MOQ: WEB SERVICES ONTOLOGIES FOR QOS AND GENERAL QUALITY EVALUATIONS

Henry M. Kim
Schulich School of Business, York University, 4700 Keele St., Toronto, Ontario Canada M3J 1P3; (416) 736-2100 x77952 [phone]; (416) 736-5687 [fax]
hkim@schulich.yorku.ca

Arijit Sengupta
Kelley School of Business, Indiana University, Bloomington, IN USA
asengupt@indiana.edu

Joerg Evermann
School of Information Management, Victoria University, Wellington, New Zealand
joerg.evermann@vuw.ac.nz

Abstract

When describing Web services, one of the obvious aspects that needs representing is Quality of Service" (QoS), the capability of a Web service to meet an acceptable level of service as per factors such as availability and accessibility. However too much of a focus on developing functional QoS ontologies has led to an over-emphasis on representing solely QoS metrics and units of measurement. For instance, what does round trip time actually mean? Is the round trip time of every data item measured? Is it an average, or is every nth item measured? Is it the actual time that is important or just the % of items that are beyond a certain range? Arguably existing QoS ontologies cannot readily answer many of these design questions because these questions have less to do with evaluating QoS and more to do with representing "what is quality?" Therefore, there is an unmet need for Web services ontologies that are designed at a higher level encompassing domain independent concepts, and generally applicable beyond QoS evaluations. The MOQ set of ontologies designed from the premise that quality is "conformance to requirements" aims to fill this need. Comprised of ontologies of requirement, measurement, traceability, and quality management systems, MOQ can be extended to encompass QoS metrics and measurement units or be designed to interoperate with existing QoS ontology. Either way MOQ use promises to ensure that ambiguity in QoS evaluations is minimized.

Keywords: semantic Web, ontologies, QoS, quality

Research: research paper

1 INTRODUCTION

There are many efforts such as OWL-S (Ontology Web Language—Services) (Martin et al., 2004) to enable ontology development for "semantic Web services." When describing Web services, one of the obvious aspects that needs representing is Quality of Service" (QoS), the capability of a Web service to meet an acceptable level of service as per factors such as availability, accessibility, integrity, performance, and reliability, regulatory compliance, and security (Mani & Nagarajan, 2002). Not surprisingly then, Web services and semantic Web standards and languages are meant to work with ontologies with which QoS requirements can be represented, and compliance to them, reasoned about (Bilgin & Singh).

In this paper, we argue that not only is there a need to develop QoS ontologies, but there is a concomitant need to think about QoS "ontologically"; that is, to study the ontology of QoS and quality from a bit more of a meta-physics level, and not solely develop yet another computer-based model of QoS that happens to be implemented in an AI ontology language. To that end, we present the framework for Mid-level Ontologies for Quality (MOQ), which extends the TOVE ontologies for quality management (Kim, Fox, & Grüninger, 1999), originally developed for enterprise modeling. These ontologies are deemed mid-level because their representations are less general than concepts found in SUMO (Suggested Upper Merged Ontology) (Pease & Niles, 2002). However MOQ also delves more into the meta-physical question of "what is quality?" more so than say the FIPA QoS ontology (FIPA, 2003).

This paper is organized as follows to justify MOQ development and describe MOQ. In §2, a review of relevant literature is presented. The review shows that MOQ fills a research need unmet by existing QoS and similar ontologies useable for Web services. In §3, a motivating scenario and competency questions for MOQ, its use case, is presented. In §4, the ontologies that comprise MOQ are outlined and some of the ontologies' representations are presented in first-order logic. Finally in §5, concluding remarks and future work are discussed.

