Utilising the Universal Description, Discovery and Integration registry to provide business performance and quality of service data for web services

Mark Fox

Faculty of ACES

Sheffield Hallam University

Abstract

Web services can be defined as 'a family of interrelated standards that work together to provide a simple way to allow program functionality in different languages and on different platforms to interoperate' (Mattern and Woods, 2006, p. 13). Interoperability is therefore the key to the technology as it can provide an organisation with the ability to communicate and perform business tasks with other business partners regardless of language or platform.

Fischer and Werner in Abecker, Grimm and Struder (2007) describe the issue of interoperability as that of implementing effective integration between business partners. The effectiveness of this integration can be measured not only in terms of the performance of the service itself but also in its appropriateness for the consumer to achieve quality of service (QoS). At present, a consumer can query the Universal Description, Discovery and Integration (UDDI) registry looking for a desirable service and acquire the Web Service Definition Language (WSDL) that enables interoperability.

This paper looks to review the impact of performance indicators and quality of service with regards to web service selection for consumers. I combine the ideas of three published papers to suggest a model that improves the capacity of performance indicators together with the effectiveness of service discovery by utilising the UDDI registry itself. I will review each paper to note their contribution before presenting my own model and conclude with recommendations for future work in the field.

Performance monitoring and web services

The framework developed by McGregor and Schiefer (2003) separates the performance data of the web service from either the service provider or the service consumer. By externally operating and storing the data outside of the Service Oriented Architecture (SOA) triangle, processing power and storage is not taken away from systems used to execute the business processes themselves. The benefit of this is a reduction in the complexity of the original systems as they are effectively reducing functionality and drag on processing and storage operations.

However, the primary function of the framework proposed by McGregor and Schiefer (2003) is that a Solution Management Service be implemented that utilises web service concepts as part of a distributed system in that its capabilities are accessible to all involved business partners. Three solution management web service type categories of *define*, *log* and *analyse* are first registered with the service registry, possibly the UDDI, which allows all parties to interact with the solution management web service architecture. Its aim is to 'support a near real-time integration of audit trail data from the executed processes' (McGregor and Schiefer, 2003, p.406) and reduce the time taken for requests to be served once they have arrived. This would provide an almost constant stream of performance data to decision makers to aid business processes and thus the 'time it takes to make the business decisions' (McGregor and Schiefer, 2003, p.406).

All of this lends to the web service based solution management framework providing a better return of investment for the organisation, the aim of which is to establish a competitive advantage in the marketplace by innovative use of a decision support system (DSS). It is also beneficial and of interest to both business partners as it essentially supports a mutual desire to increase efficiency of service operation.

There are however difficulties with this approach. Aggregated web services would be more difficult to monitor and would have to serve multiple business partners. There is also the issue of who would fund the costs of running an external solutions management service and the ever increasing storage space required for such data. It could be argued that as the service provider is providing the service at a cost in the first place, then responsibility would lie with them. However, where aggregated web services are used across a number of service providers, which of them would therefore absorb the costs? Additionally, security becomes an issue as service providers might be unwilling to allow other service providers access to their performance data that could be used against them.

Finally, the issue of monitoring business performance data itself needs to be addressed. The suggested solutions management service registers the *define*, *log* and *analyse* categories with a service registry however, many organisations employ very company-specific performance indicators to cater to their own needs. Any attempt for a web service to apply their own performance indicators may mean an organisation is not completely in control of what they can monitor and analyse.

Performance indicators to quality of service indicators

In their paper on monitoring business performance in web service compositions, Abeck, Gebhart and Momm (2009) note a number of initial problems with the configuration of current monitoring tools. Firstly, such specific configuration reduces scalability and results in solutions that are not portable (cost and complexity issues inhibits movement to alternative tools). Secondly, a focus of effort during design time can lead to solutions that are not robust and lead to erroneous modelling activities due to changing business priorities. Finally, as each performance indicator has to be defined in its entirety regardless of often similar calculation rules, redundancy and complexity for the developer becomes an issue.

To solve these problems they use a 'template based mechanism for defining performance indicators' (Abeck, Gebhart and Momm, 2009, p. 343) which basically reuses the calculation patterns or functions from existing solutions that have been fully defined to create a monitoring model. This approach addresses the first problem of scalability as generic monitoring models produced using the templates are portable to any monitoring tool or framework and thus are platform independent. The second problem is addressed as the templates reduce complexity and takes the focus away from intensive designing. Changing business priorities can then be tackled more smoothly and errors are less prominent. The last problem is also addressed by the reduction in redundancy and complexity yielded from templates, as the developer does not have to rely on cut and paste to recreate similar parts of calculation rules.

