OPOSSum - An Online Portal to Collect and Share SWS Descriptions

Ulrich Küster, Birgitta König-Ries, and Andreas Krug
Institute for Informatics, Friedrich-Schiller-University Jena, Germany
ukuester|koenig|andreas.krug@informatik.uni-jena.de

Abstract

Semantic web services have received a significant amount of research attention in the last years but too little effort has been put into the evaluation of the approaches so far. The main blocker of thorough evaluations is the lack of large and diverse test collections for semantic web services. In this demo we present a portal designed to help working towards common test collections by making it easy to collect, search for, and compare semantic service descriptions across various formalisms.

1 Introduction and Motivation

In recent years semantic services research has emerged as an application of the ideas of the semantic web to the service oriented computing paradigm. Semantic web services (SWS) have received a significant amount of attention and research spending since their beginnings roughly six years ago [3]. However, despite of a wealth of theoretical work, surprisingly little effort has been spent towards the comparative evaluation of the competing approaches [2]. This lack of experimentation that might prove the applicability and relative advantage of SWS to real world problems is a critical blocker to the further development of the field as well as the industrial adoption of the developed technologies. The main problem in evaluating SWS technology is the lack of sufficient test data, i.e. semantically annotated web services, which, as recent studies have shown [1], is not yet available. Large and diverse SWS test collections cannot be built by a single group alone but must be created by community effort. This is only possible, if tools exist, which support to collect, share, and improve SWS descriptions collaboratively. So far, such tools have been lacking.

The authors of OWLS-TC, the largest publicly available SWS test collection, have recently set up the SWS-TC-Wiki\(^1\) to foster community participation in improving OWLS-TC. SWS-TC-Wiki enables easy sharing of semantic descriptions but does not allow to edit existing descriptions easily or to search for descriptions with particular properties and to compose collections based on results of such searches. It is the goal of the OPOSSum (Online Portal for Semantic Services) Project described in this demo to improve in this direction. OPOSSum aims to be an infrastructure to ease collecting, sharing, editing, and comparing SWS descriptions across formalism by means of a public structured database of service descriptions. Compared to SWS-TC-Wiki, the only existing effort in this direction, it features much stronger structuring and can thus support more complex use cases, in particular more powerful search and comparison capabilities. OPOSSum has been implemented as a web application and is available online\(^2\). The existing test collections SWS-TC 1.1 and OWLS-TC 2.2 as well as some services from other sources have been fully integrated. OPOSSum currently contains more than 1300 service descriptions for more than 1250 services, making it the largest publicly available collection of SWS.

2 Goals and Layout of the Demo

OPOSSum is meant to be a starting point and tool to help the community to build the urgently necessary better and larger SWS test collections. In this aspect community participation is needed to bring the project alive by adding new data to OPOSSum and by improving the quality of the data already listed. However, to really make OPOSSum a tool which fulfills the needs of the community, feedback from the community about lacking features or flaws in the usability of existing ones is essential, too. The demonstration is intended to publicize the project and collect this feedback. Therefore, the demo will not be a traditional presentation. Instead, participants will only be assisted where necessary and otherwise encouraged to actively try and use OPOSSum themselves.

\(^1\)http://www-ags.dfki.uni-sb.de/swstc-wiki/index.php/Main_Page

\(^2\)http://hnsp.inf-bb.uni-jena.de/Opossum/
3 Presented Functionality

In the following, examples of the main functionalities available during the demo will be presented. This is done by illustrating how OPOSSum supports Alice and Bob, two imaginary users, performing tasks typically done when working with SWS collections.

Bob is looking for test data to evaluate the runtime performance of his OWL-S matchmaking algorithm. He uses the search function to search for service descriptions written in OWL-S 1.1 and discovers 1244 descriptions. OPOSSum’s data model allows to specify the resources (e.g. ontologies or schemas) that a description references. Upon request, OPOSSum writes the 1244 services and the resources referenced by them to files and makes them available for download as a zip archive. Working with the data, Bob realizes that the ApothecaryOntology is inconsistent. He accesses the list of resources available in OPOSSum, uses the search field to locate the ApothecaryOntology, opens it online and fixes the problem. The fixed version is immediately available to all other users of OPOSSum.

In order to improve his matchmaker evaluation, Bob needs to create a set of OWL-S descriptions for the same service following different modelling techniques. He decides to use a travel scenario as the base for the services. He browses the 43 services listed under the category travel to select a suitable service. He selects a flight booking service, downloads the available descriptions and creates the necessary variations. Upon completion he decides to share his work and attaches the created descriptions as alternative descriptions to the same original service instance. In order to be later able to easily find them, he tags them with an identifier of the modelling approach used.

Alice would like to perform a comparative evaluation of different service description formalisms, in particular OWL-S and WSMO/WSML. Furthermore she would like to compare how those descriptions relate to traditional WSDL descriptions. Thus she needs to find corresponding descriptions for the same service in all three formalisms. She uses OPOSSum’s search function to search for services with attached descriptions in all of OWL-S, WSML, and WSDL and retrieves those descriptions as a zip archive.

4 Implementation

OPOSSum has been implemented as a PHP-based web application on top of a MySQL database. Thus a web browser and an Internet connection are the only prerequisites of using it. OPOSSum’s data model is structured around the notion of a Service, independent from particular Descriptions, which can be attached to a service object. A service in OPOSSum is primarily described by a natural language text. To add more semantics without binding to a particular formalism, OPOSSum uses WordNet synsets to disambiguate types and tags used throughout the system. This provides a kind of semantics that ensures an excellent compromise between being unambiguous, flexible, easily usable and language/formalism independent.

5 Summary

In this paper we argued that too little effort is put into the evaluation of SWS technology. One of the main reasons is the lack of diverse test collections and the lack of tools to support the community in building them collaboratively. We presented OPOSSum, a portal that aims to be such a tool by providing an infrastructure to ease the sharing, editing and comparison of SWS descriptions across formalisms.

References


WordNet (http://wordnet.princeton.edu/) is a semantic lexicon for the English language developed at Princeton University. It uses the notion of synsets to collect synonyms and disambiguates homonyms. Sense keys are used to reference synsets, thus providing a unambiguous identifier of a particular semantic meaning.