Ulrich Lampe, Melanie Siebenhaar, Stefan Schulte, Ralf Steinmetz:
A Graphical Evaluation Tool for Semantic Web Service Matchmaking.
In: Axel Polleres, Huajun Chen: Poster and Demo Proceedings of 9th International Semantic Web Conference (ISWC2010), November 2010.

A Graphical Evaluation Tool for Semantic Web Service Matchmaking

Ulrich Lampe, Melanie Siebenhaar, Stefan Schulte, and Ralf Steinmetz

Multimedia Communications Lab (KOM),
Technische Universität Darmstadt, Germany
ulrich.lampe@kom.tu-darmstadt.de

Abstract. Semantic matchmaking – i.e., the task of finding matching (Web) services based on semantic information – has been a prominent field of research lately, and a wide range of supporting tools both for research and practice have been published. However, no suitable solution for the visualization of matchmaking results exists so far. In this paper, we present the *Matchmaking Visualizer*, an application for the visual representation and analysis of semantic matchmaking results. It allows for the comparing of matchmaking approaches for semantic Web services in a fine-grained manner and thus complements existing evaluation suites that are based on rather coarse-grained information retrieval metrics.

1 Introduction

In the envisioned Internet of Services (IoS), organizations will be able to realize business processes using not only internal (Web) services, but also external (Web) services from public marketplaces. One prerequisite to this vision is that functionally equivalent services can be effectively and efficiently identified in the potentially huge collection of available service offers. This service discovery process based on functional requirements is referred to as *matchmaking*. Recent approaches to matchmaking frequently utilize semantic information in addition to the purely syntactical service descriptions [1–3], based on the notion of Semantic Web services (SWS) [4].

In current research, the evaluation of matchmaking approaches is mostly based on standard Information Retrieval (IR) metrics such as precision and recall. These metrics are, for instance, employed in the popular *Semantic Matchmaking Evaluation Environment* (SME2) tool[1]. IR metrics, however, exhibit a rather coarse-grained representation of matchmaking performance, because the service rankings are merged into a selected number of (more or less representative) figures. This type of evaluation provides little insight into the actual matching process. For instance, the similarities that have been assigned to certain pairs of components in a service, such as operations, and the resulting assignments can not be assessed in detail. However, this type of information can be beneficial

[1] http://projects.semwebcentral.org/projects/sme2/

in order to comprehend a matchmaking process in detail. It can also help to analyze required modifications to a (Web) service for actual consumption within, e.g., a complex business workflow.

In this paper, we present the *Matchmaking Visualizer* (MV) as a complement to existing matchmaking evaluation tools. It facilitates the visual representation and analysis of individual service offers and requests and the respective matching results. This permits a much more fine-grained evaluation and assessment of matchmaking engines.

2 Prototype Description

2.1 Representation of Matchmaking Results

A matchmaking result consists of three parts in principle: The description of the initial (1) service request and (2) service offer, and (3) the collection of matching results between pairs of components in both services. The latter express the similarity between individual components in service request and offer, based on their functional and/or semantic description.

We assume that both service request and offer are represented in the identical form, namely as a complete service description file. Reflecting the variety of available service description formats for both WS-* and RESTful services, MV handles the Web Service Description Language (WSDL)[2] in conjunction with Semantic Annotations for WSDL and XML Schema (SAWSDL)[3], Web Ontology Language for Services (OWL-S)[4], Web Application Description Language (WADL) [5], and HTML for RESTful Services (hRESTS) [6].

Each individual matching result consists of the *qualified names* (QNames) of the two regarded components and a flag that indicates if the two components have been assigned to each other. The matching result further links to at least one similarity assessment, which consists of a numerical and textual similarity and a free-text comment. QNames comprise the namespace of a component as well as its local name and are unique to each component in a service description. Because some components, such as individual parameters, may be referenced multiple times, the QName of any parent of a component may optionally be provided as well. Both numerical and textual similarity have been included because they also appear in existing matchmaking approaches. For example, discrete *Degrees of Match* are used in [2], whereas [1, 3] apply continuous numerical measures. The data structure for the representation of matching results is made available as Java classes. This facilitates compatible external matchmakers generating the appropriate data at runtime.

[2] `http://www.w3.org/TR/wsdl`, `http://www.w3.org/TR/wsdl20/`

[3] `http://www.w3.org/TR/sawsdl/`

[4] `http://www.w3.org/Submission/OWL-S/`

2.2 Integration of External Matchmakers

Because MV constitutes an (unofficial) complement to SME2, we have decided to facilitate SME2's plug-in architecture for the integration of external matchmakers. It requires matchmakers to be provided as a JAR file (optionally with additional libraries) along with an XML-based description. As does SME2, MV requires each compatible matchmaker to implement a Java interface class. This interface, *IMatchmaker*, specifies one method, *matchServices*, which conducts the matchmaking process for a service request and offer file. The locations of the respective description files are provided as *URLs*. The matching result is returned using the Java data structure outlined in the previous section. Alternatively, complete matchmaking results may be provided to MV in the form of XML-based files. This can be useful to matchmakers that do not facilitate MV's plug-in architecture, e.g., if they are not implemented in Java.