Another advantage of the proposed template based solution is that if a monitoring model is platform independent, it can be used concurrently with web services in a SOA environment. That is, there are no issues of interoperability and can thus operate outside of the constraints of a platform dependant application. The benefits of this would be the increased ease at which performance indicators can be universally understood by multiple business partners and supports the external operation and storage of a solution manager service.

Abeck, Gebhart and Momm test their template based solution with a basic real-life scenario of a watch manufacturer collecting fitness data via a fitness watch then combining it with a health check service to produce performance results. The test proved that templates were can initially work effectively and also be highly reusable. What is interesting about this is that QoS parameters can also be calculated in the same way using templates. This receives limited mentioned in the paper confined only to discussion of related work, but the potential for it still remains.

Further to this, it would be easier to implement template based performance indicators (and possibly QoS indicators) in near real-time within a web service framework which can supply an almost constant stream of data. The scalability of the templates and the shorter space of time specific indicators can be written also increases the desirability of template usage.

Again, the overall aim of a template based solution would be to reduce development costs and produce faster responses to changing business priorities. Business decisions can be made quicker and a service can be implemented in less time for a consumer making the business more attractive. It could also make it easier to calculate QoS data to attract consumers however this was not the intended focus of Abeck, Gebhart and Momm in any great detail.

Quality of service in web services

In reference to web services and service oriented architecture (SOA), QoS is defined as ‘all runtime processing aspects that go beyond the surface interface’ (Margolis, 2007, p.18). For Margolis this refers to reliability, security, service coordination and runtime updates, all of which can be calculated in some shape or form. From the perspective of a service provider, QoS is a strong selling point for their product and an opportunity to attract consumers. From the perspective of a consumer, QoS is therefore an indication of a level of service which can be expected and forms criteria from which appropriate services can be chosen.

What Mingjing et al (2009) propose is that even though QoS describes an organisations web service capability, the widely used UDDI registry doesn't provide QoS information. In other words, consumers cant ‘search criteria based on QoS parameters...and compare between services before service invocation’ (Mingjing et al, 2009, p.353).

The UDDI is an open source initiative and enables service providers to register their details. However, as Werner and Fischer (2007) also note, consumers can only search for desirable services based on three criteria: basic business information (white pages), business type (yellow pages) and web service access information (green pages). The UDDI does not provide any other search options and so organisations can not obtain concrete measure of quality before invoking the web service itself.

To tackle this problem, Mingjing et al (2009) suggest that not only is QoS data externally stored in a similar way to McGregor and Schiefer (2003), but that it is employed as a 'QoS based web service selection model (WSSM-Q) to provide QoS support for service publishing and selection' (Mingjing et al, 2009, p.353). Mingjing et al call this a QoS Broker which provides near real-time QoS support to all service providers and consumers, not just existing business partners as McGregor and Schiefer (2003) suggest.

The benefits of a brokered approach, aside from those already discussed, would be an increase in the quality of available services in general as published QoS data promotes a more efficient industry and ensures higher standards. Consumers are also benefiting from the reduced time to locate more efficient services in ranking order vital to the success of their businesses.

A brokered approach does however have trade-offs for both the provider and the consumer. These are not due to any technical constraints but more so as a business model. The brokers could just as easily include the same search criteria as the UDDI rendering it useless. At the same time, a broker is itself offering a service and thus is entitled to charge for the services they provide whereas although the UDDI is limited, it at least is free to register and query.

Businesses would be split in their decision to either pay extra or spend longer searching for an efficient service. This would of course be dependant on the amount that they charge but they could end up in a position where brokers control the market, not something which was envisioned as an open source initiative to promote standards. Also, brokers could argue a charge from providers for effectively advertising their business. Providers would have to pay, not so much as to get a competitive advantage, but to ensure that they remain in competition themselves.

Proposed model

This paper has discussed the benefits of a web service based solutions manager service displaying performance indicators (McGregor and Schiefer, 2003), template based performance calculators (Abeck, Gebhart and Momm, 2009) and web service QoS brokers (Mingjing et al, 2009). Each one has their merits and can contribute to improvements in web service usage however, certain elements from them can be combined to produce the same intended results without the same draw-backs and trade-offs.

