Semantically Annotating RESTful Services with SWEET

Maria Maleshkova
Knowledge Media Institute
The Open University
Milton Keynes, MK7 6AA, UK
m.maleshkova@open.ac.uk

Carlos Pedrinaci
Knowledge Media Institute
The Open University
Milton Keynes, MK7 6AA, UK
c.pedrinaci@open.ac.uk

John Domingue
Knowledge Media Institute
The Open University
Milton Keynes, MK7 6AA, UK
j.b.domingue@open.ac.uk

ABSTRACT

This paper presents SWEET, the first tool developed for supporting users in creating semantic RESTful services by structuring service descriptions and associating semantic annotations with the aim to support a higher level of automation when performing common tasks with RESTful services, such as their discovery and composition.

1. INTRODUCTION

Since the advent of Web service technologies, research on semantic Web services (SWS) has been devoted to reduce the extensive manual effort required for manipulating Web services. The main idea behind this research is that tasks such as the discovery, negotiation, composition and invocation of Web services can have a higher level of automation, when services are supplemented with semantic descriptions of their properties. Recently, the world around services on the Web, thus far limited to "classical" Web services based on SOAP and WSDL, has significantly evolved with the proliferation of Web applications and APIs, also called RESTful Web services. RESTful services conform to the REST architectural style [1] and are characterised by their relative simplicity and their natural suitability for the Web, which is indeed closely related to the growing popularity and use of Web 2.0 technologies.

Despite their success, RESTful services are currently facing the same limitations that were identified for traditional Web service technologies and present even further difficulties. In particular, as opposed to WSDL services, there is no widely accepted structured language for describing RESTful ones. As a consequence, in order to use RESTful services, developers are obliged to manually locate, retrieve, read and interpret heterogeneous documentations of RESTful services in HTML, and subsequently develop custom tailored software that is able to invoke and manipulate them.

In order to address these challenges, we introduce SWEET (Semantic Web sErvices Editing Tool), a tool for the creation of semantic RESTful services, which supports both the creation of machine-readable RESTful service descriptions and the addition of semantic annotations in order to better support discovering services, creating mashups, and invoking them. SWEET uses the hRESTS (HTML for RESTful Services) microformat [2] for the creation of machine-readable service descriptions and supports the semantic annotation of service properties, which are specified in MicroWSMO [3]. SWEET enables the semi-automatic creation of semantic RESTful services, by hiding formalism complexities from the user and assisting him/her in makings semantic annotations, therefore directly contributing to increasing the level of automation when performing common service tasks.

2. SWEET

SWEET is a Web application developed using JavaScript and ExtGWT[1], which is started in a Web browser by calling the host URL. It takes as input an HTML description of a RESTful services and offers functionalities, which enable users to create annotations on it. As it can be seen in Figure 1, the API description is loaded in the *Navigator* panel, which implements a reverse proxy [4] that enables the communication between the annotation functions and the HTML by rerouting all sources and connections from the original HTML through the Web application. Based on this, the HTML DOM of the RESTful service can freely be manipulated by using functionalities of the *Annotation Editor* panel. First, the user is guided thorough the process of marking service properties with hRESTS tags, by limiting the available tags depending on the current state of the annotation. This implements measures for reducing possible mistakes during the creation of annotations. Second, based on the hRESTS tagged HTML, which provides the structure of the RESTful service, the user can link service properties to semantic content. This is done by selecting service properties, searching for suitable domain ontologies by accessing Watson [5] in an integrated way, and by browsing ontology information. Based on this details the user can decide to associate a service property with particular semantic information by inserting a MicroWSMO model reference tag. It is envisioned that this process is supported by an annotations recommender component, which assists the user in annotating a service by suggesting suitable annotations for the service as a whole (domain ontology recommendation) and for its individual properties. Still, SWEET effectively supports users in marking service properties and in attaching semantic information to them. The current status of the
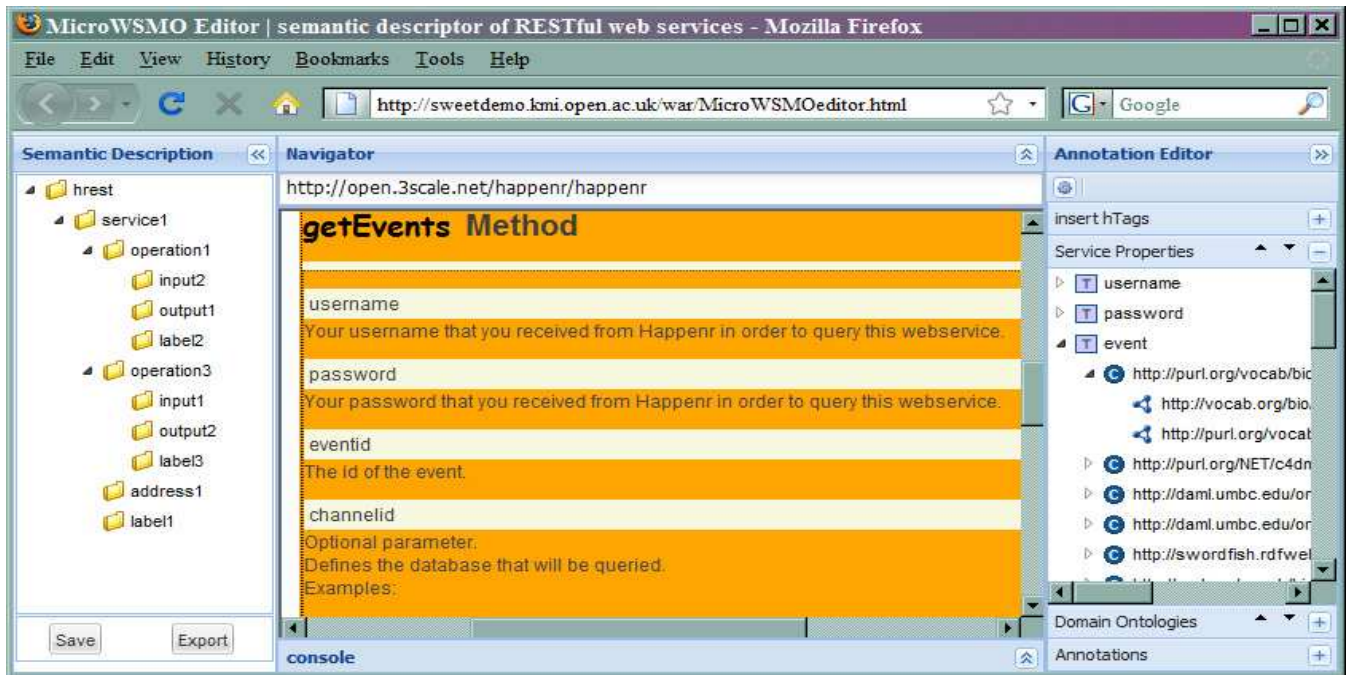
---

