International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 15 (2018) pp. 12104-12110
© Research India Publications. http://www.ripublication.com

Review of SOA Technologies and Versioning Methods

Leenus M[1], Senthil Kumar S[2]

[1]*Assistant Professor, Department of Computer Science, Periyar Govt. Arts College, Cuddalore, India.*
[2] *Research Scholar, Manonmaniam Sundaranar University, Tirunelveli, India.*

Abstract

The architectural approach in the development of distributed applications has resulted in the growth of Service Oriented Architectures (SOA). A service-oriented architecture is essentially a collection of services. These collective services communicate with each other which involve either by simple data passing or multiple services coordinating a goal specific activity. Services have to be connected with each other based upon their requirements.

The aim of this paper is to provide an overview of the SOA techniques and their Versioning techniques used in service oriented design, development and also provide a context for a deeper understanding of services and service-oriented architectures for enterprise-scale software solutions. In specific, it explores services in relationship to the more established concept of software components.

Keywords: Service Oriented Architecture, Web Service, Versioning.

INTRODUCTION

The emergence of Web services and its developments and standards in support of automatic business integration has driven major technological advances within the integration software space, most notably, the service oriented architecture (SOA) ([1, 3]). The aim of this architecture is to handle the requirements of loosely coupled, standards based mapping enterprise information systems (EIS) and protocol independent distributed computing. In an SOA, software package resources are a unit prepackaged as “services”, which are well defined, self contained modules that give standard business functionality and are independent of the state or context of other services. Services are in a standard definition language, have a published interface, and communicate with each other request execution of their operations in order to collectively support a common business task or process [2]. Services that use Web services standards like Web Services Description Language (WSDL), Universal Description, Discovery and Integration registry (UDDI) and Simple Object Access Protocol (SOAP), are the most desired type of services available now. In this Review article, we survey the methodologies, technologies and approaches of SOA that enables business integration projects and deliver a flexible and adaptable environment. All functions in an SOA are termed as services [4]. This includes pure business transactions, system service functions and business functions composed of lower level functions as well. All services are independent and their operations are utilized by external components. Clarity of Service ensures that the external service seekers and providers perform the required function. The application and discharge space of the application providing the desired functionality is encapsulated behind the service interface. In most of the cases the interface between the services are communicable. Irrespective of whether services are internal or external, the interconnect schemes are classified in order to enable the infra-structure components to establish the connection between them.

This survey shows the concepts, developments and principles in the area of middleware integration brokers, Application integration, SOA, adapters and event-driven computing. It also explains how these components operate as part of emerging distributed computing techniques. This paper also extracts the basic concepts related to SOA versioning from our expertise and the content of related literature survey and uses these concepts to develop a set of related guidelines and practices that are summarized throughout the text as guidelines. These guidelines, practices and recommendations highlights the issues that come from using specific SOA techniques as well as discusses what stakeholders must address when developing service-oriented systems.

SERVICE-ORIENTED ARCHITECTURE

Service-oriented architecture is not a new notion; it is very important at this time because of the increase in the need for Web services technology. As one of the books noted that the value of a service-oriented architecture as “Service-Oriented Solutions and Applications must be developed as independent sets of interacting services offering well-defined interfaces to their potential users. Similarly, supporting technology must be available to allow application developers to browse collections of services, select those of interest, and assemble them to create the desired functionality.”

For the purposes of this document, we shall consider the following definition of a service: A service is generally implemented as a course-grained, discoverable software entity that exists as a single instance and interacts with applications and other services through a loosely coupled (often asynchronous), message-based communication model.

In many ways, the terminology for services is much the same as the terminology used to describe component-based development; however, there are specific terms used to define elements within Web services, as shown in figure 1. The relationship between the service providers and clients is shown in figure 2.

