



Provided by the author(s) and NUI Galway in accordance with publisher policies. Please cite the published version when available.

Title	Web Services: Quo Vadis?
Author(s)	Bussler, Christoph; Fensel, Dieter
Publication Date	2003
Publication Information	Christoph Bussler, Alexander Maedche, Dieter Fensel "Web Services: Quo Vadis?", IEEE Intelligent Systems, Trends and Controversies: Web Services: Been There, Done That?, 18(1), 2003.
Item record	http://hdl.handle.net/10379/1323

Downloaded 2019-02-23T03:13:35Z

Some rights reserved. For more information, please see the item record link above.



Web Services: Quo Vadis?

Christoph Bussler
Oracle Corporation
Redwood Shores, CA 94065, USA
Chris.Bussler@Oracle.com

Alexander Maedche
FZI Research Center for
Information Technologies at the Univer-
sity of Karlsruhe
76131 Karlsruhe, Germany
maedche@fzi.de

Dieter Fensel
Leopold Franzens Universitaet
Innsbruck, Institut fuer Informatik
6020 Innsbruck, Austria
dieter.fensel@uibk.ac.at

Abstract

For some time now Web Services are touted as the technology-based solution to many different significant integration problems in information technology. These integration problems are of a so different nature that the following question cannot be avoided and must be asked right now: Web Services: Quo Vadis?

This article illustrates the currently existing immense discrepancy between the magnitude of the integration problems and the simplistic functionality provided by Web Services, the impossibility to address all these integration problems with equal fidelity and it brings forward a proposal for the future direction of Web Services being able to cope with the most complex problems in the solution space in form of the Web Service Modeling Framework (WSMF).

1. Web Services at the Crossroads

Only when the Crossroads are reached the question about the future direction can be asked. In this section we therefore put the spotlight on the current state of the art of Web Services and the situation of this "movement" (or "hype" as some would be more than happy to state). We believe Web Services are at the Crossroads because of the following observations that can be made currently from various viewpoints:

- Web Services are touted as the Silver Bullet for a (too) wide array of simple and complex integration problems that are being address currently by complex solutions like B2B integration technology for B2B integration [Bussler 2002a] or Java for remote server invocation [Java].
- Many different, competing, conflicting and overlapping standards exist that are addressing Web Service functionality [Robin Cover] [zapthink].
- Many research projects are repackaging existing work as Web Service work.
- Many start-up companies provide technology implementing Web Service technology. Some of them are [Cape Clear] [Actional] [Intalio].
- Major infrastructure companies provide already implementations of Web Service technology [BEA] [IBM] [Microsoft] [Oracle].
- Broad journalistic coverage of Web Services exists in the trade press like [eAI Journal] or [ebizQ].
- Web Services have no real underlying conceptual model and are defined as implementation technology [SOAP].
- No analysis of the existing and available solutions for the problems are conducted like EDI translation software [Sherif 2000] and no lessons learned are reported from those deployments (more than 300000 EDI deployments exist world-wide according to DISA [DISA]) that could advance Web Service technology functionality.

This current situation cannot continue for a long time and a consolidation of the Web Service space has to happen in one form or another soon focusing Web Service technology toward a well-defined and well manageable set of integration problems. The current expectations of Web Services are too high and to varied for one technology being able to provide the solution for the given integration problems.

Before we outline a possible future direction for Web Services development from the Crossroads going forward some of the "Silver Bullet problems" are discussed next in more detail clearly showing the vast spectrum Web Services are said to solve.

1.1 Yet another RPC (or Distributed Object Management Technology for that matter)

Originally, SOAP was the acronym for Simple Object Access Protocol [SOAP]. Nomen est omen: the underlying paradigm follows the client/server model. The SOAP specification in conjunction with the Web Service Description Language (WSDL) [WSDL] clearly allows to define the external interface of a server, but not of its clients violating the general peer-to-peer (P2P) behavior required by most of the integration solutions. The only pattern supported is the one-way invocation with and without results between the defined roles of client and server.

In this sense SOAP in conjunction with WSDL clearly implements a remote procedure call (RPC) model. This is also explicitly called out in the SOAP specification [SOAP]. Since it is based on the ubiquitous HTTP protocol in conjunction with XML as the message syntax [XML] computer system boundaries are easy to overcome and the notion of the better RPC was born leading to the wide awareness of Web Services in the developer community.

1.2 The Better Application-to-Application Integration Technology

Since SOAP messages can be transported over HTTP in order to implement the runtime invocation it was very soon clear that bridging computer system boundaries is easy due to the commodity of the HTTP protocol.

If only packaged applications like Enterprise Resource Management (ERP) systems like [Oracle] or [SAP] were to expose WSDL interfaces. Then integrating them with Web Services would be easy and fast and cheap and manageable and ...