2 LITERATURE REVIEW

There are several ontologies in literature that are explicitly called QoS ontologies: FIPA's (FIPA, 2003); MILO's (Teknowledge, 2004); METEOR-S's (Cardoso, Sheth, Miller, Arnold, & Kochut, to appear); and those by Zhou *et al.*(Zhou, Chia, & Lee, 2004) and Tien *et al.*(Tian, Gramm, Naumowicz, Ritter, & Schiller, 2003). These ontologies are specifically catered to evaluating QoS metrics such as bit error rate, and contain terms related to IT for Web services such as "logical name for transport protocol." Alternatively, a more general approach is to specify an ontology for Service Level Agreements (SLA's) (Ludwig, Keller, Dan, & King, 2002). Though QoS metrics would not be part of such an ontology, the ontology would facilitate specification of metrics including but not limited to QoS ones between the parties who agree on the SLA. This approach of representing more general contracts is more in line with the spirit of MOQ since a fundamental premise of TOVE is that quality is "conformance to requirements (Crosby, 2001)."

Tosic *et al.* (Tosic, Esfandiari, Pagurek, & Patel, 2002) look at the design of "formal representation (ontologies) of QoS and other constraints for Web services." They deem that necessary ontologies are those of QoS metrics, measurement units, currency units, measured properties, and measurement methods. They are clearly stating that measurement concepts need to be represented. However, when Web based measurement ontologies such as SHOE's (Heflin, 2000) and SEEK's (Bowers & Ludäscher, 2004) are examined, there is strong tendency towards representing physical metrics such as length and units such as 'cm' in the ontologies, and away from representing more about measurement properties and methods. On the other hand, the TOVE Measurement Ontology does indeed represent these oft unrepresented concepts, albeit it is not represented in a Web-based ontology language.

It can be concluded from the review that there is a need for a mid-level ontologies representing general quality concepts that can be used for Web services, among other applications, Moreover requirements and measurement are fundamental quality concepts that need to be ontologies in MOQ, Finally, MOQ should ultimately be implemented in OWL and be consistent with either a) direct representations of QoS metrics and measurement units into MOQ or b) inter-operating with existing QoS or measurement ontologies.

In the next section, we motivate the specific representations of the MOQ.

3 MOQ: MOTIVATING SCENARIO AND COMPETENCY QUESTIONS

In a well-known ontological engineering methodology (Uschold & Gruninger, 1996), the development of an ontology starts with a prose form description of a business situation for which an ontology based systems would be used. This description is called a motivating scenario. Next, the scenario is parsed to state higher level business questions that an ontology based system would be able to help answer. These are called competency questions. Below, the motivating scenario for a Web services application is presented as well as the competency questions it motivates.

3.1 Motivating Scenario

Putnam Lovell Securities Inc. is an investment bank, which consistently uses latest technologies including Web-based tools to deliver information to its employees and partners. They are using Web services technologies to integrate their business processes with those of two of their partners—Salesforce.com and BlueMatrix. The delivery of research reports to clients have been a manual, time-consuming process entailing database queries, building/editing mailing lists, and emailing reports. Their Web services based processes include service management features including configurable QoS, unified security model, and utilization and status reporting (IBM, 2004).

If in the future they wish to open up their processes to more ad hoc partners, they may need to explicitly represent business rules about these processes (and concomitant QoS expectations) in ontologies[1].

3.2 Competency Questions

With ad hoc partners, Putnam Lovell must be able to explicate and organize their partners' QoS requirements. So their ontology-based system must be able to help answer, "Is this a QoS requirement?" It must also be possible to answer, "Is this requirement satisfied?" If the QoS requirements are not satisfied, an investigation may take place. The system must serve some utility for this; i.e. it must help answer, "Within which Web service do problems arise?" The next step in diagnosis is asking, "Which software agents are responsible for that Web service?"

These questions characterize that fundamentally what constitutes quality—irrespective of technology used—must be known (requirements) and measured. When measurements point to a problem, the fundamental capability to resolve that problem is traceability. A quality management system—comprising in part of software agents in the Web services context—ensures that problems do not persist. The

[1] This paragraph extends an existing Web services scenario, assuming of course, that there will be a compelling business need for ontologies on top of Web services platform.

questions then characterize the four ontologies of MOQ: Requirements, Measurement, Traceability, and Quality Management.

4 MOQ

4.1 Requirements Ontology

Competency Question: Is this a QoS requirement?

