

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/230651552>

An Application For Semantic Annotation Of Web Services

Conference Paper · February 2010

DOI: 10.13140/RG.2.2.21911.16807

CITATIONS

4

READS

181

4 authors, including:



Milos Jovanovik

Ss. Cyril and Methodius University in Skopje

58 PUBLICATIONS 184 CITATIONS

[SEE PROFILE](#)



Riste Stojanov

Ss. Cyril and Methodius University in Skopje

33 PUBLICATIONS 97 CITATIONS

[SEE PROFILE](#)



Dimitar Trajanov

Ss. Cyril and Methodius University in Skopje

151 PUBLICATIONS 537 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



HC-Data: Data Analytics in Healthcare [View project](#)



Master Degree [View project](#)

AN APPLICATION FOR SEMANTIC ANNOTATION OF WEB SERVICES

Koste Budinoski, Milos Jovanovik, Riste Stojanov, Dimitar Trajanov
Faculty of Electrical Engineering and Information Technologies
Rugjer Boskovik, bb, 1000 Skopje, Republic of Macedonia
kostabudinoski@gmail.com, milos@feit.ukim.edu.mk,
ristes@feit.ukim.edu.mk, dimitar.trajanov@feit.ukim.edu.mk

Keywords: Semantic web services, Annotation, Ontologies, Applications.

ABSTRACT

The progress in the field of web services has made it possible to publish, locate, and invoke specific applications across the Web. This is the reason why more and more companies and organizations now implement their core business, or outsource other application services, over the Internet. Therefore, the efficient selection and integration of inter-organizational services on the Web, at runtime, becomes an important issue to the web service provision. The general question is how to develop mechanisms to automatically locate the correct web service in order to meet the user's needs. In special cases, if no single web service can satisfy the functionality required by the user, there should be a possibility to combine existing web services together, in order to fulfil the request.

In this paper, we present a web application that uses ontologies in order to semantically annotate web services, in order to enable the necessary processes, such as automatic discovery, automatic invocation and automatic composition of semantic web services [1]. The application uses the SAWSDL mechanism [2] to connect the concepts of semantic models from an ontology, with the WSDL [3] components. Its main aim is to provide an easy and flexible user interface, presenting the web service description as a tree structure, and allowing the addition of ontologies that would be used for annotation. All of the web services are stored into a one central public repository, which would later be used for automatic web service discovery, automatic invocation and automatic composition.

I. INTRODUCTION

The industry views Web services as abstract, standardized interfaces to business processes. The specification of a web service is expressed in WSDL, which specifies only the syntax of the messages that enter or leave a computer program. In order to move the web services towards the idea of the future of the Web, i.e. the Semantic Web, the functionality of a web service needs to be described with additional pieces of information. With the Semantic Web infrastructure in place, practical and powerful semantic applications can be written for automatic discovery, execution, composition and interoperability of web services.

To allow the semantic web services to become a reality, the markup language used for the process of annotation of

web services must be descriptive enough so that a computer or a software agent can automatically determine its meaning. For this purpose, we developed a web application which provides an easy and flexible user interface for semi-automatic semantic annotation of web services. For the semantic annotation we use OWL ontologies [4] and the SAWSDL mechanism [2], to connect the ontologies concepts of the semantic models with the WSDL components.

Currently, web services that use the .NET framework and the J2EE framework are struggling to expand against the limitations of the existing Web architecture and the conflicting proprietary standards. With software vendors battling for any advantage, the semantic web services offer a giant leap forward; the first-time developer can successfully exploit its latent potential to deliver such applications as semantic search, collective email and collaborative Web word processing.

II. MOTIVATING TASK

With the semantic annotation of web services, the services are descriptive enough so that a computer or a software agent can automatically determine its meaning and implement tasks such as automatic service discovery, invocation and composition of web services:

- **Automatic web service discovery** [5] - It allows us to automatically discover web services, for a specific need of the user. With semantic annotation, web services gain importance in enabling efficient ways of searching and locating the required web service.
- **Automatic web service invocation** - It allows us to process automatic service invocation using a computer program or a software agent, given only a declarative description of that service.
- **Automatic web service composition** [1] - This task involves the automatic composition and interoperation of web services required to perform some complex task, which includes results from more web services.

A similar desktop application, which offers the opportunity to annotate semantic web services, is WSMO Studio [6], which is built on top of the Eclipse platform [7]. The application has a built-in SAWSDL editor for adding semantic annotations to WSDL documents which are then stored in a local repository. Our goal is to implement a web application which uses the same SAWSDL mechanism [2], but is available for users on the Web. This way, users can

publish their existing web services in one central public repository, and then use the mechanism for SAWSDL annotation to create semantic web services, which will also be available (through the web application) to many users on the Web. The users can use existing ontologies, or they can add new OWL ontologies into the ontology repository.

