Towards Context-based Mediation for Semantic Web Services Composition

Michael Mrissa, Chirine Ghedira and Djamal Benslimane
Université Claude Bernard Lyon 1
Lyon, France
{firstname.surname}@liris.cnrs.fr

Zakaria Maamar
Zayed University
Dubai, United Arab Emirates
zakaria.maamar@zu.ac.ae

Abstract

Web services composition is a keystone towards the development of interoperable systems. However, despite several efforts for explicit semantic description, Web services still face data-heterogeneity challenges. Correct interpretation of data exchanged between composed Web services is hampered by different implicit modeling assumptions and representations. In this paper, we demonstrate the importance of context to facilitate data exchange, and describe a context representation model for Web services. Then, we present a context-based mediation architecture that helps performing meaningful composition.

1. Introduction

In the domain of service-oriented computing, Web services[1] are now accepted as a standard means for enabling composition scenarios, and their capacity for loosely-coupled interactions allows answering to complex user needs. Despite these advantages, the standard Web services protocol-stack[2] was not initially planned for satisfying the requirements of semantic interoperability. Recent languages and frameworks such as OWL-S [12], the Web Services Modeling Ontology WSMO [1] or WSDL-S [13], explicitly describe the semantics of Web services, then referred to as semantic Web services [7]. These initiatives use ontologies[3] as shared vocabularies to facilitate semantic reconciliation.

However, such approaches do not take the context of information into consideration. The term context is defined in the rest of this paper as the collection of implicit assumptions that are required in order to perform accurate data interpretation. A semantic concept should be interpreted differently, depending on the context it relates to. Consequently, semantic description and interpretation mechanisms need consider context in order to interpret information efficiently. In the domain of Web services composition, context interpretation is generally buried in code, and its explicit descriptions are inexistent. Changes in context are dealt with manually, which requires lots of time and handmade work, and reduces availability and reliability of information systems.

In this paper, we provide a framework for explicitly describing and managing context in order to enable meaningful data exchange between Web services that participate in a composition. Our contribution consists of 1) a proposition of context representation model for Web services, followed by 2) a method for annotating WSDL input and output messages with context, and 3) a service-based mechanism for context-aware mediation in a composition. This paper is organized as follows. In Sect. 2, we show, with a working example, the value added by managing the context of data in Web service composition. In Sect. 3, we present the notion of *semantic object*, introduced in [2], that supports our work, prior to showing how we adapt the associated model to the domain of Web services. In Sect. 4, we detail our annotation for integrating context representation into the Web service protocol stack, and present a context and service-based mediation architecture for Web services composition. In Sect. 5 we overview related work on mediation and semantics for Web services, and context representation. Finally, Sect. 6 concludes the paper and sets guidelines for future work.

2. Motivating Example

We demonstrate, with a simple booking example, how context changes the interpretation of information that flows between Web services. The example concerns a trip to Japan. A hotel, which has attractive rates, provides a Web service endpoint described with the usual WSDL file, for booking rate estimate. In order to estimate the affordability

[1] A Web Service is a software component that is described and being accessed via standard XML-based protocols.

[2] Simple Object Access Protocol SOAP [3], Web Service Description Language WSDL [5] and Universal Description, Discovery and Integration protocol UDDI [16].

[3] An ontology is a shared description of a domain knowledge [11].

of this hotel for a European passenger, the following composition of Web services occurs: *hotel booking* WS_1 to calculate charges based on the number of booked nights, and *banking* WS_2 to manage payment.

From a technical perspective, WS_1 sends a parameter[4] named "price_yen", and WS_2 receives a parameter named "price_euros". Although it is possible to use different type systems, we consider, for illustration purposes, both parameters are described in XML Schema type system [17], and are of type "double". This information, obtained from WSDL description, shows low level data compatibility between Web services.

Also, these parameters bind to particular semantics. WS_1 delivers a value that should be interpreted as a figure in Yens, whereas WS_2 expects a value that it will interpret as a figure in Euros. We assume both parameters bind to the same "Price" semantic concept, described in a common ontology. Existing approaches to semantic description and mediation of Web services, overviewed in Sect. 5, explicitly describe this information, to allow inferring a relation between these parameters, and performing appropriate conversion. Such approaches use ontologies as references to solve structural and semantic heterogeneities.

