Decoupling of Heterogeneous Semantic Service Annotations and Their Semantic Models – Towards an Integrated Approach in a Multi-Service-Provider-Scenario

Jannis Rake and Marten Schönherr

Faculty of Electrical Engineering and Computer Sciences
Technische Universität Berlin
Franklinstr. 28/29, 10587 Berlin, Germany
{Jannis.Rake,Marten.Schoenherr}@sysedv.tu-berlin.de

Abstract. The introduction of Service-Oriented Architectures often promises effective and efficient service support of the organization's business processes. Matching and combining the right services to support the processes can only be ensured, if the service functionalities are semantically annotated. However, service discovery within heterogeneous annotations can become a problem. The following paper introduces an integrated approach to deal with heterogeneous semantic annotations of different service providers. Based on deduction from the state-of-the-art of semantic annotations for services, the developed approach establishes four strategies as the prerequisites for a common search base.

1 Introduction

Bridging the gap between business requirements and IT support has long been a much debated and researched issue. With the dawn of Service-Oriented Architectures (SOA), promising flexible and dynamic support of processes, closing this gap might become easier. However, supporting business processes with exactly the right service becomes a non-trivial issue considering the vast and ever growing number of available services. One solution to deal with the problem of finding exactly the right services is enriching them with meaningful semantic annotations. While there are projects researching the utilization of semantics for business process management like SUPER [1], the following paper will discuss issues and solutions related to dealing with heterogeneity of semantic annotations created by different service providers.

The scenario discussed in this paper is related to a research project initiated by the cooperation of one of the world's leading software providers and a global telecommunication company. Both project partners have a number of services available to be consumed by enterprises and to support their business processes in various ways. However, no consistent structure or method for semantic service annotation exists. Additionally, it seems rather unlikely that future service annotations will be homogeneous, also due to the fact that services may be provided by different departments within the companies. This setting creates a need to evaluate current approaches of semantic annotations and to assess these based on a scenario. An integrated approach

E. Di Nitto and M. Ripeanu (Eds.): ICSOC 2007 Workshops, LNCS 4907, pp. 188–199, 2009.

is chosen to compare different techniques and technologies, and to evaluate possibilities of dealing with the heterogeneity of annotated services. The approach is ongoing work and situated in the scientific domain of design science in information systems research [2]. Current findings are deducted from the analysis of relevant literature and the scenario requirements.

The paper is structured as follows. General terms and definitions, including SOA, Web services and business process support, are introduced in the following section. Once these are established, a closer look is taken at the issues of dealing with heterogeneous service annotations and semantic interoperability with special considerations of ontological solutions (section 2). A state-of-the-art survey of semantic annotation techniques in section 3 discusses possible solutions to enable meaningful business process support by services. The fifth section describes the scenario and motivation for the approach. Thereafter, an analysis of the shortcomings of current solutions leads to the formulation of an integrated approach to deal with heterogeneous service annotations of different providers. Finally, an outlook is given discussing future research issues and possible solution implementations.

2 General Terms and Definitions

Several terms and conditions have to be clarified in the context of this paper. A short introduction to SOA and (Web) services is given to gain a common understanding of both terms and their interrelations. Furthermore, the motivation of service orientation for business process support is introduced. Such a support can only be realized if services can be discovered and assembled into meaningful compositions. For that reason the notions of service discovery and composition are explained briefly. Effective service discovery can only be achieved if meaning can be connected to a service. Therefore, semantic services and knowledge representation via ontologies are clarified in the last paragraph of this section

In literature and practice a multitude of definitions can be found on what a Service-Oriented Architecture is (or what it is not). General agreement can be seen on the rather trivial fact that an SOA relies on services. Services can be seen as software components that provide certain functionalities by exposing one or more interfaces [3]. According to [4], the multitude of definitions originates from several views on an SOA. Therefore, the authors developed a service layer and interactions (SL&I) model. For the remainder of this papers we will focus on the services that support – directly or indirectly – business processes of the SL&I (see Fig. 1).

In order to achieve service-oriented concepts like loose-coupling, interface standardization, encapsulation of logic, autonomy and statelessness [5] many approaches to realize SOAs are based on Web services, e.g. [6]. Furthermore, various concepts, which are relevant for this paper and are rooted in the Semantic Web Community, rely on a large amount of standardized Web service technologies. Therefore, when talking about services in this paper the technical realization of such a service will be based on Web service technology.