The central repository of semantic web services of the application can be used for automatic discovery, invocation and composition of semantic annotated web services.

III. WEB APPLICATION FOR SAWSDL ANNOTATION

The web application allows the users to annotate web services, semi-automatically, using the SAWSDL mechanism [2] for connecting the concepts of the semantic models with the WSDL components. The Semantic Annotations for WSDL and XML Schema (SAWSDL) is a W3C Recommendation, and it defines the mechanisms for adding semantic annotations to the WSDL components of the web service. SAWSDL does not specify a language for representing the semantic models, e.g. ontologies. Instead, it provides mechanisms by which concepts from the semantic models, which are defined either within or outside the WSDL document, can be referenced from within WSDL components as annotations. These semantics, when expressed in formal languages, can help disambiguate the description of web services during automatic discovery and composition of the web services. The specification enables semantic annotations for web services using and building on the existing extensibility framework of WSDL.

When we start the web application it looks like Fig. 1.

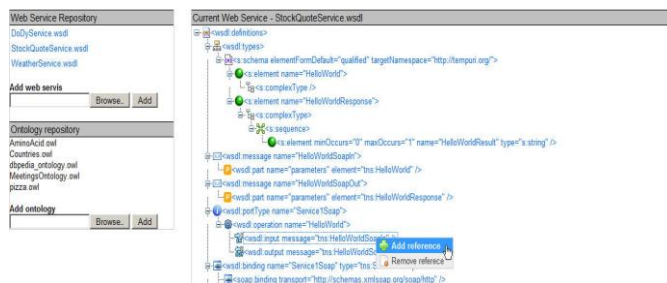


Figure 1. Web application for semantic annotation of web services

On the left side, you can see the repository of web services and the repository of ontologies. The web service repository stores all the web services which are semantically annotated, i.e. the semantic web services. Also, the users have the option of adding new web services which should be semantically annotated later on.

Below it you can see the repository of ontologies. It keeps all of the ontologies that would be used to process the annotations of a web service. The elements of a web service will be annotated using the semantic concepts from these ontologies. Also, the users have the option of adding new ontologies, which are then used for connecting concepts with the components of the web services.

When the user selects a web service from the repository, the right side loads the web service and represents it as a tree structure, with each of its elements (operations, input and output elements, interfaces, etc.) and its attributes. With this tree structure, the user gets an easy and flexible user interface for semantic annotation of web services.

With the right-click on any of the elements of the loaded web service, a context menu appears, with two options. The first option is used to add a reference to a concept of an existing ontology, i.e. to connect an element of the web service with a semantic concept of ontology. The other option is to delete the existing reference to a semantic concept.

By choosing the first option from the context menu, the element is actually associated with a given semantic concept or, in other words, the user describes the element with additional pieces of information. A new modal window opens and loads all of the current ontologies available in the repository, and it gives the user the opportunity to choose which ontology to use for the process of annotation.

If the user wants to change a semantic concept of an element of a web service, he/she can do it in two ways. The user can either delete the existing semantic concepts, and then go through the previous procedure again, or he can directly add a new semantic concept. In the latter case, the web application takes care to delete the previous semantic concept and set the new selection.

When the user deletes an existing semantic concept, a new modal window appears which asks whether he/she is sure that he/she wants to delete the current semantic concept. When the user verifies, the web application deletes the connection of the WSDL element with the current semantic concept.

In conclusion, the web application contains three main parts: a web service repository, an ontology repository and a mechanism for semantic annotation of web services which are stored in the repository. They are connected together as shown in Fig 2.

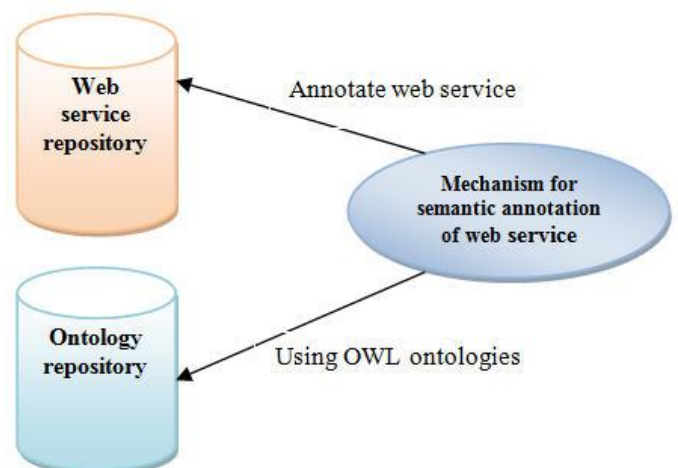


Figure 2. Mechanism for semantic annotation of web services

Using this mechanism for semantic annotation of web services and in the same time storing the annotated web services in one central public repository, we can implement

the automatic discovery of the necessary web services, the automatic invocation (using a computer program or a software agent) and the automatic composition of web services, required to perform some complex task which includes results from more web services.