Now, let us inject context into these parameters. WS_1 binds to a "Japanese Hotel Booking" context, in which charges have a scale factor of 1000, prices do not include Value-Added Tax (VAT), dates required for conversion rates are in Japanese format (yyyy.mm.dd); and WS_2 binds to a "French Banking" context, where charges have a scale factor of 1, prices include Value-Added Tax, dates required for conversion rates are in French format (dd.mm.yyyy). This example shows that heterogeneous contexts exist too, so an agreement on the interpretation of values must be reached, by reconciling context descriptions. Context information is related to the local and implicit assumptions on the interpretation of data, and is not described in the domain ontology. Usually, in a semantic composition, context heterogeneities

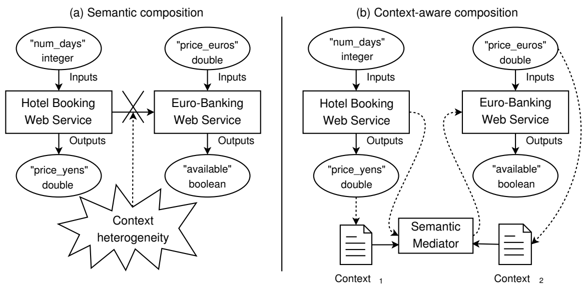

Figure 1. Semantic vs. Context-aware composition

[4] We refer to parameter as the parts of messages described in WSDL

are resolved in an ad-hoc way at the level of the receiver Web-service, if at all, as presented in Fig. 1-(a). This reduces adaptability of Web services as parts of composition, and furthermore, it gives them the responsibility of solving context heterogeneities. To conduct context-aware composition, the context of data must be explicitly described, and a mediation mechanism must handle data flow, as shown in Fig. 1-(b). As standard semantic Web services do not handle such representation, we hence propose an annotation of WSDL to explicitly describe context of data, on the basis of the following model.

3. Context Representation

In this section, we study a model for describing the semantics of semi-structured data [2]. Then, we adapt this model to facilitate context description for data exchange in Web service composition.

3.1. Definition of a Semantic Object

Based on Sciore et al.'s notion of *semantic value* [14], Bornhövd defines a model for describing the underlying semantics of semi-structured data, with the concept of *semantic object* [2].

An ontology Φ is defined as a vocabulary for the notation and representation of a given application domain, and formalized as a finite set of concepts:

$$\Phi = \{C_1, \ldots, C_n\}, n \in \mathbb{N} \ .$$

Each concept C_i has a physical representation type $RepType(C_i)$ associated with it, to which the domain $Dom(RepType(C_i))$ specifies the possible values for the representation of data corresponding to this concept. Also, a semantic object $SemObj$ is defined as a triple:

$$SemObj = < C, v, \$ >,$$

Where $C \in \Phi$ denotes the ontology concept to which the semantic object adheres, and the value $v \in Dom(RepType(C))$ is the physical representation of v according to the physical representation of type C. $\$$ specifies the context of $SemObj$, and is defined as:

$$\$ = \{< C_1, v_1, \$_1 >, \ldots, < C_k, v_k, \$_k >\}, k \in \mathbb{N} \ .$$

where $< C_k, v_k, \$_k >, 1 \leq i \leq k$, are semantic objects, referred to as "modifiers" in this paper, that describe different semantic aspects of $SemObj$. In turn, modifiers may provide additional context information in $\$_k$. Are also introduced the notion of complex semantic object, the means for semantic conversion, comparability and equivalence of semantic objects, that build the basis of this framework.

3.2. A Context Model for Web services

The model described previously provides a sound basis for describing messages parts of Web services as semantic objects, its main advantage being the combination of intensional and extensional descriptions. The concept of a semantic object is intensionally described in a domain ontology, while context is extensionally described under the form of additional attributes, preventing an excessive complexity of the domain ontology.

Bornhövd's model was designed for handling semi-structured data in a large sense. However, in the domain of semantic Web services, a major design choice to semantic description is the separation of the grounding and abstract views on data. Generally, the grounding representation of data follows the XML Schema type system, while its abstract part is described in some ontology language. Such separation of concerns requires an explicit description of the physical representation of semantic objects. Therefore, we deem appropriate to include the physical representation in the definition of semantic object, as follows:

$$SemObj = < C, v, t, \$ >,$$

where $C \in \Phi$ denotes the ontology concept to which the semantic object $SemObj$ adheres, the value $v \in Dom(t)$ is the physical representation of v according to t, that specifies the type of v defined in a specific type system, and $\$$ specifies the context of $SemObj$. Such definition does not modify previous work on the model, however, it clarifies the definition of data type, which is now clearly distinguished from the conceptual reference C of the semantic object. Then, existing mediation approaches presented in Sect. 5, that also deal with low level descriptions of data types, can be combined to the context-based mediation architecture presented in the following. In the example of Sect. 2, a possible semantic object would be:

<ns:Price, 55.00, xsd:double, Context>

where ns:Price is a qualified name referring to the concept of price described in a domain ontology, 55.00 is the value sent by the Web service, xsd:double is the type in XML Schema, and Context is a list of semantic objects that describe implicit semantic aspects such as VAT rate and scale factor.