In such a scenario Web Services would be the future Enterprise Application Integration (EAI) solution overcoming the current costly integration technology deployments since the whole integration world would be a homogeneous sea of SOAP messages implementing WSDL operations.

1.3 THE Business-to-Business Integration Solution

And, finally, since SOAP messages can be exchanged over the Internet with ease the idea was not far to use Web Services as THE means to implement business-to-business (B2B) interactions between companies (inter-enterprise B2B communication).

The vision in this case is to replace EDI [EDI], SWIFT [SWIFT] and other existing B2B standards (that often exist for over 30 years) in no time at all.

1.4 The Current State of Web Services

These three Silver Bullet problems span an impressive array of serious requirements for an integration solution that Web Services would have to address:

- **Efficiency.** In order to scale on an industrial basis Web Service execution has to be very efficient.
- **Expressiveness.** B2B interactions in supply-chain scenarios are complex and this requires an expressive set of supported integration concepts.
- **Security.** Interactions within as well as across enterprises have to be secured to prevent security attacks of all types. Non-repudiation has to be provided.
- **Reliability.** Remote and distributed communication has to be reliable and messages have to be sent exactly once in order to ensure dependable interactions.
- **Manageability.** Especially inter-enterprise communication changes frequently requiring easily manageable technology.

These requirements pose a high demand on a technology that addresses their implementation.

On the other side there is the current state of Web Services technology. Currently, as introduced earlier, the following standards define Web Services: SOAP [SOAP], WSDL [WSDL] and UDDI [UDDI]. This standards triple, even if fully implemented, is very far from addressing all the above requirements since it is based on the primitive notion of synchronous remote invocation following a simple client/server model. Complex integration problems like B2B integration problems that are based on a peer-to-peer paradigm are not in the scope of the standards triple due to missing concepts, missing reliability protocol definition or missing security concepts.

Since Web Services have caught on quite impressively across the developer and business community it is worthwhile exploring potential future developments to capitalize on the state Web Services have reached.

2. Directions Leading from the Crossroads

The discrepancy between the current simplistic Web Service technology and the complex integration problems in reality is by now very apparent. So, what could the future of Web Services be?

The question of where to go from here can only be asked if there are alternative paths going forward. Currently, many alternatives present themselves and in order for Web Service technology to be relevant in the future (e. g. for wide industrial deployment) only some are interesting.

Some of the possible directions for Web Services are

- Be the better RPC.
- Be the better plumbing for integration.
- Be the better intelligent agent platform.
- Be the best of all of the above.

There seems to be the implicit agreement in general that Web Services should be THE new and THE better technology for integration (and especially for B2B integration). This can be seen when looking at the vast array of industrial standard proposals that all together (from some distance) provide an almost complete set of concepts required for implementing complex B2B integration like supply chain management, health care organization integration or financial transaction processing.

At [zapthink] a poster can be found that shows 150 of the at that time 450 industrial standards classified and related to each other. They define business documents, security services, transport protocols, packaging, trading partner management and many other relevant standards.

Not only the industry works on a complete set of standards addressing the complete integration space. Academic research is also tackling the vast array of problems in inter-enterprise integration as can be seen when looking at workshops like [TES 2002] or [WES 2002] as well as research papers like [Casati et al. 2001], [Bussler 2002b], [Maedche et al. 2003].

The big HOWEVER is, that the need for 150 (450) standards indicates a broken conceptual model since the standards conflict,

contradict, overlap, compete or disagree with each other significantly.

Another big HOWEVER is, that semantic unification is not taken into consideration at all. Only a few contribution (standards and research) actually highlight and emphasize the need for solving the semantic mismatch problem that inherently exists in inter-enterprise integration and that no real integration solution can do without.

THEREFORE, a comprehensive conceptual model is required that combines all aspects of a solution for the inter-enterprise integration problem. From that a comprehensive modeling and execution framework can be derived that can really address the integration problem completely.

Such a framework has to provide a holistic and general conceptual model that allows to characterize all integration problems. From that a set of expressive modeling constructs can be derived that allows to model web services, their invocation as well as all the other requirements like composition as well as security addressing even the most complex integration scenarios.

Furthermore, such a framework has to provide an expressive and scaleable mediation approach that allows to tackle the semantic unification problem through powerful mediation.

The next and final section will introduce such an effort in more detail called the Web Service Modeling Framework (WSMF) [Fensel et al. 2002].

3. The Future of Web Services: WSMF

Web Services that can address the complex integration problems require conceptual modeling based on a well-defined set of integration concepts. This set of concepts has to be able to represent all relevant aspects of integration including document formats, processes, security, trading partner management, mediation, discovery, amongst others.