SURVEY ON EXISTING METHODOLOGIES OF SOA

IBM Service-Oriented Analysis and Design (SOAD): SOAD proposes elements that should be part of a design methodology and service oriented analysis [5]. SOAD builds upon existing techniques, such as OOAD, CBD, and BPM. It also introduces SOA-specific techniques, such as service categorization, aggregation and service conceptualization, policies and aspects, meet-in-the-middle process, semantic brokering, and service harvesting.

IBM Service Oriented Modeling and Architecture (SOMA): SOMA modeling methodology by IBM recommends the following steps: components realizing services identification, specification, and realization of services flows. The activity is highly iterative and incremental [6]. However, because SOMA is proprietary to IBM, its full specification is not available. It has been recently announced that the Rational Unified Process has been associated with SOMA to execute results in what is called IBM RUP for SOMA [7].

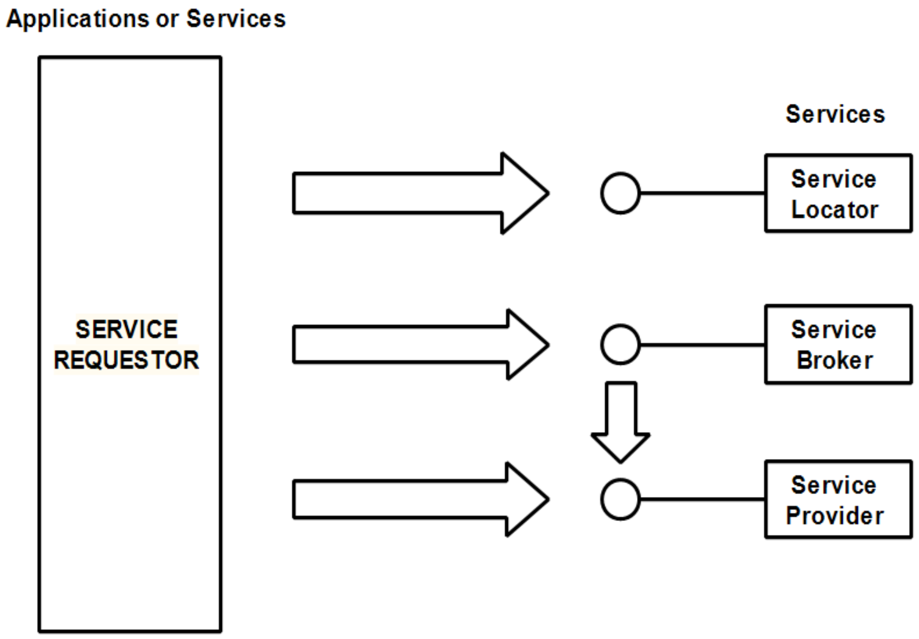

Figure 1: Service Terminology

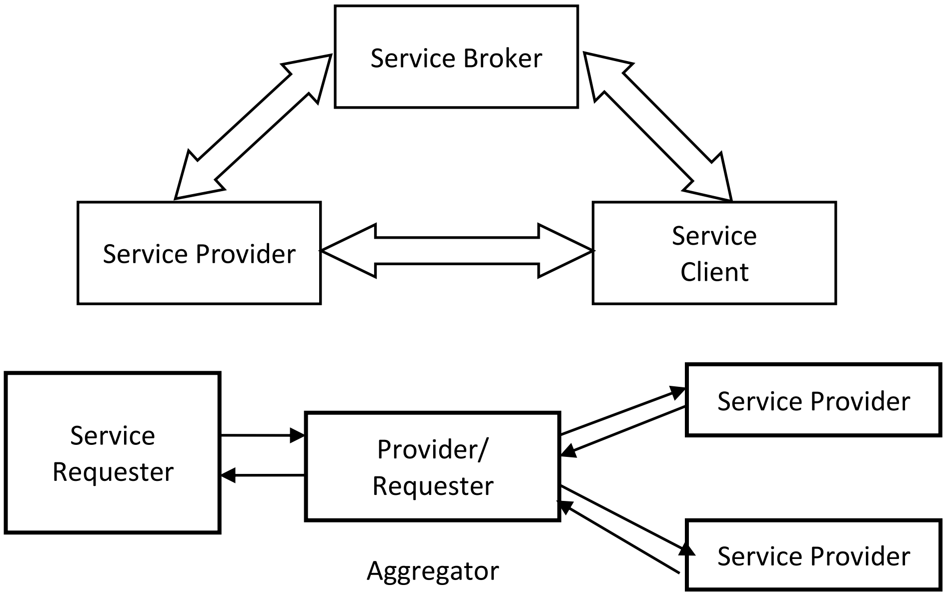

Figure 2. Implemented services

SOA Repeatable Quality (RQ): SOA RQ is a property methodology by Sun Microsystems that is based on a RUP like iterative and incremental process consisting of following five phases: elaboration, transition, conception, inception and construction[8]. UML compliant artefacts are applied for filing various deliverables of these phases.

CBDI-SAE Process: The four key discipline areas described in CBDI process are: enable, provide, manage, and consume [9]. Each area groups with same disciplines that are further multiplied down to process units and then to tasks. The CBDI is a SOA methodology as part of its CBDI SAE SOA Reference Framework (RF).This methodology aims business IT integration through top down analysis of business requirements as well as down up legacy system integration. The CBDI SAE process aims to cover the whole SOA lifecycle, including deployment, monitoring, and governance activities.

Service Oriented Architecture Framework (SOAF) [10]: SOAF consists of five main phases: roadmap and planning information elicitation, service identification, service definition, service realization, and service implementation. It is concurrently based on two types of modeling activities: “Tobe” modeling, which is the top down business oriented approach describing the required business processes, and “Asis”.

