ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2013

Addressing the Contract Issue, Standardisation for QOS

Dr.R.Udayakumar

Associate Professor, Department of IT, Bharath University, Chennai, TN, India

ABSTRACT: Higher level service support mechanisms are an integral part of the future vision for Web / Grid Services. This paper argues that the areas of discovery, differentiation, negotiation, monitoring and non-repudiation of agreements cannot be considered in isolation to each other. The areas outlined above are examined, primarily from a trust perspective, focusing on the use of contracts to guarantee QoS attributes. The paper explores the need for greater standardisation in the area, to specify semantics more clearly; in doing so we outline the progress of our own QoSOnt QoS attribute specification ontology. The paper goes on to briefly discuss two tools; firstly, SQRM, designed to allow service discovery, querying and requirement specification utilising the QoSOnt ontology; and secondly, TRANSACT, an existing contract negotiation tool designed to provide end-to-end contracting, encompassing an automated negotiation engine.

1. INTRODUCTION

The ongoing adoption and commercialisation of Web and Grid Service Oriented Architectures (SOA) has increased the pressure to develop new high level service support. With Web and Grid Service architectures converging, it is becoming clear that it is the higher level functionality that now requires addressing. Service negotiations currently generalise to one of two situations:

Free Operation

Free operations are common during the development of new tools and services, but are suitable only in business situations encompassing a single administrative / organisational boundary (where QoS guarantees are less of an issue). Given the trend towards distributed computing through VO (Virtual Organisation) development [1] this model is untenable for widespread use.

Economically supported services

A more feasible business model but brings with it additional concerns. An economic model requires attention to higher level service provision, including support for discovery, negotiation, monitoring etc. These are covered in more depth in section 2.

In particular this paper focuses on the support mechanisms required in the provision of an economic service framework, made up of clients and providers, exchanging services for financial gain.

Figure 1 illustrates the higher level service provision necessary to support economic Web / Grid service operation. However, in order for these mechanisms to operate, shared terminology and use of language is required, something that is currently lacking.

The areas shown above are not trivial to provide solutions to, nor can they be addressed in isolation. However, the areas of law and banking are seen as external for the purposes of our research, as they represent systems already in place.

The remainder of the paper is as follows: Section 2 examines the objectives for research in the area. Section 3 provides an overview of the methodology for the work. Section 4 provides implementation details on our research. Section 5 provides a brief overview of the technologies involved in the research area. Section 6 provides information on our evaluation methods. Section 7 provides a summary of the business benefits to research in the area. Finally, section 8 concludes with a summary of the achievements made and the scope for future work.

II. OBJECTIVES

The main objective in any contracting domain is the establishment of trust between parties. The following points examine the area in further depth through each stage of the service operation cycle:

- Service discovery

Service discovery mechanisms such as UDDI [2], UDDIe, WSIL, etc are rudimentary in that the information stored is often not semantically rich enough for service differentiation. A number of issues exist with centralised service discovery mechanisms like UDDI [3]; it would be a mistake to automatically consider a UDDI registry an unbiased, trustworthy third party. Current public UDDI registries contain much that is out of date, un-vetted and inconsistent. Private UDDI registries like E2Open Process Directory (E2PD) [4] can avoid this but at the expense of public availability.

- Service negotiation

  Negotiation consists of the determination of a compromise position between a client's requirement specification, and a service's capability specification. Existing research in this area has often concentrated on too small a part of the problem, for example only considering structure [5]. There are some limited commercial products emerging such as eMediator, but more research in the area is needed**.**

- Service Agreement

  The signing of documents to guarantee service level attributes. This could involve a contract / SLA. For clarification a contract is a legally binding document; an SLA is not legally binding, unless linked to a contract. SLAs tend to store more quantitive data, in the form of metrics, values, ranges etc.

- Service mediation

  Mediation requires the specification of standardised complaint and renegotiation policies. Common methods include levelled commitment contracting [6], which place a penalty on either changes to existing contracts or the collapse of a contract.

- Service monitoring

  The monitoring of service use, based on data provided by a combination of client, service and possibly trusted third parties. The trust angle to service monitoring is substantial [7]. The trust relationships can be broken down roughly thus:

  - Trust of service to provide statistics accurately
  - Trust of third party to remain unbiased
  - Trust of client to supply statistics accurately
  - Trust in the associated contract, and its legal backing

  This paper does not aim to solve the monitoring issue directly, but instead uses it to illustrate the problem that incorrect understanding / ambiguity can cause in the monitoring process, and to reinforce the need for greater standardisation.