The above question, stated formally in first-order predicate logic, is:

$$\mathrm{qos\_requirement}(Q). \tag{1}$$

Above, Q is a variable and represents the name or ID of a QoS requirement. A QoS requirement is a special type of a quality requirement, which in turn is a requirement.

$$X\ [\ \mathrm{qos\_requirement}(Q) \rightarrow \mathrm{quality\_requirement}(Q) \rightarrow \mathrm{requirement}(Q)\ ]. \tag{2}$$

There are some things that can be reasoned about a requirement just from its ID irrespective of its contents—e.g. its structure. Requirements are organized hierarchically. The predicate representing that a requirement Qi is a sub-requirement to a requirement Q is:

$$\mathrm{has\_requirement}(Q,Qi). \tag{3}$$

Assume that it must be explicitly stated that a requirement is a quality requirement However once stated, all its sub-requirements and their sub-requirements, etc. are also deemed quality requirements. Same holds for those quality requirements that are QoS requirements.

$$Q\ [\ \ Qi\ (\mathrm{has\_requirement}(Q,Qi)\ \varpi\ \mathrm{quality\_requirement}(Q) \rightarrow \mathrm{quality\_requirement}(Qi)]. \tag{4}$$

$$Q\ [\ \ Qi\ (\mathrm{has\_requirement}(Q,Qi)\ \varpi\ \mathrm{QoS\_requirement}(Q) \rightarrow \mathrm{QoS\_requirement}(Qi)]. \tag{5}$$

4.2 Measurement Ontology

A requirement without sub-requirements is primitive requirement. A requirement then either has sub-requirements or is a primitive requirement.

$$Q\ [5\ \ Qi\ \mathrm{has\_requirement}(Q,Qi) \rightarrow \mathrm{primitive\_requirement}(Q)\ ]. \tag{6}$$

$$Q\ [\ \mathrm{requirement}(Q) \leftrightarrow\ \ Qi\ \mathrm{has\_requirement}(Q,Qi)\ \omega\ \mathrm{primitive\_requirement}(Q)\ ]. \tag{7}$$

Whether a primitive requirement is satisfied should be directly measurable. A measurement can be identified by a uniquely identified resource unit Rt that is measured (e.g. an invoice data item), the attribute At that is measured (e.g. average round trip time), the actual measured value Mp, and a time point Tp at which measurement takes place.

measurement_pt(Rt,At,Mp,Tp). (8)

If the measurement point lies within an acceptable range, then that point is deemed a conformance point Q; if it does not, it is deemed a nonconformance point Q.

conformance_pt(Q,Rt,At,Tp). (9)

nonconformance_pt(Q,Rt,At,Tp). (10)

If Q is the ID of a primitive requirement and it is a conformance point, then Q is satisfied; if it is a nonconformance point, then it is not satisfied.

Q [Rt At Tp conformance_pt(Q,Rt,At,Tp) ϖ primitive_requirement(Q) → satisfied_requirement(Q)]. (11)

Q [Rt At Tp nonconformance_pt(Q,Rt,At,Tp) ϖ primitive_requirement(Q) → unsatisfied_ requirement(Q)]. (12)

Though it is possible that a requirement is satisfied if only some of its sub-requirements are satisfied, it is definitely true that a requirement is satisfied if *all* of its sub-requirements are satisfied[2].

Q [Qi (has_requirement(Q,Qi) → satisfied_ requirement(Qi)) → satisfied_ requirement(Q)]. (13)

Now the following competency question can be answered.

Competency Question: Is the QoS requirement satisfied?

QoS_requirement(Q) ϖ satisfied_requirement(Q). (14)

Similar ontologies in literature can be used to answer this question. However as pointed out in the literature review, they do not represent all of the following measurement concepts of a measured attribute At, which MOQ does:

has_sample_sizing(At,Sz). (15)

- Is a sample from a population measured? Is the sample size 1 or >1?

has_sampling_plan(At,Sp). (16)

- Is the sampling plan variable or attribute sampling?

has_standard_value(At,Sv). (17)

[2] The ontology does not directly represent any other type of requirements satisfaction. However this could be explicitly represented outside of the ontology.

- what is the standard value, i.e. the attribute's μ?

has_specification_set(At,Ssl). (18)

- What are the tolerance specifications, or specification set?

has_unit_of_measurement(At,U) (19)

4.3 Traceability Ontology

Competency Question: Within which Web service do problems arise?