Service Oriented Unified Process (SOUP)[11]: This approach defined by K. Mittal is based on the Rational Unified Process [RUP]. Its lifecycle consists of six phases: define, design, incept, construct, support and deploy. SOUP lacks detailed documentation and leaves room for adaptation. It is used in different variations: one adopting RUP for initial SOA projects and the other adopting a mix of XP and RUP for the maintenance of existing SOA rollouts.

Methodology by Papazoglou et al [12]: The methodology uses an iterative and incremental process that comprises one preparatory and eight distinct main phases. Papazoglou et al has proposed and tested a service development methodology from the point of view of both consumers and providers, which attempts to cover the full SOA lifecycle. It is based on well established development models, such as the CBD, RUP and BPM.

Thomas Erl’s : This methodology [13] is a step by step guide through two main phases namely : design and analysis. This service oriented design and analysis methodology is considered as the first vendor friendly one to be published. The activities in the design analysis phase take a top down business view where service candidates are identified. These serve as data for the next phase of service oriented design in which the service candidates are mentioned in detail and later known as Web services.

BPMN to BPEL[14]: The authors coined the term business process oriented programming to refer to an evolutionary step in software engineering where programming power is handed over to the business analyst. In this approach the business process is expressed in an abstract model and according to transformation rules it is automatically mapped to an execution language that can be executed by a process engine.

Steve Jones’ Service Architectures[15]: The aim of this top down methodology consists of the first steps in a project necessary to ensure that true SOA properties are satisfied in the last delivery. It is technology sceptic and takes a top down business view reaching up to the point of service candidate discovery. The methodology adopts a four step process: Why, How, What and Who, of which the first three are covered in preparation for the fourth step.

