Electronic version of an article published in **ASQF E.V. (Hrsg) Software Quality in Service-Oriented Architectures. Proceedings of the CONQUEST 2006, pp. 203 - 214**

Web Service Quality Descriptions for Web Service Consumers

Jürgen Rückert, Barbara Paech

Institute for Computer Science, University of Heidelberg, {rueckert, paech}@informatik.uni-heidelberg.de

Abstract

Web service consumers require web service quality descriptions for a variety of tasks; to understand the quality offered, to define the quality required, to create service level agreements and to monitor the quality received from service providers. Currently, only task-specific quality description languages support these tasks. Our goal is that web service consumers should be able to fulfill all of these tasks and that quality descriptions can be applied without language barriers. Thus, quality descriptions have to be merged. This article presents the tasks of a web service consumer that deal with quality descriptions and it defines requirements for a quality description language that supports all introduced tasks. Existing web service quality descriptions are analyzed and it is checked how many of the presented tasks they fulfill.

1 Introduction

The service-oriented architecture (SOA) enables enterprises to compose their applications using services that are offered either in-house or in the Web [WSA]. For users of web services technology there is a great variety of web services specifications that promise advantages due to standardization; service consumers gain the advantage of interoperability and self-descriptiveness of functionality and quality [ZTP03]. The specification of the functionality of web services by interface syntax and behavior is not sufficient – the quality has to be specified as well, especially for contractual obligations. The contract itself is the basis for commercial usage of web services.

This article presents existing approaches that support web service usage based on quality descriptions from the point of view of a service consumer. In section 2 we identify new service consumer tasks that are important when dealing with quality. And we define requirements for a quality description language that is able to support all identified tasks. In section 3 we list available quality description languages. Based on this overview, we judge if the quality description language is suitable to fulfill the identified

new tasks. In section 4 we summarize the article and define future steps for our research.

2 Application and benefit of quality descriptions

In this section we introduce new tasks for service consumers that are important when dealing with web service quality descriptions. At the end of the section we define requirements for a quality description language that is able to support the new tasks.

2.1 Querying offered quality

In the simplest case a service consumer just queries the offered quality of a service (*query offered quality* in Figure 1). This is based on the assumption that the service provider offers descriptions of the deliverable quality (*describe offered quality* in Figure 1). If a service provider wants to make multiple offers for a service, it is reasonable to offer several service quality levels (Quality of Service (QoS) levels). A stock information service e.g. could offer real-time stock quotes at an availability of 99.99% without encryption or delayed stock quotes with 100% availability but without encryption.

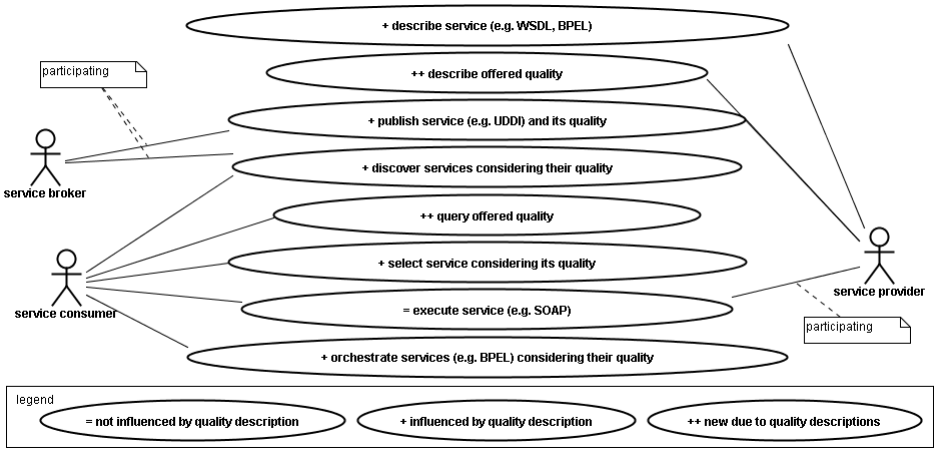

Fig. 1: Commonly accepted roles and tasks for web service usage

The offered quality might serve to choose services (*discover services considering their quality* in Figure 1) [DSS+04] [MCDM05] or it might serve for orchestrating services *(orchestrate services considering their quality* in Figure 1) [YL05]. After service discovery, which results in a list of services, the service consumer might want to evaluate the quality description for each discovered service in order to find the best fitting service (*select service considering its quality* in Figure 1). For service discovery, the service provider has to publish quality information (*publish service and its quality* in