III. METHODOLOGY

The areas outlined in section 2 all rely upon standardised semantics for information provided and exchanged through the contracting process. In order to address these problems standardised forms of QoS attributes / metrics are required. We believe these can most accurately be stored in the form of a QoS ontology capable of linking different metrics, units and attributes in order to control usage.

In software engineering, an ontology can be defined as "*a specification of a* conceptualization" [8]. More precisely, an ontology is an explicit formal specification of how to represent the objects, concepts, and other entities that exist in some area of interest and the relationships that hold among them. In general, in order to be useful, an ontology must represent a shared, agreed upon conceptualisation. The use of ontologies in computing has gained popularity in recent years for two main reasons:

1. They facilitate interoperability.
2. They facilitate machine reasoning.

In its simplest form an ontology is simply a taxonomy of domain terms. However, taxonomies by themselves are of little use in machine reasoning. The term ontology also implies the modeling of domain rules. It is these which provide an extra level of machine "understanding".

Only once terms have been standardised does it make sense to automate the service negotiation / procurement process. In doing so, the remaining bottleneck to efficient dynamic utilisation of resources based on a VO model can be removed. Given this advance, it is then easier to design higher level support tools to aid in service discovery, capable of providing feedback and of assisting in service differentiation for the user. Standardisation of this type also allows more advanced negotiation engines to be created, to automate parts of the negotiation process, and to allow contract construction from pre-written standardised clauses.

IV. DEVELOPMENTS

The following sections provide overview information of the ontology we have developed, along with two tools designed to provide proof of concept for the research completed in the area.

a. Quality of Service Ontology (QoSOnt)

QoSOnt [8] was developed by a process of examining existing QoS specification languages [9]. The main focus for the ontology so far has been in the area of dependability, where we have built upon existing work, by making use of an existing taxonomy [10] and modelling commonly used metrics. Our approach has been to provide a base set of useful constructs which cover common cases. These also exist as an example to others who wish to model their own QoS viewpoint on top of the basic QoSOnt Classes. To facilitate reusability and extensibility the ontology has been designed to be modular in nature. Each "module" is effectively an ontology in itself. Figure 2 illustrates the way in which the ontology is constructed, and provides an idea of the relationships held therein.

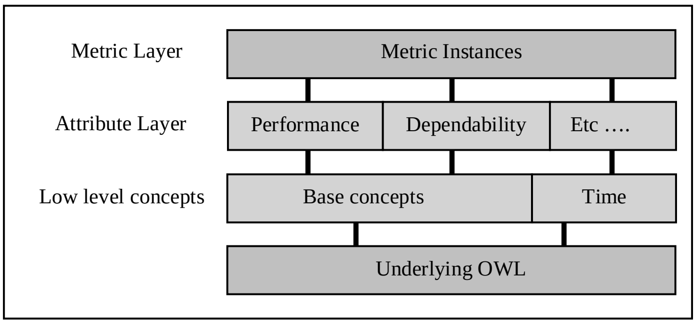


Figure 2: QoSOnt Stack

Figure 2 is, in part, an oversimplification of the structure, as the underlying OWL structures are in fact used throughout the stack. It is however useful to show the separation between general dependability concepts and specific metrics, stored separately.

We use the term attribute to refer to a general QoS property (e.g. dependability, reliability, performance), and metric to refer to a specific way of measuring an attribute. Metric, Attribute and other basic QoS concepts are defined in the base ontology. Concepts specific to some attribute can then be built into separate ontologies on top of the base concepts. We choose not to define specific metrics here as we do not wish to tie the generic concepts of dependability to specific ways of measuring dependability. We therefore have a separate ontology of actual metrics.

A Metric represents one way of measuring a specific QoSAttribute. It must result in a numerical value and must be calculable in practice as well as theory. For instance, a statement that a service has transactional throughput of 1000 transactions per second can be falsified by a single party (be they a client, provider or monitoring service) but cannot generally be measured by a client or third party, as they have no access to the traffic statistics for the service. A Metric is defined to consist of a description, an acceptability direction and zero or more values. The acceptability direction indicates whether higher or lower values are preferable for the Metric (e.g. A low probability of failure on demand is more desirable). It must be remembered that these classes can be extended or constrained by their subclasses, so being over-specific at this base level is undesirable.

