

February 2007

# Negotiations in Service-Oriented Architectures

Falk Kretzschmar

*Martin-Luther-University Halle-Wittenberg*, kretzschmar@wiwi.uni-halle.de

Follow this and additional works at: <http://aisel.aisnet.org/wi2007>

---

## Recommended Citation

Kretzschmar, Falk, "Negotiations in Service-Oriented Architectures" (2007). *Wirtschaftsinformatik Proceedings 2007*. 29.  
<http://aisel.aisnet.org/wi2007/29>

This material is brought to you by the Wirtschaftsinformatik at AIS Electronic Library (AISeL). It has been accepted for inclusion in Wirtschaftsinformatik Proceedings 2007 by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

In: Oberweis, Andreas, u.a. (Hg.) 2007. *eOrganisation: Service-, Prozess-, Market-Engineering*; 8. Internationale Tagung Wirtschaftsinformatik 2007. Karlsruhe: Universitätsverlag Karlsruhe

ISBN: 978-3-86644-094-4 (Band 1)

ISBN: 978-3-86644-095-1 (Band 2)

ISBN: 978-3-86644-093-7 (set)

© Universitätsverlag Karlsruhe 2007

# Negotiations in Service-Oriented Architectures

Falk Kretzschmar

Dept. of Management Information Systems  
Martin-Luther-University Halle-Wittenberg  
Universitätsring 3  
D-06108 Halle, Germany  
kretzschmar@wiwi.uni-halle.de

## Abstract

The software infrastructure of today's enterprises is transforming from monolithic application software into more flexible component-based and modular architectures. In this context Service-Oriented Architectures (SOAs) are discussed as a paradigm where application components can be accessed and exchanged through a service broker. Existing standards for implementing the service broker in SOA only allow searching for services based on functional parameters. Especially in a B2B setting where the involved enterprises are economically independent instances the price for the exchanged services is of particular interest. This causes the increasingly important question of how the economic perspective can be added to SOA. Negotiations between service provider and service requestor are one possible extension to the basic architecture. Subject of these negotiations can be non-functional parameters such as price and Quality of Service (QoS) for the exchanged services.

## 1 Introduction

Negotiations in Service-Oriented Architectures (SOAs) are a current and interdisciplinary field of research. The idea that applications can be automatically composed out of reusable components seems to be very promising for researchers as well as for practitioners. One foundation to achieve this flexible application architecture is to choose the right components based on their functional and non-functional parameters. A further step to a more flexible composition of reusable components is to negotiate the terms of use for these components.

In this article we summarize key concepts for negotiations in general and we discuss their applicability for extending SOA. The article is organized as follows: After a short introduction and the definition of fundamental terms the second part of the article introduces main negotiation concepts for including an economic perspective in SOA. The third part suggests technologies for the implementation of the described negotiations with special emphasis on semantic description for the exchanged services. The article ends with a discussion of the proposed concepts and a short outlook on future work.

The main idea of SOA is to provide the functionality of applications as a service and to allow a simple way to access this service via a web infrastructure. Important goals for using a SOA are reuse of existing components, interoperability, loose coupling, and the possibility to flexibly adjust the applications to the company's business processes [Alon03], [Zimm03]. According to the World Wide Web Consortium (W3C) specification SOA consists of three main parts: The service provider publishes and provides services, the service broker contributes to publishing and finding of the services and the service requestor finds the adequate services. Figure 1 shows the mentioned SOA components and their interaction [W3C104].