COMPARISON OF THE LISTED METHODOLOGIES ACCORDING TO THE IDENTIFIED CHARACTERISTICS IS SUMMARIZED IN TABLE 1:

	IBMSOAD	**IBMSOMA**	**SOARQ**	**CBDI-SAE**	**SOAF**
Delivery Strategy	M	M	M	M	M
Life cycle coverage	A&D	A&D	Complete	Complete	A&D and planning next Phase
Prescriptive	1	4	3	4	3
Proprietary	YES	YES	YES	NO	NO
Agile	n/a	3	4	2	2
Existing Process	NO	RUP(Recents)	RUP	?	NO
Existing Techniques	OOAD,BPM	?	?	?	NO
UML	YES	?	YES	?	?
Applied in Industry	YES	extensively	extensively	Not Yet	A Case Study
Consumer View	YES	YES	YES	YES	YES
Provider View	YES	YES	YES	YES	YES

	SOUP	**Papaz.[12]**	**Erl's[13]**	**BPMN to BPEL[14]**	**Jones 'SA[15]**
Delivery Strategy	M	M	T	T	T
Life cycle coverage	Complete	Complete	A&D	A&D Impl.	A&D and planning next Phase
Prescriptive	1	2	4	2	1
Proprietary	NO	NO	NO	NO	NO
Agile	5	3	1	n/a	n/a
Existing Process	NO	RUP(Recents)	RUP	?	NO
Existing Techniques	RUP,XP	RUP	NO	NO	NO
UML	NO	NO	NO	NO	NO
Applied in Industry	NOT YET	NOT YET	NOT YET	NOT YET	NOT YET
Consumer View	YES	YES	YES	YES	YES
Provider View	YES	YES	YES	NO	NO

Table 1.Comparison of SOA development methodologies. (A relative quantitative scale 1-5 is used for some criteria.Also, M=Meet in the Middle, T=Top-Down, B-UP, and ? =No Data)

VERSIONING IN SOA

In service oriented system growth, versioning is a key process for indicating the impact of change on service consumers, Versioning enables in minimizing maintenance costs, and improving the overall management of services [17, 18, 16]. Moreover, software versioning tends to be much more complex when dealing with service-oriented systems, mainly because of the distributed nature of SOA development[17]. It is common for different groups, from either the same organization or different organizations, to develop service-oriented systems collaboratively[19]. Hence it is necessary for the developers to address several software versioning methods that are specific to the distributed development paradigm. The first challenge of versioning service oriented systems is the absence of centralized control. When software is developed in a distributed manner across a number of organizations, the control is that of a sole distributed management group, which usually is distributed within multiple organizations. Hence it becomes necessary that developers must adapt versioning processes that require centralized control to account for coordination and communication among different groups that access the architecture and services.

