Building Ontology-Driven, Deployable Systems: An Engineering Perspective

Robert Dourandish
Quimba Software
4 West 4th Ave, Suite A
San Mateo, CA 94402
650-342-2337
bob@quimba.com

Nina Zumel
Quimba Software
4 West 4th Ave, Suite A
San Mateo, CA 94402

Michael Manno
Air Force Research Laboratory
26 Electronic Parkway
Rome, NY 13511

Ontologies represent a significant advance in knowledge-driven systems because they naturally bridge *descriptions* and *models* in a unique way that (a) can be designed by Subject Matter Experts (SME); and (b) can, when appropriately expressed, directly be used by computer programs,[1], [2]. The combination of the two capabilities promises the prospect of creating computing platforms where SMEs can quickly "solve" a problem – by creating an appropriate ontology, and having that onology be immediately deployed. This paper describes a multi-year project that focused on the challenges of delivering this capability as a production-grade environment. We first briefly review the project's goals and objectives before embarking on a discussion of what challenges, in our opinion, inhibit widespread, production-grade, commercial acceptance of ontologies.

Keywords: Ontology, Web Services, Systems Engineering. OWL, Protégé.

Introduction

For the past several years we have been researching massively scaled, multi-organizational, automated information sharing infrastructures with a focus on emergency response[3]. The project used ontologies in two distinct ways: in a *descriptive* mode where the ontology expresses the components of an entity such as a fire department; and as a *modeling* instrument to express concepts such as 9-11 call handling, bio-surveillance, information sharing process, and Health Information Portability and Accountability Act (HIPAA) that regulates data sharing in the health care context.

Lessons Learned and Recommendation

Throughout a three-year research project we have explored using Ontologies to allow Subject-Matter Experts (SME) to specify ontologies that would automatically be deployed to a production environment. Our research demonstrated that ontologies not only have the potential to replace software requirements, they also offer significant

---

[1] Cui Tao, Yihong Ding, and Deryle Lonsdale, "Automatic Creation of Web Services from Extraction Ontologies" In Proceedings of the First International Workshop on Semantic Web Applications: Theory and Practice *(SemWAT 2006) in conjunction with ER 2006*, LNCS 4231, Tucson, Arizona, 2006

[2] Dourandish, R., Zumel, N, Manno, M., "A Design Pattern for Automatic Generation of Web Services from Domain Ontologies", Proceedings, 3rd International Conference on Web Information Systems and Technologies (WEBIST), Barcelona, Spain, 2007.

[3] Dourandish, R., Zumel, N., Manno, M., "Automated Military-Civilian Information Sharing", MilitaryCommunications Conference (MILCOM), 2nd IEEE Workshop on Situation Management (SIMA),Washington, D.C., 2006.

advantages to traditional requirements engineering. Chief amongst these benefits is the fact that ontologies designed directly by SMEs can significantly reduce software engineering tasks, duration and cycle. Our research toward achieving this goal also revealed a number of deficiencies, discussed in this paper. Our vision is ubiquitous utilization of ontologies in production environments and, eventually, as core components of mission-critical platforms. Our goal here was to bring the most significant of these discoveries to the attention of the community with the hope of contributing to the advancement of ontologies. In that spirit, we put forth the following recommendations:

1. JDBC-like Ontology Reader
We venture to guess that everyone in the Protégé community is familiar with using JDBC to access relational databases. The interface is simple, well understood, and allows programs to use the same mechanism to access databases produced by different Data Base Management System (DBMS) vendors. We believe a similar approach will make knowledge bases significantly more accessible to engineering development and is the enabler to widespread commercial use.

2. Database Interface
We believe a tighter integration with relational databases will go a long way in making Protégé environment appeal to the software engineering community. Even without extensive integration, ability to query a database to instantiate the ontology, even in "assisted mode", will in our opinion be of tremendous value. The key to effectiveness here is to design the interface (or the API) for a *software* engineer, as opposed to a knowledge or database engineer.

3. Platform Consideration
Protégé forces the designer to explicitly import every single ontology – even those that are embedded in previously imported ontologies. This is inconsistent with either Java or C standards that (software) engineers are used to. It also has, as it turns out, a surprisingly high "annoyance" impact and, without a doubt, is an issue if one were to use Protégé for large-scale ontology design – particularly when dealing with distributed ontologies.

In our experience the use of predicate logic interface is counterproductive - particularly since it can be replaced with simple natural or artificial language phrases such as if-then-else.

3. Software Consideration
We believe ability to extend the ontology using in-line code and/or scripting will significantly enhance ontology flexibility as not all knowledge or processes can (easily) be expressed in a declarative, stateless manner. An interface to workflow engines is simply required to promote wide utilization.

ACKNOWLEDGEMENTS
This work was supported in part by a Small Business Innovation Research award from the US Air Force, under contract number FA8750-05-C-0085. The views and opinions presented in this paper represent the views of the authors and do not, necessarily, represent the views of the DOD.