2.3 Implementation

MV has been implemented in Java using different existing frameworks. For the visualization of graphs, the Java Universal Network/Graph Framework (JUNG)[5] is used. To read semantic service descriptions, we employ Woden4SAWSDL[6] and OWL-S API[7] to read (SA)WSDL and OWL-S documents respectively. For the processing of WADL documents, the generic JDOM[8] framework is utilized. hRESTS documents are first converted into an RDF model, using a XSLT stylesheet by Kopecký[9], and subsequently processed using the JENA framework[10].

2.4 User Interface

MV provides a platform-independent Graphical User Interface (GUI) based on the Java Swing framework. A portal site, including a demo video, screenshots, and additional example resources, is available at:

`http://www.kom.tu-darmstadt.de/~lampeu/iswc-2010/`

The GUI depicts a service request and service offer in the form of a tree structure, representing the increasing level of abstraction. The root, i.e., the whole service, branches into individual leaves, i.e., components, with parameters constituting the most fine-grained level of abstraction. The level of expansion for the tree may be globally selected, thus allowing to focus on the level of abstraction of highest interest. Between all components on the currently selected

[5] http://jung.sourceforge.net/

[6] http://lsdis.cs.uga.edu/projects/meteor-s/opensource/woden4sawsdl/

[7] http://on.cs.unibas.ch/owls-api/index.html

[8] http://www.jdom.org/

[9] http://cms-wg.sti2.org/TR/d12/v0.1/20081202/xslt/hrests.xslt

[10] http://jena.sourceforge.net/

level, the matching results are visualized in the form of edges. By selecting a component, the associated source code from the service description file may be displayed in the GUI. Further, by selecting a connecting edge, details about the respective matching result can be displayed.

MV allows conducting matchmaking on a selected service request and offer using any of the plug-in matchmaking engines. Results may also be combined for direct comparison. That way, users may assess in detail which similarities have been given to pairs of components and why certain components have been assigned to each other. Further, an existing XML result file may be loaded or an individual service may simply be displayed for analysis.

3 Summary

In the work at hand, we presented the Matchmaking Visualizer, a tool for the visualization of semantic Web service matchmaking results. It complements existing evaluation tools for semantic matchmakers that rely on rather coarse-grained information retrieval metrics. Matchmaking Visualizer permits a much more fine-grained, in-detail analysis of matchmaking results, down to the level of individual components and offers support for the direct comparison of different matchmaking approaches. For that matter, the tool is compatible with the plug-in architecture of the popular SME2 evaluation suite, offering the convenient integration of external matchmakers.

Acknowledgements. This work is supported in part by E-Finance Lab e. V., Frankfurt am Main, Germany (www.efinancelab.de).

References

1. Schulte, S., Lampe, U., Eckert, J., Steinmetz, R.: LOG4SWS.KOM: Self-Adapting Semantic Web Service Discovery for SAWSDL. In: IEEE 2010 Fourth International Workshop of Software Engineering for Adaptive Service-Oriented Systems (SEASS '10), Los Alamitos, CA, USA, IEEE, IEEE Computer Society (Jul 2010)
2. Klusch, M., Kapahnke, P., Zinnikus, I.: Hybrid Adaptive Web Service Selection with SAWSDL-MX and WSDL-Analyzer. In: The Semantic Web: Research and Applications, Proceedings of the 6th European Semantic Web Conference (ESWC 2009). Volume 5554 of Lecture Notes in Computer Science., Springer (2009) 550–564
3. Plebani, P., Pernici, B.: URBE: Web Service Retrieval Based on Similarity Evaluation. IEEE Transactions on Knowledge and Data Engineering **21**(11) (2009) 1629–1642
4. McIlraith, S.A., Son, T.C., Zeng, H.: Semantic Web Services. IEEE Intelligent Systems **16**(2) (2001) 46–53
5. Hadley, M.: Web Application Description Language (WADL). Sun Microsystems, Inc. Technical Reports; Vol. SERIES13103; SMLI TR-2006-153 (2006)
6. Kopeckỳ, J., Gomadam, K., Vitvar, T.: hRESTS: an HTML Microformat for Describing RESTful Web Services. In: 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, IEEE (2008) 619–625