4.2 Service QoS Requirements Matcher (SQRM)

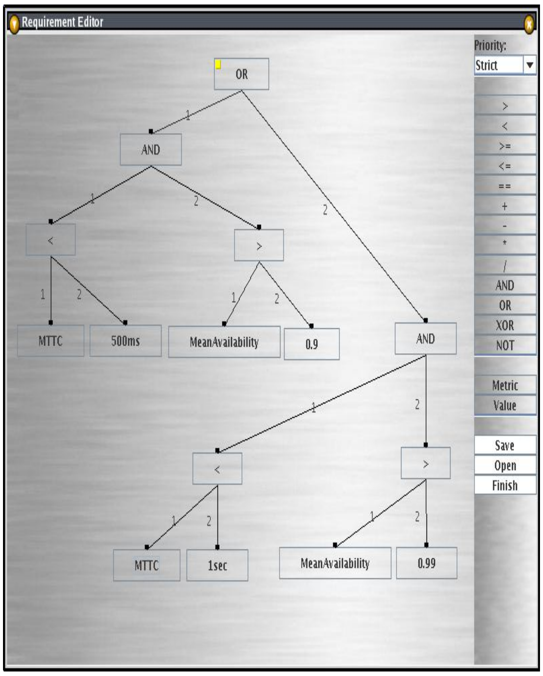
To demonstrate the use of the ontology, and aid in its evaluation, a tool for service discovery, differentiation and selection based upon QoS requirements has been developed. SQRM is designed to showcase a range of different situations in which QoSOnt could be utilised within the service domain. The tool supports the following client process:

- Service Discovery

  SQRM uses UDDI registries for service discovery. We do not attempt to address the existing problems with public UDDI registries (un-vetted / incorrect information etc), but instead use UDDI purely as one way to identify services worth further investigation. We implement this functionality using JAXR [12] which is registry independent – so theoretically would find it easy to implement support for other discovery mechanisms. Clients query the repository via keyword search. The extent to which clients can search for particular service requirements at this stage is highly restricted, this instead occurring at the next stage.

- Requirement Specification & differentiation

  QoS requirement and capability specification affects all clients and services. Without a way to specify requirements a client could not differentiate between services; without capability specification a service could not advertise its resources. The SQRM tool currently concentrates on the client viewpoint – providing a graphical means of specifying QoS requirements. The interface is point-and-click allowing the user to easily build complex statements to use in service differentiation an example of which is shown in figure 3.



Priority: Strict

Figure 3: SQRM Screenshot

4.3 Tool for Real-time Automated Negotiation of Secure Authorisation ContracTs (TRANSACT)

The TRANSACT [13] negotiation tool is aimed at grid development support with an emphasis on the negotiation of e-Science computational resources. This particular focus was chosen purely due to the existing demand for services of this type; the negotiation engine itself is generic, in that it could be used to negotiate the service agreements when the VO style of web services start to emerge in mainstream business.

The negotiating engine for both client and service sides of a given business interaction elicit requirements from their parties through a variety of mechanisms ranging from natural language statements, to metrics, ranges, etc. Different strategies for negotiation can then be employed to allow both sides to attempt to reach an appropriate compromise position, without further aid from the parties involved through a request-reply cycle. At the end of the process contract signing, exchange and storage are handled.

TRANSACT shows that, given a solution to the issue of ambiguity inherent in non-standardised views of phrases, metrics and values between parties, semi-automated tools can provide a complete contract support environment encompassing service discovery, differentiation, negotiation and contract security.

Interestingly, it is the combination of these primitive higher level contract support services that best illustrate the need for a QoS Ontology. An ontology is not merely a taxonomy of terms, but also a set of rules to govern the relationship between different components. It is this ability to translate between, for example seconds and hours or kilograms and pounds, that allow flexible automated tools to be developed across organisational / cultural boundaries.

V. TECHNOLOGY DESCRIPTION

The architecture of the QoSOnt QoS specification ontology has been aimed directly at the e-Science community, utilising OWL for ontological structuring. OWL was chosen due to its relative maturity, and the growing number of tools designed to aid in the development of ontological structures; in particular the Protégé [14] visual development environment, which allows complex ontological structures to be created alongside useful visualisation tools to aid in the understanding and analysis of developed structures. OWL is based on the popular RDF format, which in turn was originally based on XML.

Both the SQRM and TRANSACT tools were both based on Java utilising XML for querying, and contract creation. The design of the contracts themselves is based on open standards, using XML for structure, SOAP for transmission and PKI (currently X.509 [15]) for security and non-repudiation purposes.

VI. RESULTS

a. *Lessons learned*

OWL currently has limited support for XML datatypes. In order to address this we have encoded custom data types, in order to support concepts such as probabilities, percentages etc. However, in doing so it becomes difficult to get publicly available reasoning tools to process the OWL files correctly. The ability to encode more complex information, such as the way in which metrics compose, could also prove awkward with OWLs current capabilities. It is hoped that further development of OWL will remedy these issues in the near future. In the mean time however, we have had to work around many of these issues through the inclusion of custom XML inside the requirement specification and provider capability files.

b. Approach Evaluation

The evaluation of an ontology such as QoSOnt ultimately relies upon its application by the research community. We see QoSOnt as something which may, in the future, form the basis of a standard QoS ontology for use across the business / research community. During development, we have simulated its usage by generating a set of scenarios. The scenarios stem mainly from the e-Science community, which is currently the best source of potentially expensive long running services.