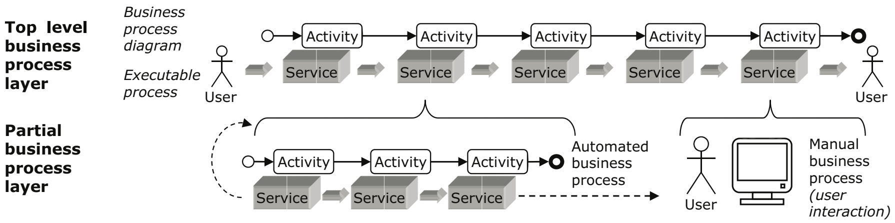
Top level business process layer
Business process diagram
Executable process
User
Activity
Service
User
Partial business process layer
Activity
Service
Automated business process
User
Manual business process (user interaction)

Fig. 1. Partial SL&I model, according to [4]

Many SOA initiatives pursue the goal of supporting business processes. The main reason for this trend seems to be the promise of flexible and dynamic IT support of the enterprise that is challenged by ever changing environments. Business processes can be implemented as (semi-)automatic service compositions. On the technical side, such implementations can be realized using the Business Process Execution Language for Web Services (WS-BPEL). [7, 8]

The right services to support a process have to be discovered. Such a discovery of a (Web) service within a given repository is generally based on a query matching one or more predefined requirements. However, the query may result in zero matches if it cannot find a single service to satisfy the given requirements. A service composition on the other hand could be able to satisfy these very requirements. Finding compositions makes the task of matching requirements to capabilities (often called service matching) even more complex. Service matching has then to consider not only single service requirements but also a large number of possible compositions that may satisfy the requested specifications. [9]

Discovery, matching and composition are only possible if the capabilities of the services are described in a certain way. The syntax of Web services is described by their WSDL (Web Service Description Language) files [10]. However, since the real functionality and meaning of a service cannot be expressed by the WSDL files properly, additional annotations are needed.

The annotation of Web services with semantics is mainly part of the Semantic Web initiative. Generally, such an effort aims at making capabilities and functionalities exposed by the interfaces of services, and maybe even the effects they can have, machine-readable and unambiguous. Exactly such a mark-up or annotation of services would enable automatic or at least semi-automatic discovery, matching and composition of services [11]. A fair amount of semantic service annotation techniques and languages has been developed so far. The fourth section will discuss these efforts in more detail.

Many annotation techniques use ontologies as means to provide a common base of knowledge. Using ontologies, conceptualizations (meaning abstract and simplified views on a certain subject) can be specified explicitly in order to be shared and thus provide a common ground of understanding [12]. Furthermore, they can facilitate semantic interoperability, as shown in the next chapter.

3 Semantic Interoperability

When service providers decide to annotate their services semantically, they will most likely not choose exactly the same terms, concepts and relationships. While infrastructures today mostly provide means to deal with heterogeneous syntax, models for semantic interoperability need to be examined more closely, especially in the context of SOAs.

As can be seen in Fig. 2, different models exist to enable semantic interoperability. Generally, a mapping has to be established between the semantics of the service providers. Two ways of mapping can be distinguished: any-to-any (1,3) and any-to-one (2,4). As a prerequisite of the former, models for mapping each semantic description onto the others have to exist. In the latter case, all models will be mapped on a common instance, generally an ontology. Furthermore, the integration logic has to be chosen. It can either be centralized (1,2) and executed by an integrator (service) or decentralized (3,4) and executed in a peer-to-peer manner. Based on this classification four of the five models in Fig. 2 can be derived. The fifth model is based on the assumption that service providers choose to use a common ontology, making semantic interoperability a quite trivial matter. [13]