An external web service that operates and analyses business performance data could use a template based calculation system to improve speed and accuracy. This improved system could also provide QoS data using template based calculations but make them available as searchable criteria in the same way a broker would. However, instead of operating as a web service based QoS/business performance broker, I suggest including it not as additional to the UDDI, but as part it.

The UDDI would essentially be expanding their role, possibly adding another 'page' to the three components that already exist to include both QoS and business performance data open to consumer query. If the UDDI was implemented as a web service in itself (instead of essentially client/server), there would still remain the problem of who would fund and maintain this expanded service. It would also take considerable effort to transform the UDDI into such a format that supports the functionality of near real-time data display.

To get round this particular problem I propose adopting a decentralised peer-to-peer (P2P) framework that undertakes all tasks of the expanded UDDI for providers and consumers to locate each other. A network protocol could be developed as an open source initiative by industry standards representatives and provide a solution to the use of brokers. A network client could also be initially developed in the same way that provides network access for participating nodes (peers acting as client and server). Each provider would act as a network node and publish their own performance and QoS data on the network via their own servers making this extremely cheap and scalable and accessible to everyone on the network. Consumers would also act as a node as they join the network looking for services.

Of course, a likely scenario would be that some businesses would use the network more than others and so peers would not be entirely equal in their network participation, however a 'tit-for-tat' protocol could be developed that would ensure fair use. That is, organisations that have a higher network usage therefore foot more of the processing power required for the search functionality. To this end super-nodes would be required but on a fair use policy. Figure 1 shows a possible P2P network topology which would support a collective sharing of QoS and performance data as a discovery service. Nodes are represented by the smaller circles and super-nodes are represented by the larger circles. The lines represent the peer connections and the routes of the search requests.

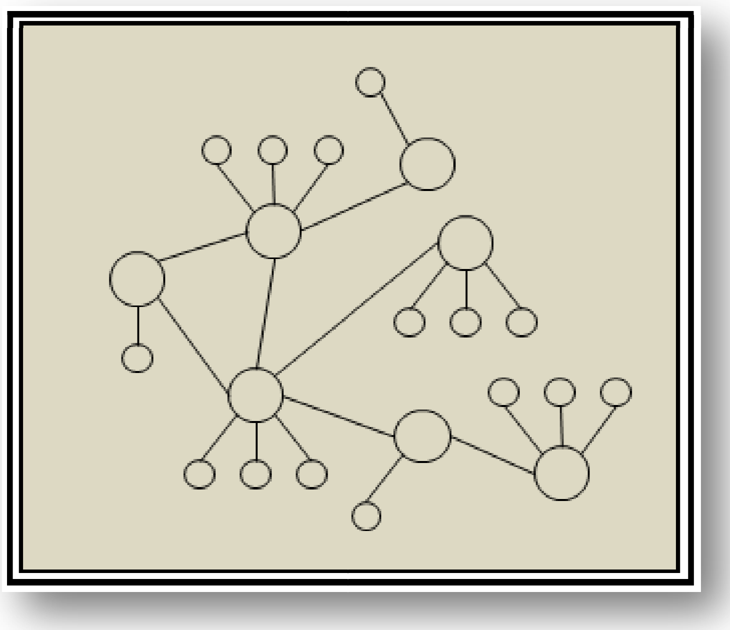

Figure 1: Network Topology

Figure 2 shows the relationship of the UDDI P2P network to the Consumer and provider. Firstly, the provider publishes the QoS and performance data on the network, then the consumer queries the network based on template calculation criteria. The network then routes the search request through the different nodes on the network (as in figure 1) and returns the required data and provider details to the consumer. Consumers would then be in a good position to make informed business decisions based on various provider data.

