SWOP: A Semantic Web services Oriented Platform

Eufemia Tinelli[1], Michele Filannino[1], Daniele Casulli[2], and Leo Iaquinta[1]

[1] Computer Science Department of University of Bari, Via Orabona, I-70125, Bari, Italy
[2] Code Architects S.r.l., via Iazzitiello n. 15/C, I-70029 Santeramo in Colle, Bari, Italy

Abstract. SWOP is an ongoing industrial research project which aims at developing a semantic-based platform of Web services for .NET developers. Our framework exploits NLP algorithms and Semantic Web technologies in order to provide semantic web service annotation and discovery.

1 Introduction

SWOP (Semantic Web Service Oriented Platform) is an ongoing industrial research project which aims at developing a framework for the annotation and the discovery of Web Services (WSs). The end-user is the WSs developer and the proposed platform provides the following main functionalities: (i) the user can deploy and store web services interfaces; (ii) the user can enhance the previous interfaces choosing relevant ontological concepts suggested by the system according to a semi-automatic approach to WSs annotation; (iii) before developing a new web service S_1, the user can perform a semantic-based service discovery for checking if web services "similar to S_1" already exist.

As it is well-known in literature, semantic annotation of WSs is fundamental to support the automation of discovery, selection and composition. Manual annotation is a simple but expensive strategy. Moreover, it suffers for annotator familiarity with the domain, amount of training, personal motivation and complex schemas. On the other hand, it is not yet possible to automatically identify and classify all entities in source documents with adequate accuracy. Thus, the annotation task should rely on some human intervention. Reeve and Han [4] summarized the approaches to support the semantic annotation of WSs. By comparison with FUSION project [3] our approach is novel because it combines benefits from both Semantic Web and Natural Language Processing (NLP) technologies. In particular, we exploit NLP algorithms to spot relevant concepts, whereas we take into account WS semantic annotations in the WS discovery. Moreover, the SWOP platform enhances an existing .NET platform by means of several technologies: WSDL 1.1 for deploying WSs, SOAP protocol for invocating them and SAWDL for creating semantically annotated descriptions. In order to add semantics to WSs, we developed a domain ontology modelling enterprise business such as sale/purchase of products. The SWOP ontology is written in OWL-DL and follows the modularization principles of Ontology Engineering. Furthermore, the service descriptions are not dependent on the domain because they concern functional and non-functional properties of a service. Thus, importing a different ontology allows to re-focus the domain of interest with specific data and operations of the WSs.

[3] http://www.fusion-strep.eu/

2 SWOP Framework

As above mentioned, our platform provides WSs annotation and semantic web services (SWSs) discovery which are briefly presented in the following:

WS Annotation - The WS developer provides a textual description for each WSDL node that he would annotate. Consequently, the system exploits the SWOP ontology to suggest a set of concepts relevant for the annotated nodes. This suggestion is performed by SAWA [4] algorithm based on the text-to-text semantic similarity between WS annotations and the comments of classes in the domain ontology. SAWA has been devised as extension of a word-to-word similarity algorithm [1, 2] that uses Wikipedia as corpus. The developer is able to either accept or reject the suggested concepts as well as she is able to directly browse the SWOP ontology. Finally, SWOP framework creates the corresponding SAWSDL description using the concepts selected by the developer;

SWS Discovery - The SAWSDL descriptions are preliminarily parsed to obtain OWL descriptions of WSs w.r.t. the SWOP ontology. As in the WS Annotation phase, the developer produces a textual description of each WSDL node of interest to obtain suggestions of the most relevant concepts. Then, the resulting (partial) WS description is translated into OWL to semantically represent the user query. Finally, the system performs a matchmaking process of the user query against the available OWL descriptions of WSs in the SWOP repository. The matchmaking process considers service functionality, category, input and output data. In particular, we exploit Pellet reasoner to perform three match classes: *exact match*, *plug-in match* and *full match* similarly to service matching presented in [3], whereas the list of the returned SWSs is ranked according to service similarity calculated by SAWA.

The SWOP platform was built according to the three-tier architecture model: the presentation layer, the business layer, and the domain layer. Each layer is independent from the others and it can run on a different physical computer. In particular, we briefly discuss the main functionalities of each of them: (i) the presentation layer contains the SWOP user graphical interface which was completely developed using Microsoft Silverlight (*i.e.*, it runs on client-side); (ii) the business layer represents the core of the platform, it takes advantage of SOA architecture developed using both .NET and Java technologies; (iii) the domain layer consists of a database server for storing and retrieving data required by the SWOP platform (WSDL, SAWSDL and OWL file components), *i.e.*, it represents our semantic-based UDDI register.

References

1. Corley, C., Mihalcea, R.: Measuring the semantic similarity of texts. In: ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment. pp. 13–18. ACL (2005)
2. Lin, D.: An information-theoretic definition of similarity. In: ICML '98. pp. 296–304. Morgan Kaufmann (1998)
3. Paolucci, M., Kawamura, T., Payne, T.R., Sycara, K.P.: Semantic matching of web services capabilities. In: ISWC '02. pp. 333–347. Springer-Verlag (2002)
4. Reeve, L.: Survey of semantic annotation platforms. In: ACM Symposium on Applied Computing. pp. 1634–1638. ACM Press (2005)

[4] http://193.204.187.223:8080/sawa/