 1, March 2007 (© 2007)
DOI: 10.1007/s10922-006-9058-1

Thresholds

Ontologies in the Engineering of Management and Autonomic Systems: A Reality Check

John Strassner,[1] Declan O'Sullivan,[2] and David Lewis[2]

In this article we examine the current trend to employ ontologies in the modelling of management systems and examine the barrier facing the integration of such modelling into the practical engineering of management systems.

1. INTRODUCTION

Network resources will always be heterogeneous, and thus have different functionalities and programming models. This adversely affects interoperability, due to the inherent complexity in managing networks and networked applications [1]. Existing management data, such as that found in MIBs or through the use of command line interface (CLI) languages, is represented in vendor- or domain-specific ways, with no standardization of data types and structures. Worse, the programming model of network devices varies significantly. Migrating to UML only partially solves the problem, as UML is not able to represent management concepts such as the formal semantics of sets, powerful relationships such as synonyms and "is similar to," the ability to link many representations of an entity to another entity (e.g., precise and imprecise), and other rich semantics. A better solution is needed to the modelling of management systems.

In response to the above problems, the network and system management community has recently become interested in applying ontology-based semantics to the development and operation of management systems. The standardisation of the Resource Description Framework (RDF) and the Web Ontology Language (OWL) [2] by the World Wide Web Consortium's Semantic Web initiative has

---

[1] To whom correspondence should be addressed at Motorola Labs. E-mail: john.strassner@motorola.com.

[2] Trinity College Dublin, Dublin. E-mail: Declan.OSullivan@cs.tcd.ie, Dave.Lewis@cs.tcd.ie.

provided the representational stability and an expanding toolset that in turn has motivated the increasing adoption of ontology languages in management system design. In self-managing, or autonomic systems, ontologies are seen to be particularly important, as they enable components of networks and IT systems to be reflective and reason about themselves and their peers. This is a necessary first step in enabling networks and systems to become self-configuring, and from there self-healing and self-optimising.

This existence of a sufficiently flexible and expressive set of standard languages enables the integration of a number of modelling concerns currently supported by separate languages, e.g. information modelling (CIM, SID, SMI, GDMO, DEN-ng), service modelling (IDL, WSDL), behaviour modelling (SDL, Z) and policy modelling (P-CIM, XACML, DEN-ng). In addition, ontology-based languages allow the more progressive modelling of a system, in that the completeness of the model can be reasoned over and understood at any point in the development process. This enables more incremental system modelling, reflecting the natural progression of conceptual understanding of a domain. Another advantage of using standard ontology-based languages is that they offer easier integration of separately sourced models, since different ontological languages can be compared and integrated (since they are formal languages). This is important, as management systems themselves increasingly require more frequent integration of multi-vendor solutions in support of value chains. Therefore, as well as being used for modelling managed resources, ontologies can be used to model the services offered by management systems in support of their integration. Also, as ontology languages are general purpose in nature, there is more opportunity to integrate them with other models from other application domains, e.g. Geographic Information Systems. This becomes increasingly important as the management of networks and IT infrastructure become wedded to the management of application service, or, as addressed in this special issue, with the operation of pervasive computing systems.

2. ENGINEERING WITH ONTOLOGY BASED SEMANTICS

There are many definitions of ontology; a shortened version of the one we use for network management is as follows:

> An ontology is a formal, explicit specification of a shared, machine-readable vocabulary and meanings, in the form of various entities and relationships between them, to describe knowledge about the contents of one or more related subject domains throughout the life cycle of its existence.

The fundamental problem with engineering using ontologies is that ontologies represent knowledge in a fundamentally different way than established information and data models. The representation of knowledge is fundamental to the understanding of associated cognitive issues, such as what a concept means and what it implies. Even for a narrow domain, it is hard for network management users to agree on how to represent knowledge. This is because ontological engineering for network management is relatively new, and because people are used to current ways of managing devices (e.g., MIBs). In fact, the formal study of representation is much more demanding than it appears. Palmer defines five features of a representational system [3]:

- What the represented world is (what we perceive)
- What the representing world is (the theoretical structures used to formalize our perception)
- What aspects of the represented world are being modeled
- What aspects of the representing world are doing the modeling
- What are the correspondences between the two worlds

Most of these representational aspects have not been formally considered as people developed existing models. Thus, there is a semantic gap between using ontologies and using current management information.