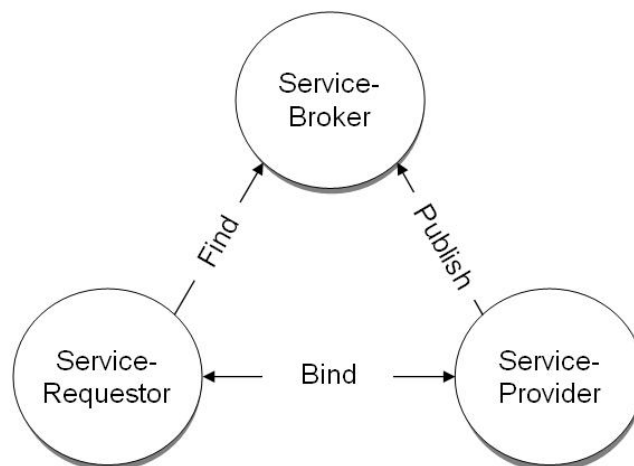


Figure 1: Components of a Service-Oriented Architecture

Web services are currently the most popular technology for implementing a SOA. The basic web services standards are Simple Object Access Protocol (SOAP) for messaging, Web Service Description Language (WSDL) for interface description, and Universal Description, Discovery, and Integration (UDDI) as an optional technology for implementing the service broker. Besides this set of XML-based technologies there exist many extensions to SOA in terms of composition and coordination of services as well as to ensure transactions and security.

In addition to these extensions there exists the approach to use the ideas of the semantic web in SOA. Standards like Web Ontology Language (OWL) and Resource Description Framework (RDF) were developed to describe the semantics of web resources. Web service discovery based on OWL descriptions is a very promising idea, but there is still a lot to be done with regard to the semantic description of services to finally provide an automated semantic search for web services. This automated semantic search is one of the main foundations for negotiations between service provider and service requestor. Before actually implementing a negotiation mechanism in SOA, there needs to be a description of the traded services which is understood by both negotiation parties. This means a common ontology is needed to specify the context of the negotiation and the terms of use for the exchanged services. Currently this is not possible with the basic SOA standards: UDDI allows the search for services based on their functionality. Other search criteria such as price and Quality of Service are not considered, which means the negotiation about economic criteria is still taking place outside the basic architecture in a separate process. This contradicts the above mentioned goals for using SOA, in particular the flexible on-demand access to services and the loose coupling between service provider and service requestor.

## **2 The economic perspective on SOA**

### **2.1 Definition and classification**

From a technological perspective SOA is an architecture for a distributed system that provides the functionality of application components as services. These services are accessed by using the common internet protocols. Whereas in the domain of computer science and information systems the technological aspects of SOA are well researched the economic perspective on SOA is often neglected.

From an economic perspective the service provider is offering a digital service and the service requestor is consuming it on-demand. So compared to the technological perspective on SOA technical interactions become economic transactions.

Negotiations in SOA describe the process of elaborating an agreement between service provider and service requestor based on economic criteria. Besides functional search criteria non-functional criteria such as price and Quality of Service are considered as important negotiation

criteria for exchanging services in SOA [Bole04], [Dan04], [Ludw03], [Sing05]. On the one hand the service provider describes a service offer based on these criteria and on the other hand the service requestor is searching for adequate services based on his search criteria. Negotiating an agreement between service provider and service requestor leads to an economic coordination between the involved negotiation parties. To analyze the economic coordination in more detail it is a common approach to look at the negotiation process using a phase model. Using this model the economic coordination in SOA is a process consisting of four transaction phases. During the information phase the service requestor is getting information about offered services and possible transaction partners. In the negotiation phase the transaction partners are selected and the contract for using the service is elaborated. This contract is called Service Level Agreement (SLA), which is established and agreed upon in the agreement phase. The fulfillment phase finally comprises service delivery, payment and further support. Figure 2 shows the four transaction phases that are similar to the transaction phases in electronic markets [Lind98].

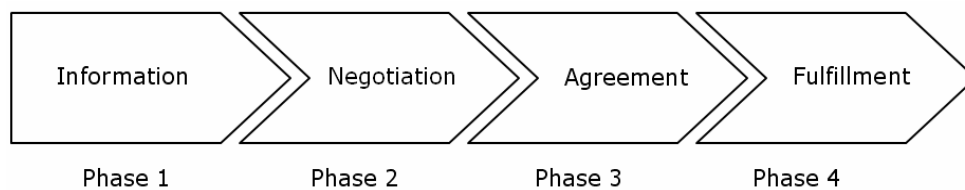


Figure 2: Transaction phases for the economic coordination in SOA

For implementing the negotiation phase there exist many different concepts. Among them are the very prominent auction mechanisms. These negotiation concepts are often referred to as general concepts without an explicit relation to negotiations in SOA. In [Sege99], [Bich01] it is mentioned that simple auctions, extended auctions, negotiation support systems, and agent-based negotiations are main concepts for implementing the negotiation phase. Table 1 shows a classification of negotiation concepts, which considers the architecture, negotiation rules, number of negotiation attributes, and the degree of automation.