This question states that a trace to a problematic Web services application procedure or routine is required. In the same vein, a trace to a problematic data item may also be necessary. Generally, the former is an activity, and the latter, a resource. Activities can be comprised of other activities, unless it is a primitive activity; resources can be comprised of other resources—e.g. an invoice is comprised or sub-forms or a lot is comprised of different batches—unless it is a traceable resource unit, a collection in which its constituents are not uniquely identified whereas one collection is uniquely identified as different from another. Tracing can be done to the level of a problematic primitive activity or traceable resource unit, but not to a level more microscopic. Some predicates of this ontology are:

primitive_activity(A). (20)

tru(Rt). (21)

- a tru is a traceable resource unit

If the series of activities and how they input and output resources are mapped at the level of primitive activities and traceable resource units, then tracing from downstream activities to upstream activities or final product to raw materials is possible as composition of primitive traces. Predicates that show this are:

prim_activity_trace(Rt,A_2,A_1,L). (22)

- a tru Rt is an output from primitive activity A_1, which is input to primitive activity A_2;
L is a path

prim_tru_trace(A,Rt_2,Rt_1,L). (23)

- primitive activity A takes tru Rt1 as an input and outputs tru Rt2; L is a path

activity_trace(A_2,A_1,L) (24)

- L is the trace path from a downstream primitive activity A_2 to an upstream primitive activity A_1.

tru_trace(Rt_2,Rt_1,L) (25)

- L is the trace path from a downstream tru Rt_2 to an upstream tru Rt_1.

4.4 Quality Management System (QMS) Ontology

Competency Question: Which software agent is responsible for that Web service?

In this ontology, organizational artifacts that characterize a QMS are represented. Hence the ontology includes definitions for terms like quality policy, evidence, and manager. A definition of the ontology is that a quality manager—could be a person when modeling a factory but can also be a software agent when modeling Web services—applies quality policy to primitive activities and tru's to provide evidence of quality. A quality policy is akin to a quality requirement on a quality manager, and quality evidence is akin to a measurement point on tru's and primitive activities controlled by the quality manager.

5 CONCLUDING REMARKS AND FUTURE WORK

It is evident that most of the semantic Web ontologies useful for assessing QoS are very implementation-focused. Such ontologies represent germane QoS metrics like "round trip time" and may even be closely coupled with Web services routines that enable effective assessment. However, what happens when we ask what is meant by round trip time? Is the round trip time of every data item measured? Is it an average, or is every nth item measured? Is it the actual time that is important or just the % of items that are beyond a certain range? In fact, is there a nominal desired value for that time? Is there a range of values that are acceptable? And what if the round trip times are not acceptable? Does the ontology provide terms that can be used to find the source of the problem or to alert those who can do something about fixing the problem? Arguably existing QoS ontologies and related ontologies like OWL or SHOE based measurement ontologies cannot readily answer many of these design questions. Why not? Because they have less to do with evaluating QoS and more to do with representing "what is quality?" In order to use QoS metrics to say evaluate metrics for semi-automated business processes that in part but not wholly use Web services, and in order to pre-empt possible misunderstandings between Web service provider and requester, there is clearly value in basing QoS ontologies in ontologies of more general quality concepts or enabling inter-operation between these sets of ontologies. We believe that MOQ based on the well-vetted TOVE ontologies for quality management can provide this value and fill a need in semantic Web services research.

There are many future work venues that we are actively following. One, we are comparing and evaluating different ontologies' representations using the popular BWW (Bunge-Wand-Weber) ontological analysis technique (Wand, Storey, & Weber, 1999) previously used to evaluate modeling languages. By applying this technique to MOQ measurement ontology, we will more systematically demonstrate the situations wherein use of MOQ is the most appropriate and how. Two, we are developing a prototype showing how MOQ can be used with specialized QOS ontologies and SOAP and WSDL enabled Web services application. Both works serve as systematic demonstrations—in lieu of a proof but more credible than arguments presented in a conceptual paper—of the stated value of MOQ.