Figure 1) and for service orchestration, the service provider has to enhance the process description with quality information (*describe service* in Figure 1). The service execution itself does not depend on any quality description (*execute service* in Figure 1). The technologies presented in Figure 1 (like Web Service Description Language [WSDL], Universal Description, Discovery and Integration [UDDI], SOAP [SOAP] und Business Process Execution Language [BPEL]) are examples that show the technical feasibility for supporting web service usage tasks. There are alternative, competing technologies like e.g. XML-RPC [XMLRPC] and REST [REST] as alternatives for SOAP, as well as semantic web service technologies like e.g. OWL-S [OWLS] und WSMO [WSMO] complementing WSDL and BPEL.

Certain quality attributes can only be rated individually by service consumers like e.g. “response time that the service consumer receives”. In such cases the service provider may offer its current experiences resulting from past web service usages [MS04]. The service provider therefore may have the task *collect service usage experiences* and *evaluate service usage experiences* in Figure 2.

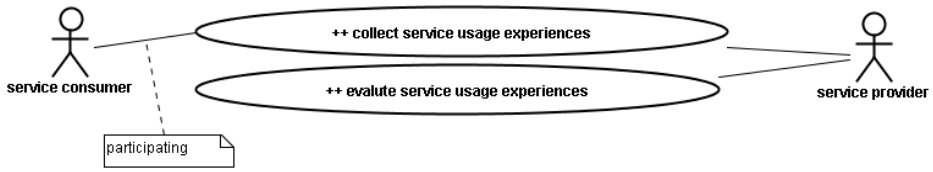

Fig. 2: Roles and tasks to query offered quality

2.2 Predicting prospective quality

Service consumers may have the additional task of predicting the quality (*predict delivered quality* in Figure 3), which is independent of quality descriptions offered by the service provider. Often, the web service quality cannot be predicted because of missing knowledge concerning past service usages (“history of prior transactions”), because the service provider cannot be trusted or because of missing service provider references. In order to compensate these limitations, the service consumer creates his/her own opinion (*evaluate service opinions* in Figure 3) using a trustworthy, third party (*service agency* in Figure 3), that aggregates web service appraisals based on service usages by as many different service consumers as possible (*collect service opinions* in Figure 3) [MS04]. The major task of a *service agency* is the collection and offer of aggregated service usage information for public service.

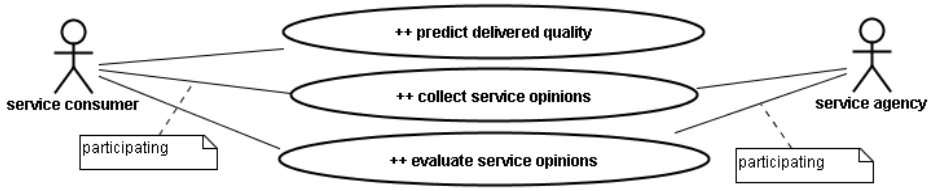

Fig. 3: Roles and tasks to predict delivered quality

2.3 Stating, negotiating and stipulating required quality

In a simple usage scenario the service consumer just executes a service - without agreeing on any contract. But even then, a contract is accepted; a so-called implicit contract that settles general conditions (fixed terms).