Category:	Negotiation concepts:
Architecture	1:1 negotiation, m:n marketplace, brokered marketplace
Negotiation rules, number of negotiation attributes	Simple auction mechanisms: English auction, Dutch auction, Vickrey auction
	Extended auction mechanisms: double, multi-unit, multi-attribute auction
Degree of automation	Negotiation Support Systems, Agent-Based Negotiations

Table 1: Classification of negotiation concepts

These different concepts for implementing the negotiation phase derive from the parameters that characterize a particular negotiation scenario. Many authors state that besides the pure negotiation mechanism the actual setting of a negotiation is of main importance [Kris02], [Milg04]. To describe the setting of negotiation scenarios in SOA we use the parameters that are listed in table 2.

Category:	Negotiation parameters:
Negotiation partners	Number of negotiation partners, Strategic behavior of negotiation partners
Negotiation mechanism	Rules for matchmaking of negotiation partners, Rules for pricing the traded service
Negotiation subject	Properties of the traded service
Technological setting	Network and server performance, Standards and communication protocols, Semantic description of services and negotiation ontology

Table 2: Parameters for the negotiation setting

The above mentioned parameters can be used for a general description of negotiation scenarios. Depending on these parameters the outcome of the negotiation phase varies considerably. For instance the number of negotiation partners influences the negotiation power of one particular

negotiation partner and leads to specific strategic behavior. Changing the number of negotiation partners means that their original strategy is not appropriate anymore. In addition to the strategy derived from the number of participants it is possible to define a basic strategy for each actor or for types of actors. This strategy is based on the rules that are defined in the negotiation mechanism.

We explained earlier that in the negotiation phase transaction partners are matched and the price of the traded service is determined. The second category of negotiation parameters describes the actual matchmaking and the pricing of services. In the case of negotiations between service provider and service requestor the properties of the traded service can be important negotiation parameters as well as technical parameters like network and server performance or the availability of semantic descriptions for services.

## **2.2 Negotiations in SOA and possible architectural extensions**

For the scope of this article the negotiation phase is of special interest. We examine the question of how the transaction partners should be matched and what pricing mechanism should be used for trading services in SOA. In 2.1 we named general concepts for implementing the negotiation phase. In real-world negotiations the different negotiation concepts can occur as a combination of the above mentioned categories. For instance, on a brokered marketplace a multi-attribute auction can be used for negotiating between software agents. Our goal is to adapt the general negotiation mechanisms so that they match with the special setting for negotiations in SOA.

SOA is an architectural concept for application software which is often referred to as software as a service [Elfa04], [Sing05]. So in broad terms the negotiation subject is a service, the negotiation partners are applications or organizations that run these applications and the technological setting is the Internet. Because of the SOA goal to flexibly adjust to a company's business processes by selecting the best suited service for the current task, we need automated selection of services based on a semantic search. For developing a concept for negotiations in SOA this means we also need an approach for automated negotiations. Negotiation Support Systems include a human decision and are therefore not matching with the above mentioned SOA goals and the technological setting. By contrast there are also negotiation concepts that allow automated negotiations between negotiation partners. Agent-based negotiations fall into that category. In agent-based negotiations a strategy based on a fixed set of rules can be



implemented. This means the software agent is able to fulfill the task of an automated search and selection of services.