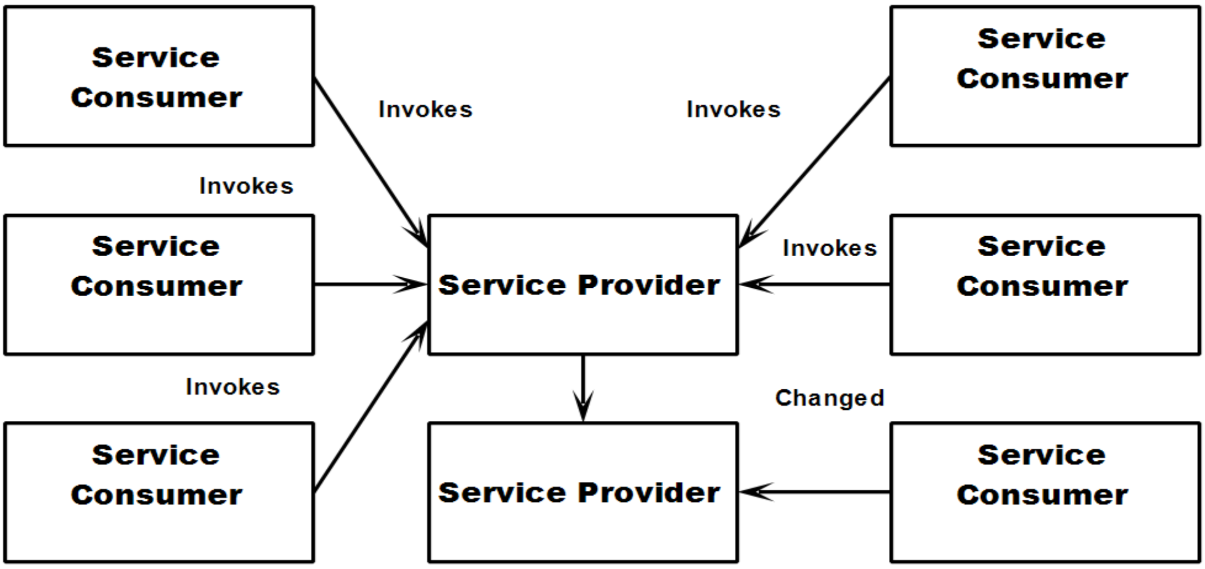

Figure 3: Multiple service versions

RESEARCH ISSUES IN VERSIONING IN SOA:

Diephouse [19] examined specific patterns using SOAP namespaces and other artifacts for versioning web services. In [20], a peer to peer based framework is investigated that allows to advertise and find services using keyword-based search, ontology based search and behavior-based search in a highly decentralized and dynamic environment. In addition, the framework provides mechanisms so that users may express and query the quality of services.

In [21], a dynamic monitoring approach is analyzed that is capable of specifying monitoring rules governing the control of WS BPEL processes. Klein et al. [22] introduced the concept of partially instantiated service descriptions containing different types of variables which are instantiated successively, thereby mirroring step by step progress in a trading process. In [23], the authors describe how an OWL reasoner can be integrated with an AI planner to overcome the problem of closed world semantics of planners versus open world semantics of Web Ontology Language (OWL).

There is a large amount of existing versioning guidance for service oriented systems. However, much of this guidance focuses on specific standards, technologies, and implementation models, such as "WS" standards for web services. For example, Brown and Ellis discuss how change type theory relates to services and demonstrate the concept using WSDL and UDDI registries [24]. Lee also discusses on versioning[25].Lublinsky explores the differences among versioning, QoS, and encoding policies as well as using standards such as WSDL 2.0, WS Addressing, and WS-Policy for implementing these policies [26]. Juric and Sasa [27] extend the BPEL language to include the management of versioned web services. This is based on an exploration of WSDL and UDDI that was originally presented by Juric and Sasa [28]. The same authors expanded their study to web-service interfaces and presented the resulting BPEL extensions [29]. Poulin provides a detailed study of key artifacts to version and contributes a naming scheme for them [30]. Several researchers also describe custom SOA infrastructures that are built specifically to support versioning. Leitner and colleagues present a limited discussion of versioning concepts, including change types and service proxies that make versioning decisions at runtime[31]. They also investigate transparency as a key attribute and propose a custom versioning solution. Similarly, Fang and colleagues offer a brief review of versioning concepts and a custom versioning solution that they implement on both the service provider and the service consumer [32]. While implementing or controlling the versioning of web services, it is also necessary to adhere to the policy efforts of standards organizations related to The World Wide Web Consortium (W3C). The W3C provides the beginnings of a semantically annotated WSDL document called SAWSDL, which would give explicit versioning capabilities to WSDL documents [33]. OASIS attempts to address the versioning problem in their WSDM MOWS document, although it is more of an exploratory paper [34]. Some works that have been discussed includes the topic of versioning under the heading of a larger topic, such as SOA governance.

CONCLUSION

In this paper, we have surveyed research technologies, approaches and issues related to services and their underlying technologies and services architectures. Particularly, we have reviewed, application servers including integration brokers, business process management, and several Service Oriented Architecture technologies that implement the backbone of an Enterprise Service Bus, which is of critical importance to develop the service oriented computing paradigm operational in a business context.