Since integration takes place always between two or more parties exposing services, any integration solution is "in-between" the parties. This calls for an integration framework that clearly identifies the system boundaries and modes of interaction between all elements. Furthermore, it has to define the execution model for the conceptual model so that the application of concepts is uniform.

Last but not least, a methodology is needed that suggests best practices for the modeling task when integrating services of several parties. The methodology suggests the appropriate use of the integration concepts.

The Web Service Modeling Framework (WSMF) is an important step towards such a conceptual model in conjunction with an integration framework and methodology. It provides all the integration concepts necessary for modeling even the most complex integration scenarios, it provides a framework that defines the meaning of the concepts and suggests a methodology.

The major aspects of WSMF are besides the mere definition of complex web services a strong support for mediating data as well as behavior. In general, different services expose business data following different schemas and expect different message exchange patterns. For two or more services to interact, the differences in data structure and meaning as well as in message exchange behavior has to be mediated. WSMF recognizes a significant set of mismatch patterns and provides mediation support for those. Only after the mediation problem is solved integration becomes possible.

In order for the web service integration and mediation to scale the concepts have to be represented as a formal language so that particular Web Service definitions are self-describing, machine interpretable and machine executable. Ontologies are a very suitable technology for this purpose due to their ability to represent semantics in form of concepts.

The formal representation of services, their data and behavior is a precondition for automatic Web Service registry, discovery and composition. Only a strictly formal approach allows the automatic and meaningful Web Service composition. [UDDI] is far

from providing such a formal mechanism today and WSMF can serve as an important input to this development.

With WSMF as a foundation service-oriented architectures become possible where all elements that require integration are represented as services. Once a service is formally defined, it is registered, can be discovered and composed in order to achieve its integration for business goals like supply-chain automation. WSMF provides the formal foundation that allows service-oriented architectures to be executable in a complex business world.

We therefore strongly suggest that Web Services will choose the path towards an integrated and complete conceptual integration model supported by a framework defining the execution semantics and a methodology for successful definition of integration.

4. References

[Actional] www.actional.com

[BEA] www.bea.com

[Bussler 2002a] C. Bussler: The Role of B2B Engines in B2B Integration Architectures. ACM SIGMOD Record, Special Issue on Data Management Issues in E-Commerce, Vol. 31, No. 1, March 2002

[Bussler 2002b] C. Bussler: The Application of Workflow Technology in Semantic B2B Integration. Distributed and Parallel Databases. An International Journal. Kluwer International Publishers. September - November 2002, Volume 12, Issue 2-3

[Cape Clear] www.capeclear.com

[Casati et al. 2001] F. Casati, M. Sayal, M.-C. Shan: Developing E-Services for Composing E-Services. In: Proceedings of the 13th International Conference on Advanced Information Systems Engineering (CAISE'01), Interlaken, Switzerland, 2001

[DISA] www.disa.org

[EDI] www.x12.org

[eAI Journal] www.eaijournal.com

[ebizQ] www.ebizq.net

[Fensel et al. 2002] D. Fensel, C. Bussler: The Web Service Modeling Framework WSMF. Electronic Commerce Research and Applications, Vol 1, No. 2. Elsevier Science B. V., 2002

[IBM] www.ibm.com

[Intalio] www.intalio.com

[Java] java.sun.com

[Maedche et al. 2003] Services on the Move - Towards P2P-Enabled Semantic Web Services. In: Proceedings of the Tenth International Conference on Information Technology and Travel & Tourism, ENTER 2003, Helsinki

[Microsoft] www.microsoft.com

[Oracle] www.oracle.com

[Robin Cover] www.oasis-open.org/cover/sgml-xml.html

[Sherif 2000] M. Sherif: Protocols for Secure Electronic Commerce. CRC Press, 2000

[SAP] www.sap.com

[SOAP] Simple Object Access Protocol (SOAP) 1.2.
<http://www.w3.org/TR/soap12-part1/> and
<http://www.w3.org/TR/soap12-part2/>

[SWIFT] www.swift.com

[TES 2002] A. Buchmann, F. Casati, L. Fiege, M.-C. Hsu, M.-C. Shan (Eds.): Technologies for E-Services. Third International Workshop, TES 2002, Hong Kong, China, August, 2002

[UDDI] www.uddi.org

[WES 2002] C. Bussler, R. Hull, S. McIlraith, M. E. Orłowska, B. Pernici, J. Yang (Eds.): Workshop on Web Services, E-Business, and the Semantic Web (WES). WES 2002, Toronto, Canada, June 2002

[WSDL] Web Service Description Language (WSDL) 1.1.
www.w3.org/TR/wsdl

[XML] www.w3.org/XML/

[zaphthink] www.zaphthink.com