For the remainder of this article we focus on an extended service broker for automated negotiations in SOA. By definition SOA implies the use of a service broker, but from the technological perspective SOA allows 1:1 negotiation, m:n marketplace as well as a brokered marketplace for negotiations between service provider and service requestor. Until now many companies do not use a service broker when implementing SOA, because it is not necessary for them. Integrating existent applications within a company is usually done without a service broker, although from the technology perspective the service broker can be used for intra-company transactions as well as for inter-company transactions. Despite the fact that the idea of a global marketplace for service has not been implemented yet, we consider the service broker to be a good starting point for extending SOA, because the broker can be used for storing metadata about services and service providers. Extending the service broker leads to a central instance that allows automated negotiations of service parameters. In table 3 main concepts for an extended service broker in SOA are listed. Applying the different pricing models means that the standard service broker is transformed into a simple catalogue, an extended catalogue, a simple auction mechanism, or an extended auction mechanism. Agent-based negotiations can also be used for automated negotiations in SOA, but they do not determine a special pricing model.

Pricing model:	Service broker:
Fixed price, “take it or leave it” offer	Simple catalogue
Fixed price, differentiated price	Extended catalogue
Dynamic pricing	Simple auction mechanisms
Negotiation of price, Quality of Service and other contract parameters	Extended auction mechanisms, Agent-Based Negotiations

Table 3: Concepts for an extended service broker in SOA

The above mentioned concepts for negotiations in SOA describe different pricing models for the traded service. Fixed prices and differentiated prices are a simple way to incorporate the economic perspective in SOA. The service provider is getting paid for the functionality of a software component and for the promise to deliver it on-demand. Ignoring the cost of developing and providing services would mean that the service provider will use other channels to distribute his software components. The problem with fixed prices and differentiated prices is that the service provider needs to know the demand function for the traded services. In real-world scenarios this is often not the case. Incomplete information about the market can be a problem for trading traditional products as well as for trading software services. With the different auction mechanisms the pricing of the traded services is done dynamically and according to the auction type specific rules. The service provider is getting information about the demand function for the traded services and usually earns higher profits.

### **3 Technologies for implementing negotiations in SOA**

#### **3.1 Existing standards**

UDDI is often called a global marketplace for offering web services. In fact it is a catalogue that can be used by service providers to publish their services and by service requestors to find them. Because of the deficit of semantic service description in UDDI, it does not support automated negotiations between the involved transaction partners.

Nevertheless UDDI is a good starting point for extending SOA, since it already implements a structure for storing metadata: “white pages” include provider information, “yellow pages” provider and service information, and “green pages” information for calling the service. How these categories are matched with the UDDI data structure is described in figure 3. In addition to that it is shown that UDDI finally provides the reference to a WSDL document [Sing05].