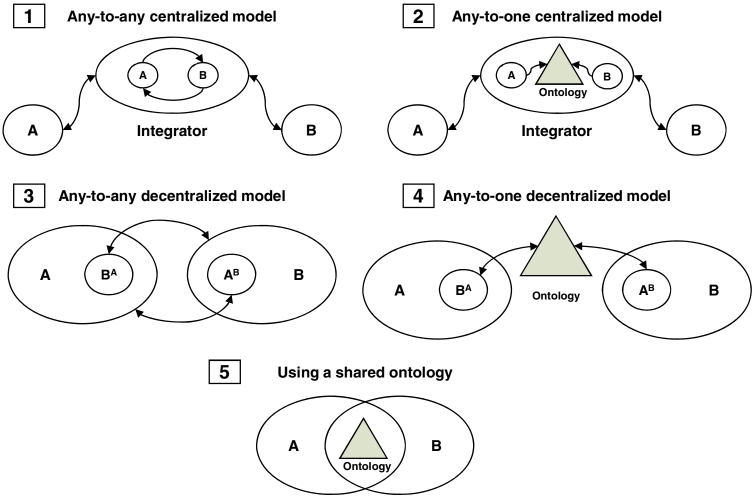
1
Any-to-any centralized model
A
B
A
Integrator
B
2
Any-to-one centralized model
A
B
Ontology
A
Integrator
B
3
Any-to-any decentralized model
A
B^A
A^B
B
4
Any-to-one decentralized model
A
B^A
Ontology
A^B
B
5
Using a shared ontology
A
Ontology
B

Fig. 2. Models of Semantic Interoperability, based on [13]

Each of the described models comes with its advantages and disadvantages. For the project at hand, four models were eliminated. Any-to-any (1,3) solutions do not scale well when increasing the number of service providers beyond two. Although such a model would suffice for the given scenario, future increase of service providers would not be possible. A shared ontology (5) is rather unlikely and highly impracticable when it comes to the effort of creation and maintenance. Leaving two alternatives, a centralized approach (2) is chosen. This decision is based on consistency issues of peer-to-peer networks in (4), when an ontology needs updating or refinement. Also, an SOA should be the perfect environment to provide a service for the integration of the models.

4 Semantic Service Annotations

An overview of semantic annotation techniques, languages and frameworks is needed to enable a full understanding of the problem of dealing with heterogeneity in such an environment. Therefore, a selection to the best knowledge of the authors is presented.

4.1 Web Ontology Language for Services (OWL-S)

The Ontology Web Language for Services has decided to build on the W3C recommendation of OWL (Web Ontology Language [14]). OWL is a language to create and share ontologies in the Semantic Web. Using this language, the developers of OWL-S created an upper ontology for services. It is therefore not a language itself but provides means to describe services and their interactions in various ways.

OWL-S consists of three subontologies, namely profile, process model and grounding. The profile provides information on the functionality of a service to the outside world, among other things aiming at making matches possible. The process model is used to describe how the service works. To access a service, the grounding details information that is normally found in WSDL files on what message formats and protocols to use, among other things. To describe capabilities, OWL-S offers either an ontology describing functionalities in classes (such as a basic taxonomy) or providing generic descriptions of their functions. Such a generic description refers to state transformations of the service. Additionally, both matchmaking and composition could benefit from the definition of inputs, outputs, preconditions and effect that come with the service profiles in OWL-S. However, the grounding of the model adds an additional layer to the actual service execution, such that interactions with e.g. WSDL services can only be accessed using a virtual machine. [15]

4.2 Web Service Modeling Ontology (WSMO)

The Web Service Modeling Ontology is essentially a meta-model for aspects related to semantic Web Services, including definitions for the structure and semantics of meta-data. Based on four concepts, namely ontologies, Web services, goals, and mediators, WSMO has the goal of enabling the automation of Web service integration, including discovery, composition, mediation and execution, among other tasks. All four concepts share a number of functionalities. They can have non-functional requirements, import ontologies and utilize mediators. Every data model in WSMO is represented as an ontology. With regard to semantic annotations, WSMO describes Web services as abstract entities including capabilities and interfaces in additions to the abovementioned elements. [16]

The capabilities of abstract Web service descriptions are characterized primarily by pre- and postconditions as well as assumptions and effects. All of these elements are designed as axioms in WSML (Web Service Modeling Language), allowing special reasoning capabilities. These descriptions allow, in addition to simple keyword searches (which reside more on a syntactic level), simple or complex semantic discovery. [17]

4.3 Semantic Annotations for WSDL and XML Schema (SAWSDL)

SAWSDL is a continuation of the W3C member submission WSDL-S that aims at providing a lightweight approach to add semantics to Web services. It is not a formal description language to express semantic models but rather extends the existing WSDL notation with references to such models. This is done by introducing extensions describing inputs and outputs as well as preconditions and effects. The models referenced by these extensions are maintained outside the WSDL files. Such an ontology agnostic approach allows the use of different ontology languages, e.g. OWL or WSMO. Thus, redundant service descriptions that are introduced in other approaches like OWL-S are being avoided. [18]

