

ActiveRaUL: A Web form-based User Interface to create and maintain RDF data

Anila Sahar Butt^{1,2}, Armin Haller¹, Shepherd Liu¹, and Lexing Xie²

¹ CSIRO ICT Centre, `firstname.lastname@csiro.au`

² Australian National University, `firstname.lastname@anu.edu.au`

Abstract. With the advent of Linked Data the amount of automatically generated machine-readable data on the Web, often obtained by means of mapping relational data to RDF, has risen significantly. However, *manually created, quality-assured* and *crowd-sourced* data based on ontologies is not available in the quantities that would realise the full potential of the semantic Web. One of the barriers for semantic Web novices to create machine-readable data, is the lack of easy-to-use Web publishing tools that separate the schema modelling from the data creation. In this demonstration we present ActiveRaUL, a Web service that supports the automatic generation of Web form-based user interfaces from any input ontology. The resulting Web forms are unique in supporting users, inexperienced in semantic Web technologies, to create and maintain RDF data modelled according to an ontology. We report on a use case based on the Sensor Network Ontology that supports the viability of our approach.

1 Introduction

The tools of choice for creating quality-assured ontology instances (the so-called *ABox*) are still ontology editors such as WebProtégé [4]. However, creating the *ABox* in an ontology editor requires some degree of understanding of RDF(s) and OWL since the user has to define to which class an individual belongs to and what are the permissible relationships between individuals. To address this issue, some Web publishing tools on top of Wikis, Microblogs or Content Management systems have been developed (e.g. the work discussed in [1], [5] and [3]) that allow a user to exclusively create ontology instances. However, they are mostly developed for a specific domain (i.e. specific ontologies) and often do not strictly follow OWL semantics and consequently allow the creation of a logically inconsistent ABox.

In this demonstration we will showcase ActiveRaUL [6], a Web service that operates on a model defined according to the RDFa User Interface Language (RaUL)³ that consists of two parts, [1.] a *form model* describing the structure of a Web form with different types of form controls, such as *Textboxes*, *Radiobuttons*, *Listboxes* etc., and their associated operations (CREATE, READ, UPDATE or DELETE) and [2.] a *data model* defining the structure of the exchanged data as RDF statements which are referenced from the *form model* via a data binding mechanism. The ActiveRaUL Web service also provides functionality to automatically generate a Web form-based user interface according to the RaUL ontology from arbitrary ontologies. We argue that the resulting user interfaces are easier-to-use for a semantic Web novice to create

³ See <http://purl.org/NET/raul#>

ontology individuals, and result in more accurate ontology instances than creating them through traditional ontology engineering tools. We validated this hypothesis in a user study comparing our system with a state-of-the-art ontology modelling tool.

2 Demonstrating the ActiveRaUL Web service

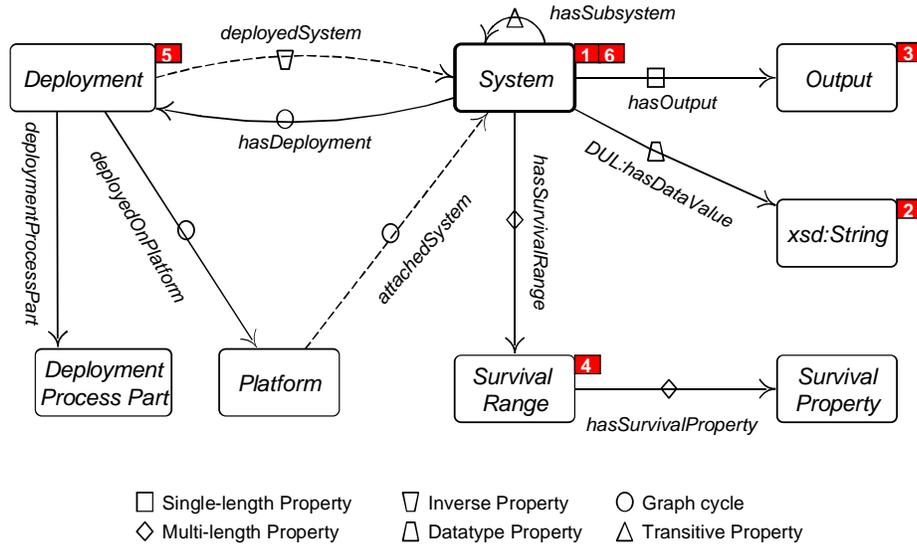


Fig. 1. Sub-graph structure for the System class in the SSN ontology

To relieve a Web developer from manually defining a Web form model according to the RaUL ontology, we have extended the ActiveRaUL service with a deployment endpoint that upon invocation generates RaUL Web forms from an arbitrary user submitted ontology. The biggest challenge in automatically creating such Web forms from an ontology is the mismatch between the graph nature of RDF and the tree structure of a Web form. In the algorithm implemented in ActiveRaUL we distinguish six different types of sub-graphs occurring in ontologies and introduce decision controls to map these sub-graphs to useable web forms. We will demonstrate these different types of mapping on a use-case based on the the Semantic Sensor Network (SSN) ontology that can be used to describe the capabilities of sensors, the measurement processes used and the resultant observations. Figure 1 shows the “System” class of the SSN ontology and its relations to other classes, whereas Figure 2 shows a screenshot of a generated Web form by ActiveRaUL of the “System” class. The numbers 1–6 in both figures indicate the six different types of sub-graphs we distinguish in the algorithm (see Figure 1) and how they are displayed in the Web form (see Figure 2).