Also, Bornhövd adds the vocabulary for describing modifiers to domain ontologies, while stating that a context description is always a subset of all the meaningful aspects of an ontology concept, which are potentially infinite. However, Web service providers should be free to decide which subset of modifiers is relevant to their application. In which case, the size of domain ontologies would grow depending on providers' needs for context description. To overcome this problem, we deem appropriate to separate context ontologies from domain ontologies, so that they do not surcharge the latter. Context ontologies should describe all the modifiers Web service providers associate to a concept.

As context ontologies specify structural representation and meaning of terms, we need to define how to extensionally describe context values for Web services. To do so, we define two categories of modifiers. Static modifiers have to be (statically) specified to clarify the meaning of data, and dynamic modifiers can be (dynamically) determined from other parts of the semantic object. We add static modifiers to the WSDL, so that our approach is compliant with the standard Web services protocol stack. We do not need to specify dynamic modifiers, as they can be inferred from static ones. In the next section, we present a solution for annotating descriptions of composed Web services, in order to make context information available at the execution stage of composition, before describing a rule-based solution for context mediation in a composition.

4. Context Management

4.1. Annotating WSDL with Context

Based on the model developed previously, we enrich the description of Web services with context information. Using the WSDL extensible elements, we annotate WSDL message parts, so that they can be described as *semantic objects*. WSDL depicts how to access a Web service and what operations it performs. Access is subject to specific communication protocols, and operations use input and output arguments that have a number, an order, and a type. To keep the paper self-contained, we overview a simplified structure of the WSDL metamodel in Fig. 2.