[1]http://extjs.com/products/gxt/

Figure 1: Graphical User Interface of SWEET

annotation is visualized in the form of a tree structure in the *Semantic Description* panel. When the user completes the semantic annotation of the HTML description, the annotated HTML can be saved and republished on the Web, representing an instance of a semantic Restful service. In addition, the resulting HTML can also be transformed into a RDF MicroWSMO description, which can be used for manipulation or storage.

There is only little research done addressing the challenges of RESTful services. So far, WADL [6] is one attempt for addressing the lack of standardised description of RESTful services. However, it does not seem to be gaining acceptance and instead Web applications and APIs are usually described in textual documentation. Additionally, there has been significant development in the area of semantic RESTful services (SRS) mostly around the definition of formalisms for creating semantic annotations. As already mentioned, MicroWSMO is one such formalism, which relies on hRESTS for describing the main aspects of a service such as its operations, inputs and outputs, and uses hooks for linking these to semantic information. SA-REST [7], on the other hand, uses the grounding principles of SAWSDL [8] and RDFa for marking service properties. Even though, there is some research done targeted at supporting the use of SRS, for example in the form of mashpus [7], there are no existing tools or approaches supporting the creation of semantic RESTful service descriptions, which therefore hinders the applicability.

SWEET is a major contribution towards providing semantics for RESTful services and supporting the wider use of Web applications and APIs. It guides users through the annotation process, reduces the amount of necessary manual work by making annotation suggestions and hides formal-

ism complexity behind an easy-to-use user interface. A demo version of SWEET is available at http://sweet.kmi.open.ac.uk. SWEET is based upon work partially supported by the EU funding under the project SOA4All (FP7 - 215219). The authors would like to thank Simone Spaccarotella for his contribution to the development of SWEET.

3. REFERENCES

[1] R. T. Fielding: Architectural styles and the design of network-based software architectures. *PhD thesis, University of California*, 2000.

[2] J. Kopecký , K. Gomadam, T.Vitvar: hRESTS: an HTML Microformat for Describing RESTful Web Services. *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence*, 2008

[3] J. Kopecký, T. Vitvar, D. Fensel, K. Gomadam: hRESTS & MicroWSMO. *Technical report*, available at http://cms-wg.sti2.org/TR/d12/, 2009.

[4] E. Gamma, R. Helm, R. Johnson, J. M. Vlissides: *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley Professional, November 1994.

[5] Watson - The Semantic Web Gateway: *Ontology Editor Plugins.* http://watson.kmi.open.ac.uk. Online November 2008.

[6] M. J. Hadley: Web Application Description Language (WADL). *Technical report, Sun Microsystems*, November 2006. Available at https://wadl.dev.java.net.

[7] A. P. Sheth, K. Gomadam, J. Lathem: SA-REST: Semantically Interoperable and Easier-to-Use Services and Mashups. *In IEEE Internet Computing*, 11(6):91Ű94, 2007.

[8] J. Kopecký, T. Vitvar, C. Bournez, J. Farrel: SAWSDL: Semantic Annotations for WSDL and XML Schema. *IEEE Internet Computing*, 11(6):60-67, 2007.