Fig. 2. Screenshot of ActiveRaUL generated Web form for the `ssn:System` class

3 Evaluation

We compared ActiveRaUL to the widely used state-of-the-art ontology editing tool, WebProtégé. The demonstration deployment of ActiveRaUL set up for the user study already pre-loading the SSN ontology is available at: <http://www.activeraul.org/demo/index.html> From the university deployment example defined by the SSN working group we extracted three test cases, each with a number of tasks. We asked users to model these test cases in WebProtégé and ActiveRaUL. For evaluating the two systems we considered three usability metrics, (1) the effectiveness of the system in supporting the user to complete the task measured by the accuracy of the resulting models; (2) the efficiency of the users in using the system measured by the time they spent on completing a task and (3) a user's subjective reactions using the system measured by the widely-used System Usability Scale (SUS) [2]. In the following we briefly outline the results of our user study. These results are based on the performance and feedback of twelve participants: five of which, based on their self-assessment, were categorised into the *semantics experienced user group*, and seven categorised into the *semantics inexperienced user group*.

Table 1. Overall accuracy in completing test cases in WebProtégé and ActiveRaUL

	WebProtégé	ActiveRaUL
Exp. Users	82.05%	91.03%
Inexp. Users	76.92%	87.91%
All Users	82.05%	91.03%

Accuracy: Table 1 shows the overall accuracy over the three test cases which shows that the participants performed better in ActiveRaUL, managing to create 91% correct triples compared to 82% in WebProtégé. For ActiveRaUL, the accuracy of the participants was already very high in the first test case, even though no participant has ever used the system before. This confirms our hypothesis that a Web form-based user interface is familiar enough to computer literate users to create RDF data correctly, even if the participants are inexperienced in semantic Web technologies.

Table 2. Average times (mm:ss) to complete test cases in WebProtégé and ActiveRaUL

	Test Case 1		Test Case 2		Test Case 3	
	WebProtégé	ActiveRaUL	WebProtégé	ActiveRaUL	WebProtégé	ActiveRaUL
Exp. Users	3:46	1:50	5:13	1:50	10:24	4:01
Inexp. Users	4:05	2:17	5:23	1:29	11:26	4:08
All Users	3:57	2:05	5:19	1:38	11:00	4:05

Efficiency: Table 2 shows the average times participants required to complete a test case. Both participant groups, inexperienced and experienced, were significantly faster (between 27% and 56% faster) completing the test cases in ActiveRaUL compared to WebProtégé.

Usability: After completion of the three test cases in both systems we asked the participants to rate their subjective reactions on the usability of the systems on a five-point Likert scale as required by the SUS methodology. SUS yields a single number representing a composite measure of the usability of a system with scores in the range from 0 to 100, 100 being the best score. Overall ActiveRaUL scored 72.1 out of 100 points compared with 32.5 for WebProtégé, indicating that the participants found ActiveRaUL easier to use than WebProtégé for the creation of ontology instances.

Concluding, our user study proved that ActiveRaUL is indeed easier, more effective and more efficient to use *for the creation of RDF data* than the state-of-the-art ontology editing tool.

References

1. J. Baumeister, J. Reutelshoefer, F. Puppe. KnowWE: a Semantic Wiki for knowledge engineering. *Applied Intelligence*, 35:323–344, 2011.
2. J. Brooke. SUS - A quick and dirty usability scale. In P. W. Jordan, B. Thomas, B. A. Weermeester, A. L. McClelland, editors, *Usability Evaluation in Industry*. Taylor and Francis, London, 1996.
3. S. Corlosquet, R. Delbru, T. Clark, A. Polleres, S. Decker. Produce and Consume Linked Data with Drupal! In *Proceedings of ISWC*, pages 763–778, 2009.
4. T. Tudorache, C. Nyulas, N. F. Noy, M. A. Musen. WebProtégé: A Collaborative Ontology Editor and Knowledge Acquisition Tool for the Web. *Semantic Web* 4(1), 2013.
5. A. Passant, J. G. Breslin, S. Decker. Open, distributed and semantic microblogging with smob. In *Proceedings of ICWE 2010*, pages 494–497, 2010.
6. A. Haller, T. Groza, and F. Rosenberg. Interacting with Linked Data via Semantically Annotated Widgets. In *Proceedings of JIST*, pages 300–317, 2011.