In an advanced usage scenario the service consumer states the desired quality that the service provider has to guarantee (*define required quality* in Figure 4). Both parties start negotiating until they agree on a service contract (*negotiate required quality and create contract* in Figure 4). From the service consumer point of view, the service contract states details of the quality that should be delivered. Contracts are legal agreements that define rights and responsibilities of both agreeing parties. Contracts have to conform to internal directives and the company budget (*check contract against internal policies* in Figure 4) [SMJS03].

The web services technology is especially focused on a machine-readable, executable contract. It is essential to define requirements for such machine-readable contracts and to focus on how to deal with the inconsistencies (that are the same as in textual contracts) [SMJS03].

The executable contract may be the result of electronic negotiations. Electronic negotiations aim at reducing costly manual intervention by an automated negotiation process. One approach is to base the negotiation on directives and tactics [GLDK03]. Service consumer and service provider do not have to negotiate directly; an independent party may support this task – a so-called contract broker (agent, coordinator). In this case the service consumer and provider may send description of their negotiation tactics to the broker. Electronic negotiations need a matchmaking algorithm that compares and reconciles desired and deliverable quality and includes the reconciled result into the contract [ZCL04].

Service Level Agreements (SLA) are such machine-readable contracts [SMJS03]. SLAs are electronic contracts between a service consumer and provider that specify the quality level of a service and optionally include costs and penalties. SLA details are defined as Service Level Objects. SLAs serve to create long lasting business relations and are not created per service usage because the SLA creation per service usages is to costly.

[LGDK05] presents a framework for automated provisioning of services that should fulfill the contractually agreed properties (*guarantee contract* in Figure 4).

Service providers that use orchestrated service might want to derive the service contract (or a template) based on the contracts of orchestrated services [BP05].

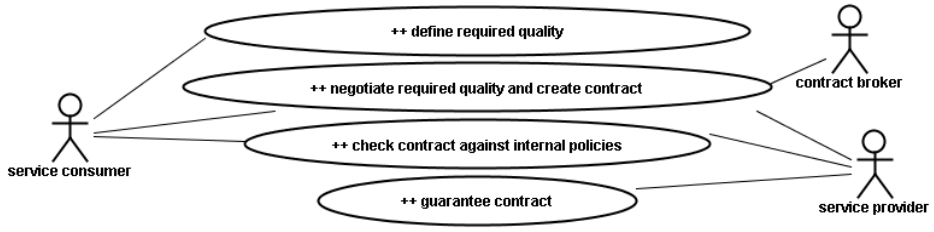

Fig. 4: Roles and tasks to stipulate quality

2.4 Measuring delivered quality

Both, service consumer and provider, are interested in measuring (*measure delivered quality* in Figure 5) the quality actually delivered in order to answer at least these two questions: “Did the service fulfill the agreed quality?” (*check contract fulfillment* in Figure 5) [BGO06] and “Which actions do we have to take in order to fulfill the agreed quality?” [SBNH05].

When monitoring the service quality, perceptions of requirements engineering might be useful. One approach is to apply the goal-based requirements analysis technique KAOS for discovering the metrics (data to measure) and automatically derive monitoring components from requirements [Rob03]. A common requirement for monitoring components is parallel execution in order to influence executing services as little as possible.

When monitoring the quality of an orchestrated service, which even may use services of another service provider, it is additionally important to collect, aggregate and interpret monitored single service calls that strongly depend on each other [Rob03].

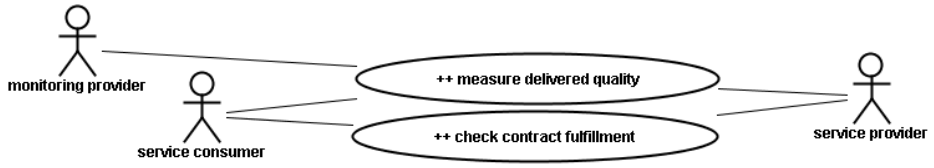

Fig. 5: Roles and tasks to measure delivered quality

2.5 Rating delivered quality

Web service scenarios may require the calculation of the total value of service usage (not necessarily a currency-based price) which rates functionality and quality (*do ser-*

vice usage rating in Figure 6). Pricing after service usage enables service providers to create invoices (*create customer invoices* in Figure 6) [JF05]. Pricing according to service usage enables service consumers to verify the delivered functionality and quality. For rating and billing several data have to be provided by the executing party: data concerning service consumer and his/her contracts (*administrate customer and contracts* in Figure 6) and contracted tariffs (*create tariffs* in Figure 6). Rating and billing do not have to be executed by service consumers and service providers themselves; these tasks may be outsourced to agencies.