Building upon WSDL-S, SAWSDL not only allows semantic model references but also schema mappings within WSDL or XML (eXtensible Markup Language) documents. Schema mappings are realized by lowering and lifting of data, in order to increase their interoperability. Model references can be attached to interfaces, operations and faults within the WSDL files. Based on the introduced annotations, service discovery and composition are facilitated. [19]

4.4 Semantic Web Services Framework (SWSF)

The SWSF is an initiative similar to OWL-S and WSMO, aiming to add machine-readable semantics to Web services. To reach this goal the SWSF consists of two components: the Semantic Web Services Language (SWSL) and the Semantic Web Services Ontology (SWSO). The SWSL introduces two sublanguages namely SWSL-First Order Logic and SWSL-Rules. Based on this language the SWSO presents a conceptual model for describing Web services with a focus on axiomatization. Expressed in the respective sublanguages, a First-Order-Logic (FOL) form and a rules-based form of the ontology are established. In contrast to OWL-S, the FLOWS (First-Order Logic Ontology for Web Services) allows advanced reasoning beyond the decidable description logic of OWL. As compared to WSML, although both languages use similar logical expressions, FLOWS focuses more on the functionality of the rule language while WSML tries to achieve good usability. [20]

4.5 Universal Service-Semantics Description Language (USDL)

Also based on OWL, USDL aims at providing service developers a formal language to specify the semantics of Web services. Similar to the other approaches discussed so far, USDL provides conceptual modeling of Web services allowing search and possibly automated composition of services. Acknowledging the potential of initiatives like OWL-S and WSMO, the authors, however, criticize their impracticality and complexity, especially with respect to the creation of specific domain ontologies to achieve interoperability. Building on WordNet, their goal is to describe services based on a universal ontology. Additionally, they define a set of atomic concepts together with a restricted set of relationships, claiming that any service can be described that way. Interestingly, the problem discussed in this paper is quite similar to the motivation of

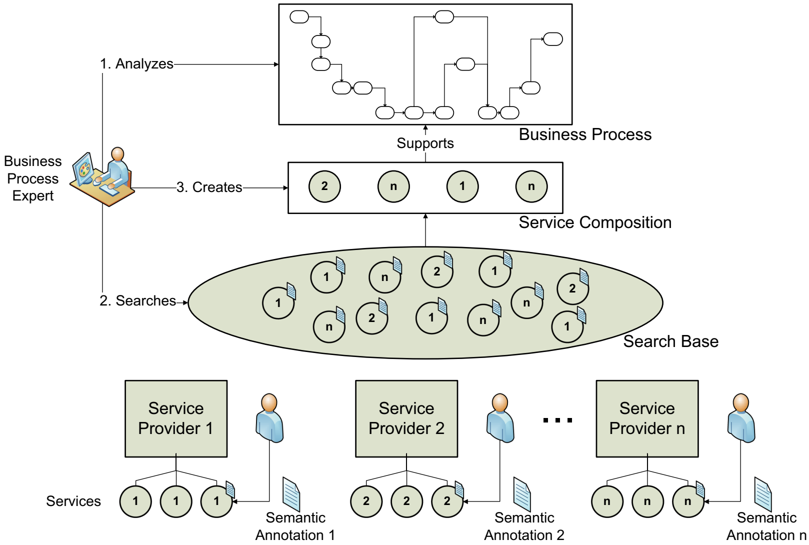

Fig. 3. Heterogeneous semantic service annotations scenario

creating USDL: the impracticality of two or more organizations agreeing on exactly one domain specific ontology. Unfortunately, due to the fact that USDL seems to be in a rather early stage of development, only limited documentation is available to the public. [21]

5 Scenario

This section contains a brief description of the scenario of the project. The scenario establishes requirements and limitations for the chosen approach and is illustrated in Fig. 4.

As mentioned before, two (or more) service providers will be offering services for organizations to support their business processes. Let us assume that all service providers 1 to n are annotating their services semantically. First of all, in order to support a business process, its functionality and goal has to be analyzed. In the scenario this is done by the business process expert (BPE). After analyzing the process the resulting requirements have to be matched with corresponding service functionalities, which is part of the general service discovery [15]. It seems rather unlikely that the BPE knows all the services of all relevant service providers by heart. Therefore, he will probably need to be supported by a tool. However, such a tool needs a common search base to return fitting and non-redundant services based on their semantic annotations. A matchmaking and composition algorithm might even be able to suggest an appropriate service composition to support the whole business process or parts of it. Based on the suggestions of the tool, the BPE can than create or finalize the composition.