One scenario we are using is based upon the field of epidemiology, and the study of pandemics. The computation of the projected spread of diseases on given population models is both time consuming and of interest to multiple bodies, governmental, academic and independent. Requirements relating to different algorithms / processing capacity / time expended, make QoS specification an important factor. For example, some algorithms work better with larger datasets; others may converge on an answer in such a way as to make long processing runs unnecessary for the accuracy required; for others, short runs may render results useless. Information of this type can be built into an ontology, creating a richer information resource than a mere list of supported functions.

The SQRM tool has been written as proof of concept for the Ontology, to allow us to examine the requirements of the scenarios for flexible data querying and specification, and is gradually being integrated with the existing TRANSACT negotiation architecture.

We accept that we cannot anticipate all future service development requirements through the use of real world scenarios. We are therefore seeking to collaborate with real world service users in order to further evaluate and improve QoSOnt.

VII.BUSINESS BENEFITS

Without standardisation for contract / SLA terms, higher level service support will remain a confusing tangle of proprietary formats slowing further development in the area. Although standardisation efforts are underway for a variety of domains by organisations including UN CEFACT [16] and RosettaNet [17], it is the standardisation of QoS attributes that will be of most importance within the service negotiation domain, proving a precursor to further development.

VIII. CONCLUSIONS

In conclusion this paper examines the area of contract use in Web / Grid services. In doing so it identifies the challenges still to be faced, along with preliminary results from the QoSOnt contract ontology, and two higher level tools, SQRM and TRANSACT.

Future work in this area will be aimed at the extension of the QoSOnt ontology and further population of its existing classes. It is also hoped that the SQRM and TRANSACT tools could be brought closer together to provide a complete contractual service cycle tool.

ACKNOWLEDGEMENTS

I would like to thank the UK Engineering and Physical Sciences Research Council, grant number GR/M52786 and the Dependability Interdisciplinary Research Collaboration (DIRC).

REFERENCES

[1] Katzy R. Bernhard D. Design and Implementation of Virtual Organizations. In proceedings of the 31st Hawaii International Conference on System Sciences. http://portal.cetim.org/file/1/67/Katzy-1998-Design-and-Implementation-of-Virtual-Organizations.pdf

[2] Tom Bellwood et al, "UDDI Version 3.0.2", edited by Luc Clement et al, http://uddi.org/pubs/uddi-v3.0.2-20041019.htm

[3] Xu D et al. QoS-Aware Discovery ofWide-Area Distributed Services. University of Illinois. In proceedings of CCGrid 2001. www.cs.purdue.edu/homes/dxu/pubs/CCGrid01.pdf

[4] E2PD. Commercial website. http://www.e2open.com/

[5] Czajkowski K, Foster I, Kesselman C, Volker S, Tuecke S. SNAP: A protocol for negotiating service level agreements and co-ordinating resource management in distributed systems. *In the proceedings of GGF5*. http://www.isi.edu/~karlcz/papers/snap-lncs-25370153.pdf

[6] Anderson M. Levelled commitment contracting among myopic individually rational agents. Journal of Economic Dynamics and Control, 2001 http://citeseer.ist.psu.edu/66923.html

[7] Samani M. Monitoring of Distributed Systems. PhD Thesis. Imperial College of Science, Technology and Medicine 1995.

[8] T. R. Gruber, "A translation approach to portable ontologies", *Knowledge Acquisition*, Vol.5, No. 2, pp199-220, http://ksl-web.stanford.edu/KSL_Abstracts/KSL-92-71.html, 1993

[9] Dobson G, Lock R, Sommerville I. QoSOnt A QoS Ontology for Service-Centric Systems. Submitted EuroMicro2005

[10] Glen Dobson, "Quality of Service in Service-Oriented Architectures", http://digs.sourceforge.net/papers/qos.pdf

[11] Jean-Claude-Laprie, Brian Randell, Carl Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing", in IEEE Transactions on Dependable & Secure Computing. Vol. 1, No. 1, pp. 11-33.

[12] Sun Microsystems, "JAXR", http://java.sun.com/xml/jaxr/index.jsp

[13] Lock R. TRANSACT PhD Thesis. Lancaster University Jan 2005