As an example, consider two devices using different CLIs to provide a single end-to-end service. The object of the network administrator is to define a set of commands in each CLI that perform the same function (e.g., traffic conditioning). From a linguistics point-of-view, this is equivalent to first, defining a *shared* vocabulary and set of grammatical rules (we'll call this a "device Esperanto"), and second, defining a *translation* from each CLI to the common device Esperanto. This is a very difficult problem, as the translation requires understanding the structure and formation of the words in the vocabulary (morphology), the associated sentence structure (syntax), and the meaning of the sentence (semantics). This is impossible to do using UML, as UML cannot properly represent the linguistics of language [4]. Ontologies provide an answer using semantic similarity matching. Conceptually, this treats each command set as a graph, and finds the graph (i.e., set of commands) in one device that have the closest semantic content to a given graph for another device. This is used in the FOCALE autonomic networking architecture [1].

3. SEMANTIC INTEROPERABILITY

The promise of ontologies is in the sharing of an understanding of a domain that can be communicated between people and application systems. This promise has already been demonstrated through the conversion of various network management information models (such as DEN-ng, CIM and SMI) into OWL representations [5]. As well as enabling the interchange of management information in a heterogeneous environment, this common syntactic representation exposes clearly the different levels of semantic modeling that is embodied in existing models. The discovery and use of semantic mappings between the ontological versions of different management models is set therefore to become an increasingly important part of management system engineering. However, two challenges need to be overcome before ontology mapping can become a natural part of development in this domain.

First, fully automatic generation of mappings by applying matching algorithms to ontologies is generally considered impractical as yet. Thus, the challenge lies in tailoring the domain independent nature of most state of the art ontology mapping systems for management domain users. This consideration arises from the lack of certainty involved with the automatic matching process, primarily due to the heuristic nature of most matching algorithms. Up to now, mapping has primarily relied on a human user examining the matching information that has been generated, aided by a graphical user interface. However, as management models typically evolve slowly, there is the opportunity for mapping efforts of individuals or organizations to be productively reused. In ontology mapping systems such as COMA^{++} [6] and SWOOP [7], the focus is on providing graphical support for the presentation of matching information and point and click creation of mappings. In addition, COMA^{++} provides the opportunity for the user to browse mappings created by others. In other systems, such as Protégé, the user is led through decisions for mappings based on the iterative Anchor-PROMPT algorithm [8]. Recently, the idea of documenting ontology mapping patterns has been proposed [9], which supports domain-specific mapping decision making. However, for the management domain where the semantics of the different models are generally known or where expertise is readily available, a point and click ontology mapping system will be sufficient, as long as it is capable of importing and displaying mappings from elsewhere.

The second challenge lies in the format to be used for sharing and reusing mappings. Most state of the art mapping systems express mappings in a proprietary format that is typically aligned with the technology used by the mapping system. Increasingly, the need for an open mapping format is being recognized and proposals have begun to emerge [9]. For example, XML based formats to enable comparison of the output of a variety of matching tools were developed for international ontology mapping contests. Experience from these contests proved positive and led to the development of the INRIA ontology alignment format [10]. The format can also be rendered into different ontological forms (SWRL, OWL etc.) for the purposes of interpretation. In contrast, deBruijn et al. [9] have proposed a generic mapping language that must be grounded in a declarative logical language and thus requires a reasoner. Initial groundings to OWL (Description Logic-based language) and WSML-Flight (a Logic Programming-based language) have been developed. Unfortunately, it is too early to determine whether one of the two prominent contenders (that is from INRIA and from deBruijn et al.) will emerge as the basis of a standard format, whether another will be proposed or whether the common Rule Interchange Format (RIF) emerging from the W3C might be sufficiently expressive.

4. OUTLOOK

The costs of adopting ontologies in the engineering of management system reflect the classic bootstrap problem that faces the Semantic Web in general, which is that the benefits of using ontologies only bear fruit when there is a critical mass of models (or in web terms, content) with semantic mark-up. Someone therefore has to bear the costs of providing semantic markup of content, potentially well in advance of them realising the benefits. This markup involves developing ontological models for the wealth of management related models that exists, e.g. those documented in SMI, GDMO, IDL, XML, UML, DEN-ng, as well as the accompanying natural language descriptions. The problem is that the same information model can be used by multiple users and applications, each concentrating on a different part of the model. Hence, the semantic markups themselves will vary in depth and semantic content. Furthermore, there are a lot more people that are used to information models and data models than semantic processing, so it is likely that, the native models will remain the normative version for the foreseeable future. This creates a maintenance problem, in that different semantic content produced from the same model will need to be harmonised. Furthermore, as previously said, the semantic markup currently does not lend itself to full automation.