Quality is described using quality attributes. Preferably, a measurable metrics should be given. Another option is manual rating of non-measurable quality attributes [DSGF03]. After manual rating of non-measurable quality attributes by the service consumer and metric measure by the service provider and/or consumer, both are normalized for comparison. The total rating value is calculated by weighing both values. It is essential that all parties trust each other.

Fig. 6: Roles and tasks to rate quality

2.6 Quality description requirements

In sections 2.1 to 2.5 we presented service consumer tasks that require quality descriptions. Table 1 shows the information that a quality description should contain in order to be applicable for all of the presented service consumer tasks. This information is an aggregation of tasks introduced in previous sections.

Tab. 1: Information required for a new quality description language in order to support service consumer tasks introduced in sections 2.1 to 2.5

	Required information
Query quality (section 2.1)	I1 Deliverable quality through offering bundles I2 Consumer satisfaction with delivered quality after usage
Predict quality (section 2.2)	I3 Predicted quality I4 Trust in service providers I5 General service opinion

State, negotiate and stipulate quality (section 2.3)	I6 Required quality before usage I7 Guaranteed quality in contracts I8 Forced quality in internal contract policies I9 Quality scopes for negotiations
Measure quality (section 2.4)	I10 Measured quality after usage
Rate quality (section 2.5)	I11 Contracted quality prices I12 Quality deltas between contracted and measured quality

A quality description language that contains the information I1 to I12 enables service consumers to deal with quality without language barriers. In contrast to this approach, existing quality description languages are tailor-made for special tasks only (see section 3).

3 Available quality description languages

In this section we present available quality description languages and detect which of the tasks, identified in section 2, they support.

3.1 Overview of quality description languages

Service consumers and providers who are interested in quality descriptions have to choose from a variety of quality description languages to fulfill their tasks. A short overview for the web service technology is given in Table 2.

Tab. 2: Languages that contain some information about quality

Languages for offers and contracts	
WSLA	Web Service Level Agreement Language (IBM) [WSLA].
WS-Agreement	WS-Agreement (Global Grid Forum) [WSAgreement].
WSOL	Web Services Offering Language [WSOL].
Languages for semantic web services	
OWL-S	OWL-S [OWLS] and predecessor DAML-S [DAMLS] respectively.
WSMO	Web Services Modeling Ontology [WSMO].
Languages for policies	
WS-Policy	Web Services Policy Framework [WSPolicy].

3.2 WSLA

WSLA [WSLA] allows the description of web service SLAs. WSLA uses contracts that contain Service Level Parameters (SLP) und Service Level Objectives (SLO). SLPs

describe quality attribute conditions. SLOs describe warranties und actions that both apply if the SLPs are not fulfilled. WSLA does not offer any taxonomy for quality attributes – the description of contract details is up to the specific application domain through XML Schema sub typing.

3.3 WS-Agreement

WS-Agreement [WSAgreement] serves to establish agreements between two parties. WS-Agreement offers operations for maintaining the life cycle of agreements and it offers a language for each agreement and agreement template. WS-Agreement differs from WSLA by offering a process to create and establish contracts. WS-Agreement is based on WSLA enhancing it with recent web service standards like e.g. state-based resources [WSResource] that serve to describe the life cycle of contracts. Like WSLA, WS-Agreement does not offer any taxonomy for contract details but allows application of specific contract details such as quality attributes.