IV. ARCHITECTURE AND TECHNICAL DESIGN

The web application (Fig. 3) contains three main parts, with the main responsibility to semantically annotate web services. Fig. 3 shows the interoperations that occur between this three main parts and the user interface which the user gets.

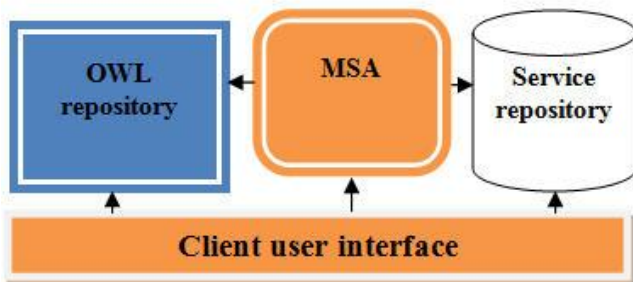


Figure 3. The main interacting modules of web application

The main architectural components include an OWL Ontology repository, a (semantic) web service repository, a mechanism for semantic annotation and a user interface.

OWL ontologies repository – Keeps track of all the ontologies which are used to process the annotations of a web service. OWL as a language provides the capability of creating classes, properties and defining instances. The elements of a web service will be annotated using the semantic concepts from the ontologies held in this repository. The repository is extendable, and the users have the option for adding new ontologies.

A (semantic) web service repository – Keeps track of all web services that need to be annotated and the semantic web services that are already annotated. The web services are represented with their WSDL documents. The service repository is extendable, and the users have the option for adding new web services. Later on, this repository would be used for the automatic web service discovery, the automatic invocation and the automatic composition processes.

MSA (Mechanism for semantic annotation) - Provides the needed interoperability between the OWL ontology repository and web service repository. It is a mediator between these two repositories and its main goal is to provide mechanisms by which the concepts from the semantic models can be referenced with the WSDL elements of the web services. We use the SAWSDL mechanism [2] for annotating the WSDL components.

User interface - It supports the user interaction with the web application. All of the web services are represented as a tree structure, which makes the navigation easy and flexible. All of the references from the ontologies can be added or deleted using the context menu, which can be activated with a

right-click on any of the WSDL elements of a web service. To support all this user interaction, and in the same time define easy and flexible user interface, we use jQuery [8], a lightweight cross-browser JavaScript library. Besides jQuery [8], the web application was developed with the use of the Microsoft .NET framework [9].

V. CONCLUSION AND FUTURE WORK

In this paper we describe our web application for semantic annotation of web services. With semantic annotation, the web services are described with additional pieces of information using concepts from OWL ontologies, in order to enable the automated processes in the Semantic Web, such as automatic discovery, automatic invocation and automatic composition of semantic web services. The web application uses one central public repository of web services which is extendable. The idea behind this repository is for it to be used for automatic web service discovery, automatic invocation and automatic composition. For the annotation of the web services, we use the SAWSDL mechanism [2].

In our future plans for the web application, we plan to include the use of preexisting online repositories of ontologies, from which the concepts will be used in the process of annotation of the web services.

REFERENCES

- [1] Biplav Srivastava, Jana Koehler: „Web Service Composition - Current Solutions and Open Problems“, ICAPS 2003 Workshop on Planning for Web Services
- [2] Farrell, J., Lausen, H.: „Semantic annotations for WSDL and XML Schema“. W3C working draft, W3C (2006) Available online at <http://www.w3.org/TR/sawSDL/>.
- [3] Web Services Description Language, <http://www.w3.org/TR/wsdl>
- [4] Web Ontology Language, <http://www.w3.org/2004/OWL/>
- [5] María Agustina Cibrán, Bart Verheecke, Davy Suvéé, Wim Vanderperren, Viviane Jonckers: „Automatic Service Discovery and Integration using Semantic Descriptions in the Web Services Management Layer“
- [6] Marin Dimitrov, Alex Simov, Vassil Momtchev, Mihail Konstantinov: „WSMO Studio – a Semantic Web Services Modelling Environment for WSMO (System Description)“. Wsmo studio available online at <http://www.wsmostudio.org>
- [7] Eclipse, <http://www.eclipse.org>
- [8] jQuery, <http://en.wikipedia.org/wiki/JQuery>. Available online at <http://jquery.com/>
- [9] Microsoft .NET Framework , <http://www.microsoft.com/.NET/>