

# Integrating Semantic Web Services for Mobile Access

Anupriya Ankolekar, Pascal Hitzler, Holger Lewen, Daniel Oberle<sup>\*</sup> and Rudi Studer  
Institute AIFB  
University of Karlsruhe  
Karlsruhe, Germany  
{ankolekar, hitzler, lewen, oberle, studer}@aifb.uni-karlsruhe.de

## ABSTRACT

We present our work in integrating Semantic Web services for access via mobile devices. We have developed a system, the `WebServiceAccessComponent`, that transforms a user request for a service on a mobile device, to a Web service request and then selects a matching service from the existing Web services of the Deutsche Telekom, which provide navigational and weather information. In this poster, we present the requirements and design of the `WebServiceAccessComponent`.

## 1. INTRODUCTION

As mobile devices become increasingly widespread and as increasing numbers of companies expose their services as Web services, enabling flexible mobile access to distributed semantic Web resources for advanced personalisation and localisation features is a very relevant challenge. For example, a user can now potentially pose a open-domain question to her mobile digital assistant (MDA), which may even require knowledge about her context. The system can draw on her mobile context and use the Web as a knowledge base as well as a source for access to Web services to provide an answer immediately.

In this poster, we present our work in integrating existing Web services to provide answers to such questions posed by a user. Given a particular user question, e.g., "What will the weather be like tomorrow in Karlsruhe", the system must know that the answer can be delivered by a Web service, must know how to select/discover the corresponding Web service, must know if a composition of several Web service might be able to provide the result, and finally how to automatically invoke it.

Our system, the Semantic Mediator (see Figure 1) and specifically the `WebServiceAccessComponent`, has been developed in the context of the SmartWeb project, which aims to

<sup>\*</sup>This author is now working at SAP Research.

demonstrate the feasibility of multimodal user interfaces to enable access to distributed semantic web resources and services on mobile devices. The SmartWeb system utilizes existing web services, including the T-Info (DTAG) web services<sup>1</sup> offered by Deutsche Telekom AG. The DTAG provides about 50 Web services providing dynamic information such as route planning, maps, weather information, GPS geocodes, locations of cinemas, playtimes of movies, events, points of interest (POI) and many more.

As can be seen in Figure 1, a query issued by the user on her MDA is transformed into an EMMA (Extended MultiModal Annotation markup language)<sup>2</sup> document, to represent the semantics of the query. This document is used by the Semantic Mediator to identify the Web services and knowledge resources that are required to answer the query. The Semantic Mediator then coordinates the access and invocation of these services and resources to return an answer to the user. Ontobroker is used as a storage and querying facility for Web service descriptions.

## 2. DESIGN OF THE WEBSERVICEACCESS-COMPONENT

The primary requirement for the `WebServiceAccessComponent` is that user queries in the form of EMMA documents must be dynamically matched to available Web services, to make the system robust and flexible. This also requires the selection and discovery of available Web services, given a particular user query. Using purely the XML Schema types specified in WSDL files, as Web services are typically described, it would be fairly difficult to automatically select and discover appropriate Web services. By semantically annotating available Web services, we describe the semantics of the inputs and outputs expected by the service. When the semantic annotations have been described through ontologies, a software program can reason about the services using the reasoning capabilities of the underlying logic of the ontology. We use an ontology inspired by OWL-S<sup>3</sup>, SmartSUMO [2], but which address some of the shortcomings of OWL-S for our context.

To address these requirements, we essentially model existing DTAG Web services and attach semantic annotations to them. Furthermore, we perform the selection and discovery

<sup>1</sup><http://services.t-info.de/soap/index.jsp>

<sup>2</sup><http://www.w3.org/TR/emma/>

<sup>3</sup><http://www.daml.org/services/owl-s/1.1/>

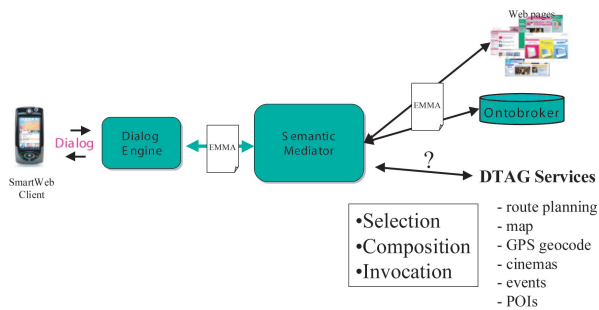


Figure 1: The Semantic Mediator in SmartWeb

of a matching Web service given a user question represented as an EMMA document. In the following, we enumerate the tasks involved for each of these and our design decisions for each of them.