3.4 WSOL

WSOL [WSOL] allows the description of quality offerings based on quality packages. WSOL is not restricted to any concrete quality taxonomy but offers the possibility to describe application specific quality conditions by linking logical expressions with web service operations. WSOL describes conditions (functional pre-, functional post- und quality), administrative data (prices, penalties, controlling parties) und service classes. WSOL uses offerings but does not know the concept of contracts. The Web Service Offering Infrastructure [WSOI] that is currently under construction should enhance WSOL with rating and billing functionality.

3.5 OWL-S

OWL-S [OWLS] is an ontology for web services. OWL-S allows the description of a web service profile (what the service does), its grounding (how to use the service) und its model (how the service works). The service profile describes the service provider, the service operations and its characteristics. The service characteristics (so called service parameters) may serve to model the quality description. OWL-S does not propose a concrete quality ontology but can be enhanced by the application of specific quality ontologies [LRPF04]. Currently it allows the namespace definition of a quality attribute and the value definition using a reference to an external ontology. OWL-S mainly focuses on automatic discovery, selection and orchestration of web services; concepts like contracts and tariffs are not included.

3.6 WSMO

WSMO [WSMO] is a service ontology based on the framework WSMF [FB02] and is defined in its own description language. WSMO allows the description of behavior, of functional and quality properties. WSMO offers concrete quality attributes; accuracy, financial QoS, network-related QoS, performance, reliability, rights, robustness, scal-

ability, security, source, subject und trust. It does not define metrics for these quality attributes. Subclassing serves to adapt and refine the application of specific quality attributes [LRPF04]. A concept for contracts is not part of WSMO.

3.7 WS-Policy

WS-Policy [WSPolicy] allows the description of capabilities and limitations (called assertions) of resources. WS-PolicyAttachments (see inside [WSPolicy]) links these WS-Policy definitions to a web service. WS-Policy does not know any special descriptions of quality and contracts. WS-Policy supports the description of application of specific quality through self-defined quality attributes, especially the description of packages that should or explicitly should not be offered.

3.8 Possibilities of applying quality description languages

Available quality descriptions are not suitable for all tasks at the same time, because they do not contain all required information required by these tasks.

WSLA und WS-Agreement are useful for describing deliverable quality (I1 in Table 1) and contracted quality (I7). It is possible to describe prices (I11) if a price is modeled as a quality attribute.

WSOL targets the description of deliverable quality (I1), monitors for measurements (I10) and the definition of prices (I11). WSOL does not allow the description of contracts (I7, I8). It is possible to reference WSOL in external contracts and contract templates.

OWL-S and WSMO are semantic languages and target at the automatic evaluation of descriptions. It is possible to describe deliverable quality and to model required quality (I1, I6) by referencing an external quality ontology. The languages do not address contracts and tariffs. Predicted quality (I3) can be modeled as well but the intended use is on service provider side.

WS-Policy does not know the terms quality, contracts and tariffs at all, but allows to state offers (I1) and quality requirements (I6). Predicted quality (I3) can be modeled as well but the intended use is on service provider side.

None of the above quality description languages are suitable for the tasks of service consumer and service provider that concern the rating of service usage (experiences and opinions, I2, I4 and I5).

Table 3 illustrates the relationship between tasks and quality description languages.

Tab. 3: Requirements and eligibility of quality description languages: intended application ✓, suitability (✓) und no usage ✗

	WSLA	WS-Agreement	WSOL	OWL-S	WSMO	WS-Policy
I1 Deliverable quality through offering bundles	✓	✓	✓	✓	✓	✓
I3 Predicted quality	✗	✗	✗	(✓)	(✓)	(✓)
I6 Required quality before usage	✗	✗	✗	✓	✓	✓
I7 Guaranteed quality in contracts	✓	✓	✗	✗	✗	✗
I8 Enforced quality in internal contract policies	✗	✗	✗	✗	✗	✗
I9 Quality scopes for negotiations	✗	✗	✗	✗	✗	✗
I10 Measured quality after usage	✗	✗	✓	✗	✗	✗
I11 Contracted quality prices	✓	✓	✓	✗	✗	✗
I12 Quality deltas between contracted and measured quality	✗	✗	✗	✗	✗	✗

4 Conclusion

In this article we showed that service consumers have new tasks when dealing with the quality of web services (section 2). Service consumers currently have the choice between several quality description languages . We showed that each quality description language is useful for specific tasks (section 3); there is no quality description language that fulfills *all* presented requirements of service consumers (Table 3).