In an ideal setting, all these steps would be seamless and efficient. However, organizations generally use different techniques and approaches to annotate their services, as well as different terms and concepts, although services might belong to the same or similar domain. This heterogeneity of their semantic annotation poses a real problem if services from different providers should one day be combinable to seamlessly support a process.

To deal with this problem, an approach is needed that starts before the time of annotation of the services. The fact that providers will presumably not use the same techniques and concepts has to be considered a priori. Motivated by this scenario the next section presents an approach that deals with the heterogeneity of service annotations in an integrated way, covering the many problems that may arise.

6 Towards an Integrated Approach

Many papers that describe implementations for service discovery and matching choose one annotation technique beforehand and design their matchmaking to deal with this annotation only, e.g. [22], [23] or [17].

Sometimes outlooks promise the extension of approaches for several annotation languages or techniques, but to our knowledge, no such approach actually exists that deals with two different annotations from the start. One reason for choosing only to use one annotation technique in the beginning may be, that approaches like WSMO or OWL-S provide a rather holistic approach, that might not be easily translated.

Based on the assessment of the current solutions and the presented scenario, we can devise a two-fold problem:

- Semantic annotations are expressed in different languages, and
- Semantic annotations may use different terms, axioms and concepts, although they exist in the same domain.

Although both parts of this problem have received more or less detailed attention in research, an integrated solution to this problem does not seem to exist. Problems may also arise, when only one annotation approach is used, but it leaves room for interpretation of its use. Therefore formal models are a prerequisite. [24]

In order to solve the abovementioned problem, an integrated approach was developed. It is divided into four strategies, combining existing solutions to achieve higher interoperability. Each one of the strategies depicted in Fig. 4 is explained in the following.

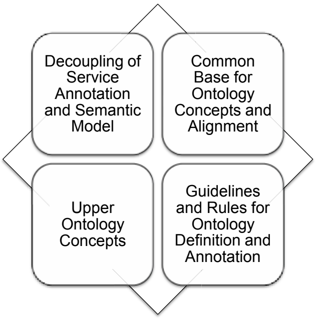
Decoupling of Service Annotation and Semantic Model
Common Base for Ontology Concepts and Alignment
Upper Ontology Concepts
Guidelines and Rules for Ontology Definition and Annotation

Fig. 4. The integrated approach

6.1 Decoupling of Service Annotation and Semantic Model

As described earlier, prevalent approaches like OWL-S and WSMO introduce redundant information for description already covered in WSDL documents. This basically adds an additional layer to the service execution, since the semantic models have to be grounded in WSDL first. Furthermore, once changes are made to the service, the descriptions have to be updated in both realms. Interoperability is inhibited, since the grounding of the semantic service description is not standardized.

Based on these issues, the integrated approach will decouple the semantic models (mostly ontologies) from the service annotation in the WSDL (the service grounding). The most promising technique to do so seems to be the SAWSDL [19] extension. Not only has the concept of decoupling provided useful in other interoperability solutions, it also fits well with principles of service orientation.

6.2 Common Base for Ontology Concepts and Alignment

It is possible that ontologies used by the service providers are represented in different languages and/ or structures. This barrier can be resolved in various ways, depending on the difference between the ontologies. One way of creating such a base is to represent all in their “greatest common denominator”, choosing the most powerful language to express all relevant concepts and axioms [25]. The representation of the ontologies in an abstract syntax, as has been done with OWL already [26], would additionally facilitate the ontology translation and alignment. It is also thinkable to use an ontology meta-model approach to support translation, e.g. as suggested by the Ontology Definition Metamodel (ODF) which is currently being finalized by the Object Management Group [27].

It may of course be possible to combine some of these approaches to yield better results for ontology translation. The option of creating an own translation mechanism should only be considered as a last resort if existing approaches do not, even after possible adjustments, meet the requirements of the scenario. This is due to the fact that the development of such a mechanism is highly complex and would probably go beyond the scope of the project.

The translation of the description logic, however, does not solve, that ontologies created by different parties differ sometimes quite largely in the terms, concepts and axioms. One simple example is the existence of homonyms, where two terms express two completely different concepts, e.g. a bow can both be a weapon or a tied ribbon. To match and align ontologies, certain algorithms can be used, some of which have already been manifested in software tools, e.g. COMA++ [28].