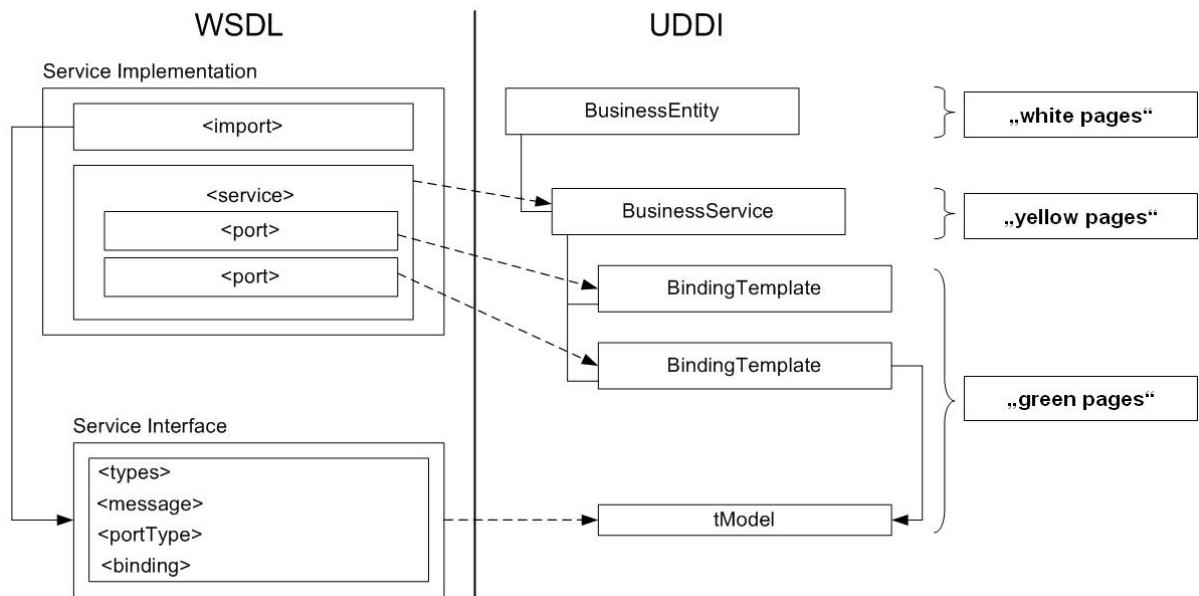


Figure 3: Link between UDDI metadata categories, UDDI data structure, and WSDL

With this reference the binding of the actual service is done through direct interaction between service provider and service requestor. Service binding is done either statically or dynamically. With the static approach services are connected to each other at design time. This can be problematic, in the case of changes of interfaces, messages, transport protocols, and network addresses. Furthermore static binding is an inflexible and in the case of frequent changes costly way to operate a distributed system. By contrast the dynamic approach allows the binding of services at runtime. With this approach UDDI delivers a set of services that implement exactly the same interface.

For automated negotiations in SOA this is not enough, because the same interface does not guarantee the same semantics of two services. In addition to that the metadata about services and service providers in UDDI is not sufficient. For that reason there already exist a number of extensions to the basic architecture. In majority these extensions were developed in research projects and make use of semantic web technologies to extend SOA. We differ between the following two categories:

1. Service broker with semantic extensions
2. Interface description with semantic extensions.

Extending the service broker is a sound method for developing a catalogue that includes more information on services than UDDI. When incorporating negotiation mechanisms like auctions the service broker is the central instance for clearing offers and counteroffers. Research projects and prototypes that fall into the category of a service broker with semantic extensions are UDDIe, OWL-S Matchmaker, and WS-QOS [Shai03], [Chen03], [Srin04], [Paol02], [Gram03]. Interface descriptions with semantic extensions match with SOA implementations that do not use a service broker, because the functional and non-functional parameters of services are described by the interface. In this case lightweight approaches of service discovery like Web Service Inspection Language (WSIL) can be used, but implementing UDDI is not necessary. Research projects and prototypes that fall into the category of interface descriptions with semantic extensions are WSOL, SLang, WSML, WSDL-S, and WSLA [Pate03], [Lama03], [Verh04], [Tian04].

### 3.2 Extended service broker architecture for automated negotiations in SOA

For implementing automated negotiations in SOA we propose the extended service broker architecture. It consists of a standard UDDI registry extended by Web Ontology Language for Services (OWL-S) service profiles, the OWL-S matchmaker and the negotiation mechanism. A conceptual overview of this extended service broker is depicted in figure 4.