We have also provided guidance on the typical issues that must be addressed while choosing a versioning policy for a service oriented system. We have given specific guidance on implementing and supporting versioning capabilities in technologies that are commonly used for building services. Finally, we highlighted common problems that can occur in a poorly versioned service-oriented system to reinforce the need for comprehensive versioning policies. We encourage researchers to use these to build comprehensive versioning policies tailored to their system context. This technology can also be used in WS4D (Web Service for Devices) which are the latest implementation in Web Services.

REFERENCES

[1] Booth, D., et al.: Web Service Architecture. http://www.w3. Org/tr/wsarch/, W3C, Working Notes, 2003/2004.

[2] Fremantle, P., Weerawarana, S., Khalaf, R.: Enterprise services. Commun. ACM 45(10), (2002).

[3] Burbeck, S.: The tao of e-business services: the evolution of Web applications into service-oriented components with Web services. IBM DeveloperWorks, 2000. http://www-106. ibm.com/developerworks/Webservices/library/wstao/

[4] Arsanjani, A.: Introduction to the special issue on developing and integrating enterprise components and services. Commun. ACM 45(10), 30–34 (2002).

[5] O.Zimmermann et al, “Elements of Service-Oriented Analysis and Design”, IBM Corporation, available from http://www-128.ibm.com/developerworks/library/ wssoad1/, June 2004.

[6] A. Arsanjani, “Service-oriented modeling and architecture”, IBM Corporation, available from http://www.128.ibm.com/developerworks/webservices/ library/ws-soa-design1/, November 2004.

[7] Ali Arsanjani, “IBM’s SOA method: SOMA, Service Oriented Modeling and Architecture”, IBM Corporation, available from http://www-03.ibm.com/developerworks/blogs/page/AliArsanjani?entry= soma service oriented modeling and&ca=drs-bl, December 2006.

[8] SUN Microsystems, “SOA RQ methodology - A pragmatic approach”, available from

http://www.sun.com/products/soa/soa methodology.pdf.

[9] P. Allen, “The service oriented process”, in CBDi Journal, February 2007, http://www.cbdiforum.com/reportsummary.php3?page=/secure/interact/ 2007-02/service oriented process.php&area=silver.

[10] A,Erradi et al, “SOAF: An architectural framework for service definition and re- alization”, in Proceedings of the IEEE International Conference on Services Com- puting, pp 151-158, Chicago, USA, September 2006.

[11] K. Mittal, “Service Oriented Unified Process (SOUP)”, available from http:// www.kunalmittal.com/html/soup.shtml, 2006.

[12] M. P. Papazoglou et al, “Roadmap of Service Oriented Computing”, March 2006, available from http://infolab.uvt.nl/pub/papazogloump-2006-96.pdf.

[13] T. Erl, Service-Oriented Architecture: Concepts, Technology, and Design, Upper Saddle River: Prentice Hall PTR, 2005.

[14] C. Emig et al, “Development of SOA-based software systems - and evolutionary programming approach”, in Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Ser- vices, p 182, Guadeloupe, French Caribbean, February 2006.

[15] S. Jones, “A Methodology for Service Architectures”, Capgemini UK plc, available from http://www.oasis-open.org/committees /download.php/15071/ A%20methodology%20for %20Service%20Architectures%201%202%204%20-% 20OASIS%20 Contribution.pdf, August 2005.

[16] Peltz, C. & Anagol-Subbarao, A. “Design Strategies for Web Services Versioning.” SOA World Magazine (April 2004). http://soa.sys-con.com/node/44356.

[17] Laskey, K. “Considerations for SOA Versioning,” 333–337. 2008 12th Enterprise Distributed Object Computing Conference Workshops. Munich, Germany, Sep. 2008. IEEE Computer Society Press, 2008.

[18] Evdemon, J. “Principles of Service Design: Service Versioning.” MSDN. http://msdn.microsoft.com/en-us/library/ms954726.aspx (2005).

[19] Diephouse, D. XFire User Guide, Versioning. http://docs.codehaus.org/display/XFIRE/Versioning (2007).