6.3 Upper Ontology Concepts

The definition of upper ontology concepts is already integrated in most annotation techniques. Nevertheless, most of them do not explicitly consider the heterogeneity of annotations, as discussed earlier. One exception is the use of mediators in WSMO. In the context of heterogeneous services it might proof useful to extend the existing concepts to facilitate comparison and matching of their annotations. Examples could be the definition of domain properties of the service annotation to facilitate alignments or predefine axioms which provide reasoning based on the origin (company/ organization) of the service.

6.4 Guidelines and Rules for Ontology Definition and Annotation

Certain conditions hold in the scenario of (cooperating) organizations that do not exist in the open Semantic Web. Organizations may be able to agree on certain specifics before annotating their services, including guidelines and rules for defining ontologies and annotating their services. Thus, an integrated approach can influence their consensus by giving rules and guidelines for their negotiations, as well as providing the abovementioned upper ontology concepts. Such an approach should define certain guidelines that will, provided the organizations follow it consistently, make the discovery in heterogeneous environments much easier. In this sense, a method is needed following the principles of robust design [29], in order to avoid variances in distributed ontology creation.

Such guidelines and rules could include limitations of the chosen annotation technique, e.g. limiting the number of axioms or using only predefined relations between concepts. Thus, the consolidation of service annotations for matching and composition can be based on the common preconditions set by these guidelines and rules, easing the alignment of heterogeneous annotations.

7 Conclusion and Further Research

In this paper we have introduced the necessity to deal with the interoperability of heterogeneous service annotations of different service providers. By assessing several models of semantic interoperability it became clear, that a certain, ontology centered solution provides the fitting model for the problem at hand. An introduction and examination of the state-of-the-art of semantic service annotations followed to aid the comprehension of dealing with heterogeneity in such an environment. The scenario section introduced the motivation for an approach dealing with the differences of service annotations, stating that eventually such a solution would allow semi-automatic discovery and combination of services. Finally, an assessment of the requirements for such a scenario leads to the development of an integrated approach. The approach consists of four strategies to ensure the interoperability of semantic service annotations. First of all, the semantic models have to be decoupled from the syntactical service description. Additionally, a common base for the ontology concepts of the annotations has to be established. Such a base allows the alignment of two or more ontologies, making the combined functionalities and additional information searchable. The definition of additional upper ontology concepts should facilitate the creation of ontology-based annotations while ensuring future interoperability with other services. Finally, rules and guidelines for the definition of ontologies and the annotation procedures are needed to limit the amount of variances in distributed environments.

In the future the introduced approach will be refined and tested. For this purpose and to demonstrate the approach's feasibility a prototype will be implemented. Based on the prototype it will be possible to compare the efficiency and effectiveness of the four strategies. Different variations of the strategies can thus be evaluated and an optimal method can be derived.

References

1. SUPER. Semantics Utilised for Process Management within and between EnterpRises (2007) (retrieved from, 01.08.2007), `http://www.ip-super.org/`
2. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28(1), 75–105 (2004)
3. Krafzig, D., Banke, K., Slama, D.: Enterprise SOA - Service-Oriented Architecture Best Practices. Prentice Hall Professional Technical Reference, Upper Saddle River (2005)
4. Offermann, P., Schröpfer, C., Holschke, O., Schönherr, M.: SOA: The IT-Architecture behind Service-Orientation. In: Steffens, U., Addicks, J.S., Streekmann, N. (eds.) MDD, SOA und IT-Management, pp. 1–11. GITO, Berlin (2007)
5. Erl, T.: Service-Oriented Architecture - Concepts, Technology, and Design. Prentice Hall Professional Technical Reference, Upper Saddle River (2005)
6. Newcomer, E., Lomow, G.: Understanding SOA with Web Services. Addison-Wesley, Reading (2005)
7. Organization for the Advancement of Structured Information Standards (OASIS), Web Services Business Process Execution Language Version 2.0. (2007), `http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf`
8. Khalaf, R., Keller, A., Leymann, F.: Business processes for Web Services: Principles and applications. IBM Systems Journal 45(2), 425–446 (2006)
9. Kona, S., Bansal, A., Gupta, G., Hite, T.: Semantics-based Efficient Web Service Discovery and Composition, Department of Computer Science - University of Texas at Dallas (2007)
10. Christensen, E., Curbera, F., Meredith, G., Weerawarana, S.: Web Services Description Language (WSDL) 1.1 (2001) (retrieved from, 02.08.07), `http://www.w3.org/TR/wsdl`
11. McIlraith, S.A., Son, T.C., Zeng, T.C.H.: Semantic Web services. IEEE Intelligent Systems 16(2), 46–53 (2001)
12. Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge Acquisition 5(2), 199–220 (1993)
13. Vetere, G., Lenzerini, M.: Models for semantic interoperability in service-oriented architectures. IBM Systems Journal 44(4), 887–903 (2005)
14. McGuinness, D.L., Harmelen, F.v.: OWL Web Ontology Language Overview (2004) (retrieved from, 28.07.07), `http://www.w3.org/TR/owl-features/`
15. Martin, D., Paolucci, M., McIlraith, S., Burstein, M., McDermott, D., McGuinness, D., Parsia, B., Payne, T., Sabou, M., Solanki, M., Srinivasan, N., Sycara, K.: Bringing Semantics to Web Services: The OWL-S Approach. In: Cardoso, J., Sheth, A.P. (eds.) SWSWPC 2004. LNCS, vol. 3387, pp. 26–42. Springer, Heidelberg (2005)
16. Roman, D., Keller, U., Lausen, H., Bruijn, J.d., Lara, R., Stollberg, M., Polleres, A., Feier, C., Bussler, C., Fensel, D.: Web Service Modeling Ontology. Applied Ontology 1(1), 77–106 (2005)
17. Keller, U., Lara, R., Lausen, H., Fensel, D.: Semantic Web Service Discovery in the WSMO Framework. In: Cardoso, J. (ed.) Semantic Web Services: Theory, Tools and Applications. IGI Publishing (2007)