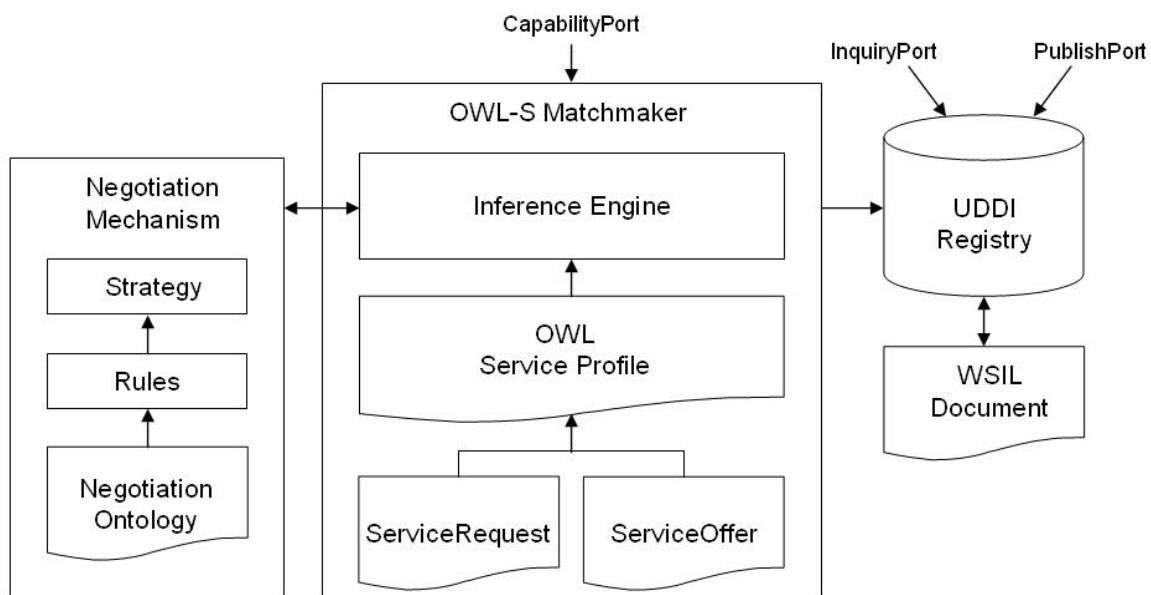


Figure 4: Extended service broker architecture for automated negotiations in SOA

Compared to a standard service broker with this architecture it is possible to describe service profiles. With the help of these service profiles the service requestor can define a service request, which contains parameters the service provider should guarantee. For the service provider the service offer contains service parameters he is able to guarantee. Based on that semantic description of services the inference engine of the OWL-S matchmaker matches the two profiles and by using the UDDI registry or the WSIL document it can finally discover and select the requested service. How the actual matching is done is implemented in the negotiation mechanism that defines the negotiation rules and the negotiation strategy.

An important part of the extended service broker architecture is the negotiation ontology. OWL-S can be used to define an ontology that allows a common understanding of the negotiation subject [Buss02], [Paol02]. For automated negotiations in SOA we propose a negotiation ontology that consists of several non-functional service parameters. Table 4 shows the main non-functional service parameters service price and Quality of Service as well as additional parameters like reputation of negotiation partners or duration of service lease.

Non-functional service parameters:	
Service price	
Quality of Service	Server performance: response time, availability, throughput, reliability Network performance: bandwidth, latency, jitter
Security	
Negotiation mechanism	
Number of negotiation partners	
Reputation of negotiation partners	
Duration of service lease	
Sanctions or penalties in case of non-compliance to the service level agreement	

Table 4: Non-functional service parameters for negotiations in SOA

Negotiating a Service Level Agreement between service provider and service requestor is a complex task. Before the actual negotiation process starts the service profiles for searching

services and for providing services have to be defined. In addition to that for the service provider it is necessary to establish an architecture for measuring and monitoring the service parameters. During the negotiation phase all parameters must be understood by both negotiation parties. The negotiation ontology describes the meaning of the above mentioned non-functional service parameters in a machine-readable way. Using this description the rules of the negotiation mechanism consider the service parameters and their meaning for the negotiation parties.

We argue that for the main non-functional parameters price and Quality of Service it is likely that service provider and service requestor have a common understanding. Furthermore service provider and service requestor know how to measure these service parameters. With the additional non-functional parameters such as reputation of negotiation partners and security it is difficult to come to an agreement. Measuring and monitoring the additional non-functional service parameters is more complex, because it requires a reputation mechanism and policies for secure service delivery.

At this point the proposed extended service broker architecture for automated negotiations in SOA is a conceptual architecture. Semantic service discovery, which is part of the architecture, is a main foundation for implementing automated negotiations in SOA.

#### **4 Discussion and future work**

This article gave a short overview of existing technologies and further concepts for implementing negotiations in SOA. We argued that in the case of economically independent enterprises the economic perspective for exchanging services is of particular interest. We introduced possible extensions to SOA and suggested current technologies for their implementation. The most important extension is an architecture that allows automated negotiations of price and other parameters such as Quality of Service for the exchanged services.

The use of semantic web technologies like OWL-S is a very promising approach to allow an economic coordination in SOA. The proposed extended service broker architecture makes use of a negotiation ontology that allows a common understanding of the negotiation subject. Service profiles are used to describe service requests and service offers that are matched by the inference engine of the OWL-S matchmaker.