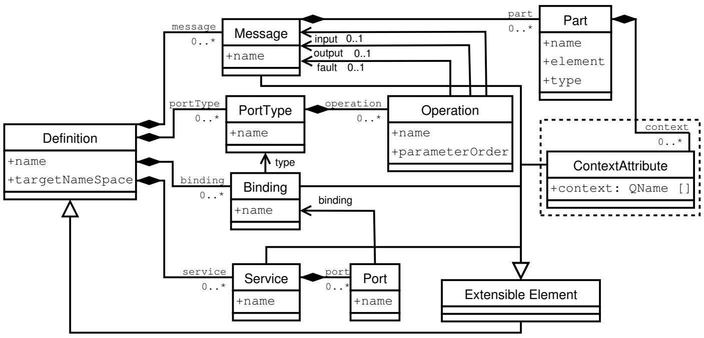

Figure 2. Context in WSDL metamodel

Our annotation takes advantage of the extensibility elements proposed in the WSDL [5], so annotated descriptions can operate seamlessly with classical or annotation-aware clients. As `<message>` elements are composed of one or several parts, we annotate `<part>` elements with a `context` attribute that contains the values of static modifiers. Each `context` attribute contains a list of qualified names. The first qualified name of the list specifies the ontology concept of the value (C). Following elements refer to instances (called individuals in the OWL vocabulary) of static modifiers described in a context ontology. Thus, with our annotation, a value v, and its data type t described in WSDL are enhanced with the concept C and the modifiers necessary to define the context $\$$, thus forming a semantic object $< C, v, t, \$ >$ as defined previously. A sample annotated WSDL file is shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="http://localhost:8080/axis/EuroBanking.jws"
 …
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
 xmlns:ctxt="http://www710.univ-lyon1.fr/ mmrissa/context/context.xsd"
 xmlns:ctxt1="http://domain.ontology.org/Price.owl"
 xmlns:ctxt2="http://context.ontology.org/context/PriceContext.owl#">
 ...
 <wsdl:message name="checkPriceRequest">
  <wsdl:part name="price" type="xsd:double" ctxt:context="ctxt1:Price
ctxt2:France ctxt2:VATIncluded ctxt2:ScaleFactorOne"/>
</wsdl:message>
 …
 <wsdl:portType name="EuroBanking">
  <wsdl:operation name="checkPrice" parameterOrder="price">
  <wsdl:input name="checkPriceRequest" message="impl:checkPriceRequest"/>
  <wsdl:output name="checkPriceResponse"
message="impl:checkPriceResponse"/>
 </wsdl:operation>
</wsdl:portType>
 …
</wsdl:definitions>
```

To obtain a complete context $\$$, values of dynamic modifiers are inferred at runtime by rule-based mechanisms. This comes with several advantages: rules are easily modifiable, so our architecture is more adaptable to changes in the underlying semantics. Also, often-changing modifiers could not be statically stored, so using rules simplifies the annotation to WSDL. Furthermore, rules separate application logic from the rest of the system, so updating rules does not require rewriting application code. In the following, we detail our context mediation architecture, that integrates into composition as a Web service, and show its interactions with a rule-based component (a rule engine).

4.2. Context Mediation for Web Services

As a first step, we examine how context management capabilities should be granted to composition scenarios. Based on previous works (Sect. 5), we follow a decoupled approach, and we encapsulate context mediation functionalities into a Web service front-end. This solution presents two main advantages. Firstly, the mediation Web service can be triggered, via its WSDL interface, so it remains independent from composition languages and engines. Secondly, it is straightforward to handle context by invoking the mediation Web service between every two composed Web services. The main problematic aspect of this solution is its limited scope, as data flow is bound to data types specified in WSDL descriptions. However, it is possible to generate adapted WSDL interfaces at design time.

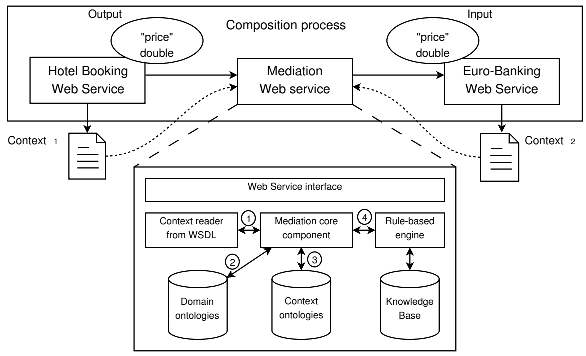

Figure 3. Functioning of the mediation Web service in the sample composition

Now, let us detail the operation of the mediation Web service. Its role is to convert data from the context of the Web service it originates (source context), to the context of the Web service it is being sent to (target context). For each message part, the role of the mediation Web service consists in 1) building source and target contexts using WSDL annotation, ontologies and rules; 2) examining differences between contexts and performing data conversion to target context, or generating an error message, and 3) sending results to the appropriate target. As Fig. 3 shows, the mediation service communicates with four components: a context reader to extract context extensibility attributes from WSDL descriptions, context and domain ontologies to identify concepts, and a rule engine to determine values of dynamic modifiers and to perform the conversion.

For each message part concerned with the mediation process, the mediation Web service performs the following steps. First step, it generates an in-memory model of WSDL descriptions, extracts context annotations and builds contexts of the message part. Second step, it identifies the first qualified name of the annotation as the ontology concept of the parameter. Then, it verifies that the concepts of both annotations match. Third step, the mediation Web service extracts contexts from the context ontologies referred to in the WSDL annotation. Contexts of semantic objects are the sets of subsuming semantic objects described in the context ontology. Additional annotation attributes help specify the values of static modifiers. Fourth step consists of two stages. Stage one, the rule engine infers the values of dynamic modifiers from the values of static modifiers of the annotation. Stage two, for each modifier, the rule engine applies appropriate conversion to the parameter, in order to convert the value of the modifier to the target context. If an error happens at some point of the process, an exception is thrown, and the mediation Web service sends a failure message. If the mediation process was performed correctly, data is converted to the target context and transmitted to the next Web service.