Therefore, the management domains likely to be early beneficiaries of ontology-based engineering are those where the level of innovation in services and networks is high, where services and networks can be freely integrated and therefore where the complexity is in the composition between COTS software rather than embedded in single vendor systems. Here, the benefits in selecting and integration separately sourced COTS software products could be quickly realized. The use of ontologies will, however, be less appealing initially where there is a large body of legacy models and complexity is monolithic. This can be characterized as knowledge domains where centralized resource control and commoditization are priorities We therefore expect that ontology-based management will appear first in domains such as Internet/mobile application services (e.g., Software as a Service), in lightly regulated wireless domains (e.g., MANETS, Mesh, and Cognitive Radio), and in Pervasive Computing. Ontologies will be less suited to engineering in the wired-backbones consisting of high-value network elements (e.g., in ISPs and 3G operators).

Effective engineering tools will be needed, that handle inconsistencies when integrating ontologies, incompleteness when building models and in browsing and searching large populations of models. Tools must include support for developing and maintaining mappings to non-ontological forms (e.g. UML) as well as support for developing semantic interoperability mappings between separately sourced models. The combination of these functions will be necessary to provide interoperability between different vendor-specific implementations using the various grounding models directly. Where policies form the management model of concern, tools should exploit ontologies to help explain reasons for policy conflicts and to reason over the model(s) being used in suggesting policy refinement paths. Existing knowledge engineering tools, such as Protégé, will not meet these requirement, as network management needs a stronger, more modular engineering platform which can more easily support the native management models (e.g., Eclipse). In general, we need to better understand when to move from general purpose ontological tools and reasoners to optimized but-functionally constrained versions (i.e., when to move from an innovation phase to a production/provision phase).

REFERENCES

1. J. Strassner, N. Agoulmine, and E. Lehtihet, *FOCALE—A Novel Autonomic Computing Architecture*, LAACS, 2006.
2. OWL Web Ontology Language Reference, available at: http://www.w3.org/TR/owl-ref/.
3. S. Palmer, Fundamental aspects of cognitive representation, In E. Rosch and B. B. Lloyd (eds), *Cognition and categorization*. Hillsdale, NJ: Lawrence Erlbaum Associates, 1978.
4. A. Wong, P. Ray, N. Parameswaran, and J. Strassner, Ontology mapping for the interoperability problem in network management, *J Selected Areas Commun*, Vol. 23, No 10, pp. 2058–2068, 2005.
5. J. E. López de Vergara, V. A. Villagrá, and J. 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)*, distributed videoconference, 11–13 June 2002.
6. D. Aumüller, H. Do, S. Massmann, and E. Rahm, Schema and ontology matching with COMA++, *Proceedings of the SIGMOD 2005 (Software Demonstration)*, Baltimore, June 2005.
7. A. Kalyanpur, E. Sirin, B. Parsia, and J. Hendler, Hypermedia inspired ontology engineering environment: Swoop, *Proceedings of 3rd International Semantic Web Conference (ISWC-2004)*, Japan (Poster).
8. N. Noy and M. Musen, Anchor-PROMPT: Using non-local context for semantic matching, *Proceedings of IJCAI workshop on Ontologies and Information Sharing*, pp. 63–70, 2001.
9. J. deBruijn, D. Foxvog, and K. Zimmerman, Ontology mediation patterns library, *IST SEKT Project Deliverable, 4.3.1*, Feb. 2005.
10. J. Euzenat, An API for ontology alignment (version 1.3), *Available at http://co4.inrialpes.fr/align/align.pdf*, June 2005. Last visited: Aug. 2005.

John Strassner is a Fellow of the Technical Staff and directs Motorola's autonomic computing research. He was a past Fellow of Cisco Systems and the Chief Strategy Officer of Intelliden. His research also includes policy management and knowledge engineering, including ontologies, machine learning and reasoning. He is an IEEE member, and was awarded the Daniel Stokesbury memorial award for excellence in network management. He is also an adjunct professor for the Waterford Institute of Technology in Waterford, Ireland.

He is a Distinguished Fellow of the TeleManagement Forum. He is the Chairman of the Autonomic Communications Forum, and also the Vice-Chair of WG6 (Reconfigurability and 233 Autonomics) in the WWRF. He has published two books and over 140 papers.

Short bio for **David Lewis** appears at the end of Guest Editorial. Short bio for **Declan O'Sullivan** appears at the end of the article "Ontological semantics for gathering and routing contextual knowledge in highly distributed autonomic systems" in this issue.