The goal of our future research is to enable service consumers to fulfill their tasks with a single quality description language. The main advantage of an integrating quality description appears to be the elimination of language barriers.

References

[BGO06] G. Boström, P. Giambiagi und T. Olsson (2006). Qualiy of Service Evaluation in Virtual Organizations Using SLAs. First International Workshop on Interoperability Solutions to Trust, Security, Policies and QoS for Enhanced Enterprise Systems (IS-TSPQ), http://istspq2006.cs.helsinki.fi/, Bordeaux, France.

[BP05] A. Brogi und R. Popescu (2005). Towards Semi-automated Workflow-Based Aggregation of Web Services. International Conference on Service-Oriented Computing (ICSOC), Seiten 214-227.

[BPEL] Business Process Execution Language for Web Services Version 1.1 (BPEL4WS, WSBPEL), 5 May 2003, http://www-128.ibm.com/developerworks/library/specification/ws-bpel/, Spezifikation, IBM, BEA, Microsoft, SAP, Siebel Systems.

[DAMLS] DAML (DARPA Agents Markup Language) Services (DAML-S), http://www.daml.org/services/daml-s. Version 0.9, 2003-05, Defense Advanced Research Projects Agency.

[DSGF03] V. Deora, J. Shao, W. A. Gray und N. J. Fiddian (2003). A Quality of Service Management Framework Based on User Expectations. International Conference on Service-Oriented Computing (ICSOC), Seiten 104–114.

[FB02] D. Fensel und C. Bussler (2002). The Web Service Modeling Framework WSMF. Electronic Commerce Research and Applications, Vol. 1, Issue 2, Elsevier Science B.V., Summer 2002. http://www.wsmo.org/papers/publications/wsmf.paper.pdf.

[DSS+04] V. Deora, J. Shao, G. Shercliff, P. J. Stockreisser (2004), W. A. Gray und N. J. Fiddian. Incorporating QoS Specifications in Service Discovery. Second International Web Services Quality Workshop (WQW). Seiten 252–263.

[GLDK03] H. Gimpel, H. Ludwig, A. Dan und R. Kearney (2003). PANDA: Specifying Policies for Automated Negotiations of Service Contracts. International Conference on Service-Oriented Computing (ICSOC). Seiten 287–302.

[JF05] R. Jurca und B. Faltings (2005). Reputation-Based Service Level Agreements for Web Services. International Conference on Service-Oriented Computing (ICSOC). Seiten 396-409.

[LGDK05] H. Ludwig, H. Gimpel, A. Dan und R. Kearney. (2005). Template-Based Automated Service Provisioning - Supporting the Agreement-Driven Service Life-Cycle. International Conference on Service-Oriented Computing (ICSOC). Seiten 283-295.

[LRPF04] R. Lara, D. Roman, A. Polleres und D. Fensel (2004). A Conceptual Comparison of WSMO and OWL-S. European Conference on Web Services (ECOWS). Jgg. 3250 of Lecture Notes in Computer Science, Seiten 254–269. Springer, 2004.

[MCDM05] O. Martín-Daz, A.R. Cortés, A. Durán und C. Müller (2005). An Approach to Temporal-Aware Procurement of Web Services. International Conference on Service-Oriented Computing (ICSOC). Seiten 170-184.

[MS04] E. M. Maximilien und M. P. Singh (2004). Toward autonomic web services trust and selection. International Conference on Service-Oriented Computing (ICSOC). Seiten 212–221. ACM, 2004.