A prototype has been fully developed as a proof-of-concept to demonstrate the feasibility of this architecture under the Java^{TM} NetBeans environment. We developed a graphical user interface to read and write context annotations from and to WSDL files. This tool enables providers or advanced users to annotate WSDL files with context, so that it becomes possible to compose them with context-aware mediation Web services. We also developed a mediation Web service, that reads context annotation from WSDL files and converts data received from its source context to a target context. Our implementation performs dynamic and accurate context interpretation, and enables meaningful execution of composition. To illustrate our proposal, we implemented the example of Sect. 2. With annotated WSDL and the mediation Web service, not only the "Price" concepts match, but data is transformed at-runtime, to comply with the different scale factors, heterogeneous date formats that allow getting up-to-date conversion rates between currencies, and different VAT rates that also are not always included in the price.

Our current composition example runs into a BPWS4J (http://www.alphaworks.ibm.com/tech/bpws4j) composition engine hosted in a Apache Axis container (http://ws.apache.org/axis). We use Jena 2 (http://jena.sourceforge.net/) API and a Drools (http://www.drools.org/) rule engine, to access and manipulate OWL ontologies, infer values that modifiers should take and perform data conversion at runtime. Our prototype includes domain and context ontologies designed with Protégé (http://protege.stanford.edu/) for describing the "Price" concept and context[5].

5. Related Work

This section presents different initiatives that relate to the semantic and mediation aspects of Web services, and to previous work on context description. These related works helped us build ideas, and backed our approach as they are important references of the domain.

5.1. Semantics for Web services

Semantic Web services constitute an active domain of research. Most approaches rely on ontologies to express the semantics of a domain, however, there are several ways of inserting semantics into Web Services. One way involves using description languages like OWL-S [12], and another way consists in extending syntactic standards like WSDL with semantic features (WSDL-S) [13].

[5] Available at http://www710.univ-lyon1.fr/∼mmrissa/price.owl and http://www710.univ-lyon1.fr/∼mmrissa/PriceContext.owl

OWL-S [12] is a subset of the OWL (ex-DAML) ontology language. It is a general ontology that provides support for building semantic Web services. OWL-S was designed to be coupled with standard description formats like WSDL. It consists of three ontologies, namely *process model*, *service profile* and *grounding*, that respectively describe *"how the service works"*, *"what the service does"* and *"how to access the service"*. Inspired from OWL-S, several research projects have been developed, such as ODESWS [6] that models Web services using problem-solving methods.

From the DERI laboratory, WSMO [1] is a formal language and ontology that describes varied aspects of semantic Web services. It supports the development and description of semantic Web services with the WSMF [7] conceptual model, that enables mediation as a service, so that it allows maximal decoupling between component Web services.

With WSDL-S, Miller et al. annotate WSDL with several extensions related to operations and messages [13]. These extensions refer to concepts of domain models to specify semantics of messages, but also preconditions and effects of operations.

5.2. Mediation between Web services

Mediation between Web services is a hot topic and receives a lot of attention from the research community. Many mediation approaches rely on the concept of mediator for solving data heterogeneities between participants of an interaction.

Cabral and Domingue [4] provide a broker-based mediation framework for composing semantic Web services. Their approach follows WSMO conceptual framework [1] that recommends using strongly decoupled and service-based mediation capacities. The mediator component is a key part of their architecture and it mediates concepts between different ontologies.Williams et al. [18] use agents to perform semantic mediation between input and output parameters of Web services by encapsulating the composition into an agent, that controls the developpement of the operation. Spencer et al. [15] present a rule-based approach to semantically match outputs and inputs of Web services. An inference engine analyzes OWL-S descriptions and generates multiple data transformation rules using a description-logic reasoning system.

5.3. Approaches to Context Description

In many works, context refers to the interactions with the surrounding environment. In this paper we follow another definition and only detail similar work for space limitation purpose. In the domain of database interoperability, the Context Interchange approach [9] provides formalisms for context representation, based on the notion of semantic value. It has proved to be a highly scalable, extensible and adaptable approach to semantic reconciliation of data, and was first introduced by Sciore et al. [14]. Goh [10] and Firat [8] presented implementations and extensions to this approach. Then, Bornhövd [2] adapted this model to the description of semi-structured data. We use this latter work as a basis to our approach.