1. Identification and representation of questions: For each DTAG service we have to identify possible natural language questions, based on the occurrence of certain keywords, and to decide how they can be represented via EMMA and SmartSUMO, respectively. The actual descriptions are shown in the poster.
2. Extensions of SmartWeb ontologies: After deciding on the ontological representations of Web services, many additional concepts and relations were identified, which did not yet exist within the ontologies used in the SmartWeb project. The SmartWeb ontologies were therefore extended to enable a single ontology for the entire SmartWeb project.
3. Creation of EMMA documents: Given the identification and representation of various kinds of possible questions corresponding to all the DTAG Web services and the respective extensions of SmartSUMO, we then needed to specify how the EMMA documents for these queries should look like. This primarily involves configuring the natural language processing components of SmartWeb such that a particular EMMA document is created, given a particular question of the user.
4. Extensions of SmartSUMO for web service annotations: SmartSUMO itself needed to be extended to semantically annotate the DTAG Web services. This extension is separate from that mentioned in point 2, because these are related to describing the semantic web service characteristics themselves as opposed to describing the domain. Thus, we extended the SmartSUMO ontology to describe the concept of a Web service, to add concepts and relations for modelling semantic inputs and outputs of Web services, and to model the behaviour of a Web service. We discuss the last two extensions in the next two points.
5. Representation of web service inputs and outputs: Web service inputs and outputs are modelled as instances. A sequence of inputs is modelled via firstInput and nextInput relations, respectively. We have an ontoType relation to specify the semantics of an input by pointing to this instance. relevantSlot is required to

match the correct Web service parameter if there are multiple parameters of the same type.

6. Representation of web service behaviour: We describe the behavioural aspects of Web services using the Ontology of Plans [1], an application of the Descriptions & Situations design pattern in DOLCE. A Plan consist of multiple Tasks, such as case, branching, synchronization, concurrency, or cycling task, etc., which are related through succession relations. We identify those tasks that are applicable for Web services in the SmartWeb domain and attach corresponding Plans to the DTAG Web services.

The annotation of all the DTAG Web services was performed manually. The selection and discovery of Web services was performed by making a simple assumption. We assumed that the focus of the EMMA document is the semantic output of the Web service, the (rdf:types of the) range of all direct relation to the focus documents are the semantic inputs. Thus, a Web service request can be easily created from an EMMA document. Given the Web service request, we then query the Ontobroker to find matching Web services by input and output type matching. Since we are currently dealing with only a few Web services, their invocation is hardcoded. We will be addressing the automated invocation and composition problems in future work.

### 3. CONCLUDING REMARKS

In this poster, we present the design of the WebServiceAccessComponent, which takes a user query for services or information and uses it to automatically discover and invoke appropriate existing Web services. This system is currently being used within the SmartWeb project, which aims to provide multimodal access to distributed semantic Web resources via mobile devices. Of course, the context here is simpler than the usual Web services scenario in that there is no problem of ontology mediation, since the entire system uses a single ontology, the SmartSUMO ontology. We are also dealing with a relatively small set of 50 Web services, making the selection, composition and invocation problems considerably simpler. However, we believe that this system is still of value to the semantic Web services community as it is a real-world working system that demonstrates the feasibility and value of the semantic Web services scenario.

### 4. ACKNOWLEDGEMENTS

This document contains the work and advice of Robert Porzel (EML), Berenike Loos (EML), Hans-Peter Zorn (EML), Michael Sintek (DFKI), Malte Kiesel (DFKI), Daniel Sonntag (DFKI), Ralf Engel (DFKI), Massimo Romanelli (DFKI), Felix Burkhardt (Telekom), Jianshen Zhou (Telekom), Jin Liu (Telekom), Andreas Eberhart (HP invent). Thank you!

### 5. REFERENCES

- [1] A. Gangemi, S. Borgo, C. Catenacci, and J. Lehmann. Task taxonomies for knowledge content. Technical report, Metokis deliverable d07, 2004.
- [2] D. Oberle. Semantic management of middleware. In A. Sheth and R. Jain, editors, *The Semantic Web and Beyond*, number 1. Springer, New York, 2005.