[OWLS] Web Ontology Language for Services (OWL-S), http://www.daml.org/services/owl-s/1.1/, 11.2004. OWL-S Coalition.

[REST] Representational State Transfer (REST). Roy Thomas Fielding, 2000. In Architectural Styles and the Design of Network-based Software Architectures, http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm. chapter 5.

[Rob03] W. N. Robinson (2003). Monitoring Web Service Requirements. In Requirements Engineering Conference (RE). Seiten 65–74. IEEE Computer Society, 2003.

[SBNH05] J. Siljee, I. Bosloper, J. Nijhuis und D. Hammer (2005). DySOA: Making Service Systems Self-adaptive. International Conference on Service-Oriented Computing (ICSOC). Seiten 255-268.

[SMJS03] E. Solaiman, C. Molina-Jimenez und S. K. Shrivastava. Model Checking Correctness Properties of Electronic Contracts (2003). International Conference on Service-Oriented Computing (ICSOC). Seiten 303–318.

[SOAP] SOAP 1.2, http://www.w3.org/TR/SOAP, W3C Recommendation, 24 June 2003. World Wide Web Consortium.

[UDDI] Universal Description, Discovery and Integration Specification (UDDI), http://www.uddi.org/. Version 3.0.2, 20041019. Organization for the Advancement of Structured Information Standards (OASIS).

[WSA] Web Service Architecture, W3C Working Group Note 11 February 2004, http://www.w3.org/TR/ws-arch/.

[WSAgreement] Web Services Agreement Specification (WS-Agreement). Proposed Recommendation, 6/28/2005. Grid Resource Allocation Agreement Protocol (GRAAP) Working Group. https://forge.gridforum.org/projects/graap-wg/document/WS-AgreementSpecification/en/7.

[WSDL] E. Christensen, F. Curbera, G. Meredith und S. Weerawarana. Web Services Description Language (WSDL) 1.1, http://www.w3.org/TR/wsdl. W3C Note, 15 March 2001.

[WSLA] Web Services Level Agreement Language Specification (WSLA), http://www.research.ibm.com/wsla/WSLASpecV1-20030128.pdf, Version 1.0, 2003/01/28. Bericht, IBM Corporation.

[WSMO] Web Services Modeling Ontology (WSMO), D2 V1.2, http://www.wsmo.org/TR/d2/v1.2/, WSMO Final Draft 13 April 2005. ESSI WSMO working group.

[WSOI] V. Tosic, W. Ma, B. Pagurek, B. Esfandiari (2003). Web Services Offerings Infrastructure (WSOI) - A Management Infrastructure for XML Web Services. Research Report SCE-03-19, The Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada, August 2003.

[WSOL] V. Tosic, B. Pagurek und K. Patel (2003). WSOL - A Language for the Formal Specification of Classes of Service for Web Services. International Conference on Web Services (ICWS). Seiten 375–381. CSREA Press, 2003.

[WSPolicy] Web Services Policy Framework (WS-Policy), http://www-128.ibm.com/developerworks/library/specification/ws-polfram/, Version 1.2, March 2006.

[WSResource] WS-Resource Framework 1.2. http://docs.oasis-open.org/wsrf/wsrf-ws_resource-1.2-spec-pr-02.pdf. Version 1.2. 6 October 2005.

[XMLRPC] XML-RPC. http://www.xmlrpc.com/.

[YL05] T. Yu und K. Lin (2005). Service Selection Algorithms for Composing Complex Services with Multiple QoS Constraints. International Conference on Service-Oriented Computing (ICSOC). Seiten 130-143.

[ZCL04] C. Zhou, L.-T. Chia und B.-S. Lee. DAML-QoS Ontology for Web Services (2004). International Conference on Web Services. Seiten 472–479. IEEE Computer Society, 2004.

[ZTP03] O. Zimmermann, M. Tomlinson und S. Peuser (2003). Perspectives on web services. Springer.