18. Akkiraju, R., Farrell, J., Miller, J., Nagarajan, M., Schmidt, M.-T., Sheth, A., Verma, K.: Web Service Semantics - WSDL-S (2005) (retrieved from, 28.07.2007), `http://www.w3.org/Submission/WSDL-S/`
19. Farrell, J., Lausen, H.: Semantic Annotations for WSDL and XML Schema (2007) (retrieved from, 01.08.2007), `http://www.w3.org/TR/sawsdl/`
20. Battle, S., Bernstein, A., Boley, H., Grosof, B., Gruninger, M., Hull, R., Kifer, M., Martin, D., McIlraith, S., McGuinness, D., Su, J., Tabet, S.: Semantic Web Services Framework (SWSF) Overview (2005) (retrieved from, 02.08.2007), `http://www.daml.org/services/swsf/1.0/overview/`
21. Bansal, A., Kona, S., Simon, L., Hite, T.D.: A Universal Service-Semantics Description Language. In: Proceedings of the Third IEEE European Conference on Web Services (ECOWS), pp. 214–225 (2005)
22. Pathak, J., Koul, N., Caragea, D., Honavar, V.: A Framework for Semantic Web Services Discovery. In: Proceedings of the 7th ACM International Workshop on Web Information and Data Management, WIDM (2005)
23. Kaufer, F., Klusch, M.: WSMO-MX: a logic programming based hybrid service matchmaker. In: Proceedings of the 4th IEEE European Conference on Web Services (ECOWS 2006), Zürich, Switzerland, pp. 161–170 (2006)
24. Wang, H.H., Gibbins, N., Payne, T., Saleh, A., Sun, J.: A Formal Semantic Model of the Semantic Web Service Ontology (WSMO). In: Proceedings of Twelfth IEEE International Conference on Engineering of Complex Computer Systems, Auckland, New Zealand (2007)
25. Keller, U., Feier, C., Steinmetz, N., Lausen, H.: Report on reasoning techniques and prototype implementation for the WSML-Core and WSMO-DL languages, Digital Enterprise Research Institute (2006)
26. World Wide Web Consortium. OWL Web Ontology Language Semantics and Abstract Syntax (2004) (retrieved from, 27.07.2007), `http://www.w3.org/TR/owl-semantics/ syntax.html`
27. OMG. Ontology Definition Metamodel (2006) (retrieved from, 01.08.07), `http://www.omg.org/docs/ptc/06-10-11.pdf`
28. Aumueller, D., Do, H.H., Massmann, S., Rahm, E.: Schema and ontology matching with COMA++. In: Proceedings of the 2005 ACM SIGMOD international conference on Management of data, pp. 906–908 (2005)
29. Arvidsson, M., Gremyr, I.: Principles of robust design methodology. Quality and Reliability Engineering International (published, 2007)