For further research it is necessary to work on a SOA specific and more flexible negotiation mechanism. The goal to adapt the mentioned standard negotiation mechanisms so that they match with the technological setting of SOA can be achieved by conducting further research on the outcome of negotiations in SOA. Our future work will include an analysis of design decisions and their impact on profit of service providers, achieved utility of service requestors, and delivery rate of services.

## References

- [Alon03] Alonso, G.: Web Services: Concepts, Architectures and Applications. Springer, 2003.
- [Bich01] Bichler, M.: The Future of E-Markets: Multi-Dimensional Market Mechanisms. Cambridge University Press, Cambridge, 2001.
- [Bole04] Boles, D.: Erlösformen für Web Content und Services. Informatik Springer, 2004.
- [Buss02] Bussler, C., Fensel, D., Maedche, A.: A conceptual architecture for semantic web enabled web services. ACM SIGMOD Record, v.31 n.4, December 2002.
- [Chen03] Chen, Z. et al.: UX -An Architecture Providing QoS-Aware and Federated Support for UDDI, ICWS '03, School of Computer Engineering, Nanyang Technological University, 2003.
- [Dan04] Dan, A. et al.: Web services on demand: WSLA-driven automated management. IBM Systems Journal, v.43 n.1, p.136-158, 2004.
- [Gram03] Gramm, A.: WS-QOS - Ein Rahmenwerk für Dienstgüteunterstützung in Web Services, Institut für Informatik, Freie Universität Berlin, 2003.
- [Elfa04] Elfatatry, A., Layzell, P.: Negotiating in Service-oriented Environments. Comm. ACM, 47 (8), 103-08., 2004.
- [Kris02] Krishna, V.: Auction Theory. Academic Press, 2002.

- [Lama03] Lamanna, D. et al.: SLAng: A language for Service Level Agreements, Departement of Computer Science, University College London, 2003.
- [Lind98] Lindemann, M.A., Runge, A.: Electronic Contracting within the Reference Model for Electronic Markets, 6th European Conference on Information Systems (ECIS-98). Aix-en-Provence, France, 1998.
- [Ludw03] Ludwig, H.: Web Services QoS: External SLAs and Internal Policies Or: How do we deliver what we promise? In Proc. of the First Web Services Quality Workshop, IEEE Computer Society, Rome, 2003.
- [Milg04] Milgrom, P.R.: Putting Auction Theory to Work (Churchill Lectures in Economics). Cambridge University Press, 2004.
- [Paol02] Paolucci, M. et al.: Importing the Semantic Web in UDDI, in Proceedings of Web Services, E-business and Semantic Web Workshop 2002.
- [Pate03] Patel, K., Tasic, V.: Web Service Offerings Language (WSOL): Characteristics, Applications, and Tools. 2003.
- [Shai03] Shaikh A. et al.: UDDIe: An Extended Registry for Web Services, Department of Computer Science, Cardiff University, UK, 2003.
- [Sege99] Segev, A., et al.: On Negotiations and Deal Making in Electronic Markets, Information Systems Frontiers, v.1 n.3, p.241-258, October 1999.
- [Sing05] Singh, M., Huhns, M.: Service-Oriented Computing: Semantics, Processes, Agents. Wiley, 2005.
- [Srin04] Srinivasan, N. et al.: Adding OWL-S to UDDI, implementation and throughput, First International Workshop on Semantic Web Services and Web Process Composition, San Diego, California, USA, 2004.
- [Tian04] Tian, M. et al.: A Survey of current Approaches towards Specification and Management of Quality of Service for Web Services. Freie Universität Berlin, Institut für Informatik, 2004.



- [Verh04] Verheecke, B. et al.: AOP for Dynamic Configuration and Management of Web Services, *International Journal of Web Services Research*, 1(3), 25-41, 2004.
- [W3C104] Booth et al.: *Web Services Architecture*. W3C Working Group Note, 2004.  
<http://www.w3.org/TR/ws-arch/>
- [Zimm03] Zimmermann, O., Tomlinson, M., Peuser, S.: *Perspectives on web services: applying SOAP, WSDL, and UDDI to real-world projects*. Springer, Berlin, 2003.