6 REFERENCES

Bilgin, A. S., & Singh, M. P. (2004, July 6-9). *A DAML-based repository for QoS-aware semantic web service selection.* Paper presented at the IEEE International Conference on Web Services, San Diego, CA.

Bowers, S., & Ludäscher, B. (2004). *An Ontology-Driven Framework for Data Transformation in Scientific Workflows.* Paper presented at the International Workshop on Data Integration in the Life Sciences (DILS'04), Leipzig, Germany.

Cardoso, J., Sheth, A. P., Miller, J. A., Arnold, J., & Kochut, K. J. (to appear). Modeling Quality of Service for Workflows and Web Service Processes. *Web Semantics Journal*.

Crosby, P. B. (2001). *Quality is Free: The Art of Making Quality Certain*. New York, MY: McGraw-Hill.

FIPA. (2003, December 3, 2002). *FIPA Quality of Service Ontology Specification*. FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS. Retrieved August 15, 2004, from the World Wide Web: http://www.fipa.org/specs/fipa00094/SC00094A.html

Heflin, J. (2000, April 3, 2000). *Measurement Ontology 1.0 (draft)*. UMD. Retrieved August 15, 2004, from the World Wide Web: http://www.cs.umd.edu/projects/plus/SHOE/onts/measure1.0.html

IBM. (2004, June 20, 2004). *IBM e-business: jStart Program: Case studies: Web services - Grand Central Networks, Inc.* Retrieved August 9, 2004, from the World Wide Web: http://www-306.ibm.com/software/ebusiness/jstart/casestudies/grandcentral.shtml

Kim, H. M., Fox, M. S., & Grüninger, M. (1999). An Ontology for Quality Management: Enabling Quality Problem Identification and Tracing. *BT Technology Journal, 17*(4), 131-139.

Ludwig, H., Keller, A., Dan, A., & King, R. P. (2002). *A Service Level Agreement Language for Dynamic Electric Services.* Paper presented at the 4th IEEE International Workshop on Advanced Issues of E Commerce and Web Based Information Systems (WECWIS 2002), Los Alamitos, CA.

Mani, A., & Nagarajan, A. (2002, January 1). *Understanding quality of service for Web services*. IBM. Retrieved August 15, 2004, from the World Wide Web: http://www-106.ibm.com/developerworks/library/ws-quality.html

Martin, D., Paolucci, M., McIlraith, S., Burstein, M., McDermott, D., McGuinness, D., Parsia, B., Payne, T., Sabou, M., Solanki, M., Srinivasan, N., & Sycara, K. (2004, July 6-9). *Bringing Semantics to Web Services: The OWL-S Approach.* Paper presented at the The First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004), San Diego, CA.

Pease, A., & Niles, I. (2002). IEEE Standard Upper Ontology: A Progress Report. *Knowledge Engineering Review, 17*, 65-70.

Teknowledge. (2004, January 6, 2004). *MILO*. tecknowledge.com. Retrieved August 15, 2004, from the World Wide Web: http://einstein.teknowledge.com:8080/download/register.jsp?fileType=.zip&fileName=Milo.zip

Tian, M., Gramm, A., Naumowicz, T., Ritter, H., & Schiller, J. (2003, December). *A Concept for QoS Integration in Web Services.* Paper presented at the 1st Web Services Quality Workshop (WQW 2003), Rome, Italy.

Tosic, V., Esfandiari, B., Pagurek, B., & Patel, K. (2002). On Requirements for Ontologies in Management of Web Services. *Lecture Notes in Computer Science, 2512*, 237-247.

Uschold, M., & Gruninger, M. (1996). Ontologies: Principles, Methods and Applications. *Knowledge Engineering Review, 11*(2), 93-136.

Wand, Y., Storey, V. C., & Weber, R. (1999). An Ontological Analysis of the Relationship Construct in Conceptual Modeling. *ACM Transactions on Database Systems, 24*(4), 494-528.

Zhou, C., Chia, L.-T., & Lee, B.-S. (2004, July 6-9). *DAML-QoS ontology for web services.* Paper presented at the IEEE International Conference on Web Services, San Diego, CA.