Simplifying Semantic Web application development and semantic data usage

Mariano Rico
Computer Science Department, Escuela Politécnica Superior, Universidad Autónoma de Madrid, Madrid, Spain
E-mail: mariano.rico@uam.es

Abstract. This thesis is focused on reducing the gap between Semantic Web technicians and common Web developers and users. To this end, software infrastructure and Web applications have been developed. Experimental results show that the effort required to develop a simple Semantic Web application is reduced in terms of development time and number of required tools. Besides, users report high values of usability and satisfaction concerning the user interface of these applications.

Keywords: Semantic Web, Semantic Web applications, Web designers, user interaction

1. Introduction

The term Semantic Web is gaining momentum. The first RDF specification is 12 years old and since then many other W3C specifications have been released. The Web is populated with thousands of ontologies and billions of RDF triples, and the Linked Data initiative has established the basis to publish data on the Web. All this wealth of high quality information needs to be visualized, managed, handled and edited. Although notable applications such as DERI Pipes (http://pipes.deri.org) or Sigma (http://sig.ma) have been released recently, the Web of Data remains hidden to most everyday users, and Semantic Web applications [1,2] barely exist beyond computer science labs.

This thesis starts by analyzing the main factors that produce this situation: the adoption barrier of the technologies associated to the Semantic Web is too high, not only for everyday users [7], but for developers too. Although details can be hidden to the user by means of intuitive applications [6], so that the user is not aware of Semantic Web or its technologies, the skills needed to develop Semantic Web applications have rapidly become an unrealistically long list.

2. Objectives and proposals

The objective has been to reduce the gap between Semantic Web experts and common Web developers and users. This thesis proposes a clear separation of roles and a set of proper tools for each role. It is argued that the Semantic Web technicians do not create Semantic Web applications because the skills required to create an ordinary Web application are very high. Imagine one of these experts, with a large background in ontological engineering and experienced with the java language but basic knowledge about Web technologies. The effort required to turn a desktop (command-line or graphical) semantic application written in Java into a Semantic Web application probably will dissuade him.

For these developers we have created Fortunata [5] (http://ishtar.ii.uam.es/fortunata), a wiki-based platform which exploits the benefits of the wiki syntax (simple and easy to learn) and the benefits of an ontology management system. The Fortunata API allows developers to focus in the key programming functionality, delegating tedious tasks such as Web client code generation and data publishing to Fortunata. The plugin architecture paradigm allows developers share and contribute functionality in a low dependent way. Any Fortunata-based Semantic Web application comprises a set of wiki pages and a set of plugins written in Java. The experimental evaluation shows that Fortunata diminish development time (−40%) and number of re-
quired tools (~60%). Although Semantic Web experts can take advantage of Fortunata to easily create Semantic Web applications, the gap between these developers and common Web developers is bridged by means of a key role: Web designers. The specific task associated to Web designers is to contribute with Web templates capable of handling semantic data. These templates can present semantic data to the user (output templates) or can request data to the user (input templates).

VPOET [3] is a Fortunata-based application created specifically for Web designers. VPOET templates are reusable and can be shared among Web designers. Web designers can start by creating templates for a given ontology element after a 20 min online tutorial. This tool has been designed to minimize the Semantic Web skills required by using another Fortunata-based tool named OMEMO. Its main feature is to provide simple descriptions of the elements of a given ontology. For example, if a Web designer is creating a VPOET template for the ontology element FOAF:Person, she can read the OMEMO page for the FOAF ontology, realize that there are 5 versions of this ontology, look for the Person element, and discover that it comprises elements such as firstName or depiction. By following the links in these wiki pages she can get a precise idea of the structure of these ontology elements. The Web designer will not deal with the semantics of these elements, since this information is not relevant to provide a user interface. The experimental evaluation of simple Fortunata-based applications shows high usability (8.2 in a [0–10] range) and user satisfaction (8.7) for a wide range of users, from amateur Web designers to professionals.

Besides being a Web application oriented to Web designers, VPOET is a semantic data source fed by the templates created by a community of Web designers. This source can be exploited easily by common Web developers, in any programming language, by means of HTTP messages. This mechanism does not require developers skilled in ontological query languages such as SPARQL, lowering the adoption barrier. Therefore, any Web applications developer, skilled in any programming language can exploit the VPOET templates to display semantic data or create semantic data in a simple way.

End-users can take advantage directly of VPOET templates. A Google Gadget named GG-VPOET exploits VPOET templates by means of the aforementioned HTTP messages. By using this gadget, any end-user can render a semantic data source or utilize a Web interface to create semantic data. GG-VPOET, as any other Google Gadget, can be inserted into a regular Web page or in Google products such as iGoogle, Google Desktop or Google Pages.

3. Conclusions

Although the experimental evaluation allows concluding that the objective has been accomplished, that is, the tools provided reduce the adoption barrier to create Semantic Web applications, some limitations must be observed. The limitations of the solutions provided open future research lines [4]. The main limitations are due to (1) the used tools, such as the wiki-engine used by Fortunata, with some limitations in forms, (2) the created tools, such as the files based mechanism used to store the semantic data, which has scalability drawbacks for (very) big data sets, (3) the experiments, such as the limited number of participants, which is a balance between precision in the measure and the difficulty to find qualified participants.

The real impact of this work can be assessed by means of the growing number of personal pages taking advantage of semantic data through GG-VPOET. Despite its dependency on professional Web designers, the contributed designs have covered the users needs. Fortunata’s source code (available at Google Code) has been downloaded 40 times for the last 6 months.

References