[20] Sahin, O.D., et al.: SPiDeR: P2P-Based Web Service Discovery In: Proceedings of the Third International Conference on Service Oriented Computing, pp. 157–170. Springer, Amsterdam (2005) .

[21] Baresi, L., Guinea, S.: Towards dynamic monitoring of WS-BPEL processes. In: Proceedings of the Third International Conference on Service Oriented Computing, pp. 269–282. Springer, Amsterdam (2005).

[22] Klein, M., KonigRies, B., Obreiter, P.: Stepwise refinable service descriptions: Adapting DAML-S to staged service trading. In: Proceedings of 1st International Conference on Service Oriented Computing, December 2003.

[23] Sirin, E., Parsia, B.: Planning for Semantic Web Services. In: Martin, D., Lara, R., Yamaguchi, T. (eds.) Proceedings of the ISWC 2004 Workshop on Semantic Web Services: Preparing to Meet the World of Business Applications, 2004.

[24] Brown, K. & Ellis, M. “Best Practices for Web Services Versioning.” developerWorks. http://www-128.ibm.com/developerworks/webservices/library/ws-version (2004).

[25] Lee,J.“Architecting Industry Standards for Service Orientation.” MSDN. http://msdn.microsoft.com/en-us/library/ms978270.aspx (2005).

[26] Lublinsky, B. “Supporting Policies in Service-Oriented Architecture.” Developer Works. http://www.ibm.com/developerworks/webservices/library/ws-support-soa (2004).

[27] Juric, M. B. & Sasa, A. “Version Management of BPEL Processes in SOA,” 146–147. Proceedings of the 2010 6th World Congress on Services (SERVICES '10). Miami, FL, July 2010. IEEE Computer Society Press, 2010.

[28] Juric, M. B., Sasa, A., & Rozman, I. “WS-BPEL Extensions for Versioning.” Information and Software Technology 51, 8 (August 2009): 1261–1274.

[29] Juric, M. B., Sasa, A., Brumen, B., & Rozman, I. “WSDL and UDDI Extensions for Version Support in Web Services.” Journal of Systems and Software 82, 8 (August 2009): 1326–1343.

[30] Poulin, M. “Service Versioning for SOA.” SOA World Magazine (July 2006). http://soa.syscon.com/node/250503.

[31] Leitner, P., Michlmayr, A., Rosenberg, F., & Dustdar, S. “End-to-End Versioning Support for Web Services,” 59–66. Proceedings of the 2008 IEEE International Conference on Services Computing, Vol. 1 (SCC '08). Honolulu, HI, July 2008. IEEE Computer Society Press, 2008.

[32] Fang, R., Chen, Y., Fong, L., Lam, L., Frank, D., Vignola, C., & Du, N. “A Version-Aware Approach for Web Service Client Application Integrated Network Management,” 401–409. IM '07: 10th IFIP/IEEE International Symposium on Integrated

Network Management. Munich, Germany, May 2007. IEEE Computer Society Press, 2007.

[33] World Wide Web Consortium. Semantic Annotations for WSDL and XML Schema. http://www.w3.org/TR/sawsdl (2007).

[34] OASIS. Web Services Distributed Management: Management of Web Services (WSDM-MOWS) (Draft). http://www.oasis-open.org/committees/ download.php/5664/wd-wsdm-mowsversioning (2004).

[35] Ludwig, H., Keller, A., Dan, A., King, R. P., & Franck, R. Web Service Level Agreement (WSLA) Language Specification. IBM Corporation, 2003. http://www.research.ibm.com/wsla/WSLASpecV1-20030128.pdf.

[36] Schepers, T. G. J., Iacob, M. E., & Van Eck, P. A. T. “A Lifecycle Approach to SOA Governance,” 1055–1061. Proceedings of the 2008 ACM Symposium on Applied Computing (SAC '08). Fortaleza, Ceará, Brazil, Mar. 2008. ACM, 2008.