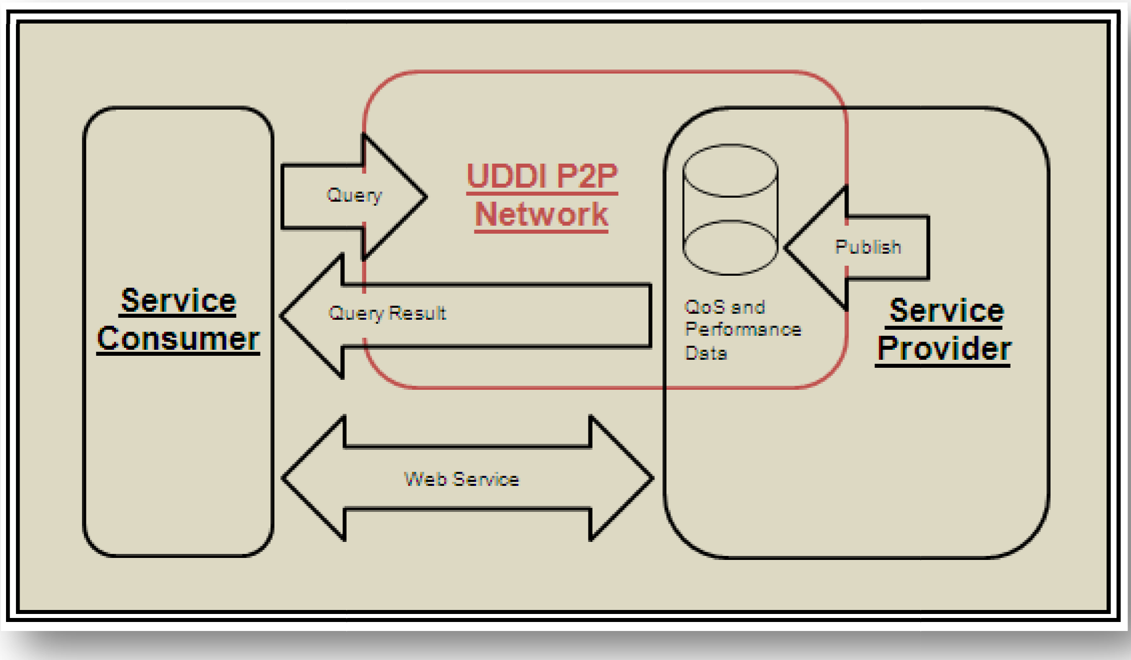

Figure 2: P2P Service Discovery

Conclusion and future work

The overall aim of this model is to maintain an effective system of locating web services that is not reliant on a third party and the costs associated with it. This would hopefully have the effect of encouraging the development of more web services of better quality but still retaining a good return on investment by keeping costs down.

The model itself is still very much in its basic form and would require development to further verify its potential. Future work could firstly benefit from the development of a protocol and network client that would support such a network. Secondly, it would also benefit from research and analysis centred on search requests in a decentralised P2P network as this would be the main functionality of the network itself, the efficiency of which could possibly be the deciding factor for provider and consumer network participation.

Lastly, the model would benefit from future work on security in a P2P environment. As each node is accessible to all others in the network, an anonymity function would have to be put in place to protect the business interests of its users. Also, business performance data intended for an organisations internal use only, would have to be identified and prevented from network distribution. Failing security could compromise the market position of an entire organisation and so security would have to be developed to a high standard to reduce risks to an acceptable level.

References

ABECK, Sebastian, GEBHART, Michael and MOMM, Christof (2009). *A Model Driven Approach for Monitoring Business Performance in Web Service Compositions,* in Internet and Web Applications and Services, 2009, ICIW '09, Fourth International Conference on 24th - 28th May 2009, p. 343 - 350.

ABECKER, Andreas, GRIMM, Stephan and STRUDER, Rudi (eds.) (2007). *Semantic Web Services: Concepts, Technologies and Applications*. Berlin, Springer.

MARGOLIS, Ben, (2007). *SOA for the Business Developer: Concepts, BPEL and SCA.* Lewisville, MC Press.

MATTERN, Thomas and WOODS, Dan (2006). *Enterprise SOA: Designing IT for Business Innovation*. Cambridge, O'Reilly.

McGREGOR, Carolyn and SCHIEFER, Josef (2003). *A Framework for Analysing and Measuring Business Performance in Web Services*, in E-Commerce, 2003, CEC, 2003, IEEE International Conference on 24th - 27th June 2003, p. 405 - 412.

MINJING, Lan et al (2009). *A QoS Based Web Service Model*, in Information Technology and Applications, 2009, IFITA '09, International Forum on 15th - 17th May 2009, p. 353 - 356.

YU, Liyang (2007). *Introduction to the Semantic Web and Semantic Web Services*. London, Chapman & Hall.