While mediation and semantic description of Web services in a composition are very active research fields, to the best of our knowledge, none of these works actually consider the use of explicit context description to solve heterogeneities of type value in Web services composition.

6. Conclusion

In this paper we highlighted the importance of context in Web services composition, and proposed an architecture for handling context, in order to obtain meaningful composition scenarios. We first demonstrate, with an illustrative example, that context is required for a correct interpretation of data exchanged between Web services. Then, we provide a solution for describing and handling such information, using context ontologies and rule-based mechanisms. This proposal includes a model for context description, coupled with a method for annotating WSDL parameters, propelling them to the level of *semantic objects*, while respecting WSDL extensibility elements; and a service-based context mediation architecture.

Future improvements of context-based mediation in Web service composition targets three aspects: first, the study of methods for enabling the integration of the mediation Web service into dynamic composition; second, the addition of capabilities to the mediation mechanism, like the ability to compose Web services specialized in conversion and to integrate them in the mediation process; and third the extension of this architecture to Web service discovery and selection stages, that could consider the possibilities of context mediation as a selection criteria in the discovery step.

References

[1] S. Arroyo and M. Stollberg. WSMO Primer. WSMO Deliverable D3.1, DERI Working Draft. Technical report, WSMO, 2004. http://www.wsmo.org/2004/d3/d3.1/.

[2] C. Bornhovd. Semantic metadata for the integration of web-based data for electronic commerce. In *Int'l Workshop on E-Commerce and Web-based Information Systems (WECWIS), Santa Clara, CA*, pages 137–145, 1999.

[3] D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. F. Nielsen, S. Thatte, and D. Winer. Simple object access protocol (SOAP) 1.1. Technical report, The World Wide Web Consortium (W3C), 2000.

[4] L. Cabral and J. Domingue. Mediation of semantic web services in irs-iii. In *First Int'l Workshop on Mediation in Semantic Web Services (MEDIATE 2005), Amsterdam, The Netherlands*, December 12th 2005.

[5] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana. Web Services Description Language (WSDL) 1.1, W3C Note. Technical report, The World Wide Web Consortium (W3C), March 2001.

[6] O. Corcho, A. Gomez-Perez, M. Fernandez-Lopez, and M. Lama. Ode-sws: A semantic web service development environment. In I. F. Cruz, V. Kashyap, S. Decker, and R. Eckstein, editors, *SWDB*, pages 203–216, 2003.

[7] D. Fensel and C. Bussler. The Web Service Modeling Framework WSMF. Technical report, Vrije Universiteit Amsterdam, 2002.

[8] A. Firat. *Information Integration Using Contextual Knowledge and Ontology Merging*. PhD thesis, Massachusetts Institute of Technology, Sloan School of Management, 2003.

[9] C. H. Goh, S. Bressan, S. Madnick, and M. Siegel. Context interchange: new features and formalisms for the intelligent integration of information. *ACM Trans. Inf. Syst.*, 17(3):270–293, 1999.

[10] C. H. Goh, S. Bressan, S. E. Madnick, and M. Siegel. Context interchange: New features and formalisms for the intelligent integration of information. *ACM Trans. Inf. Syst.*, 17(3):270–293, 1999.

[11] T. Gruber. What is an ontology? http://www-ksl.stanford.edu/kst/what-is-an-ontology.html, 2000.

[12] D. L. Martin, M. Paolucci, S. A. McIlraith, M. H. Burstein, D. V. McDermott, D. L. McGuinness, B. Parsia, T. R. Payne, M. Sabou, M. Solanki, N. Srinivasan, and K. P. Sycara. Bringing Semantics to Web Services: The OWL-S Approach. In J. Cardoso and A. P. Sheth, editors, *SWSWPC*, volume 3387 of *Lecture Notes in Computer Science*, pages 26–42. Springer, 2004.

[13] J. Miller, K. Verma, P. Rajasekaran, A. Sheth, R. Aggarwal, and K. Sivashanmugam. WSDL-S: Adding Semantics to WSDL - White Paper. Technical report, Large Scale Distributed Information Systems, 2004. http://lsdis.cs.uga.edu/library/download/wsdl-s.pdf.

[14] E. Sciore, M. Siegel, and A. Rosenthal. Using semantic values to facilitate interoperability among heterogeneous information systems. *ACM Trans. Database Syst.*, 19(2):254–290, 1994.

[15] B. Spencer and S. Liu. Inferring data transformation rules to integrate semantic web services. In S. A. McIlraith, D. Plexousakis, and F. van Harmelen, editors, *International Semantic Web Conference*, volume 3298 of *Lecture Notes in Computer Science*, pages 456–470. Springer, 2004.

[16] UDDI. *Universal Description, Discovery, and Integration of Business for the Web*, Oct. 2001. URL: http://www.uddi.org.

[17] W3C. XML Schema Part 2: Datatypes Second Edition. Technical report, W3C, October 2004. http://www.w3.org/TR/xmlschema-2/.

[18] A. B. Williams, A. Padmanabhan, and M. B. Blake. Experimentation with local consensus ontologies with implications for automated service composition. *IEEE Trans. Knowl. Data Eng.*, 17(7):969–981, 2005.