

Toward a Semantic Web editor for the layperson

Chris Donaher, Jesse Hatfield, Jessica Zeitz, Stephen Davies

University of Mary Washington, Fredericksburg VA 22401, USA
(cdonaher/jhatfield/jzeitz/stephen)@umw.edu

Abstract. We present empirical results that illuminate what types of features novices require in Semantic Web tools in order to be most effective. A prototype system was constructed to study the effectiveness of specific features in RDF authoring tools whose intent is to ease the burden of semantic data creation. After comprehensive and focused testing with a small set of users, several interesting and surprising conclusions were reached, some of which include: laypeople tend to imitate sentence structure when forming triples; they struggle with the differences between resources and datatype literals; they consistently attempt to augment their numeric data with units; and they can be encouraged to create domain and range data with the appropriate interface.

1 Introduction

In order for the Semantic Web to gain widespread usage, rank-and-file end users must be able to quickly and simply create RDF data. Unfortunately, this phenomenon is far from realization; the lion's share of Semantic Web tools have a substantial learning curve for the uninitiated, a learning curve which is arguably one of the principal reasons for the sparsity of the Semantic Web.

With this in mind, our investigation used focused empirical testing with a small number of uninitiated users (a technique defended in [1]) to study how such individuals could be expected to interact with a typical Semantic Web authoring system. We studied novices constructing knowledge in a general domain (i.e., one in which they were not domain experts), and we evaluated specific user interface features. This distinguishes our work from previous usability studies (for example, [2, 3]).

The participants were given a brief 10-minute introduction to the system and the principles of RDF triples, then presented with a structured list of tasks to perform. These tasks tested both how effectively users were able to navigate and look up data, as well as how easily and accurately the users could model semantic knowledge. The tasks included items that required traversing multiple links in order to find the requested knowledge, using existing resources as well as creating new ones, and employing reification techniques to express knowledge.

During the course of the experiment, screencast software was used in conjunction with experimenter observation to identify the specific aspects of the system that users found counterintuitive or confusing. After the participants completed the experiment, they were interviewed in order to get feedback on which of the tasks they found difficult, and how those particular tasks could be made easier.

2 Findings

By studying how well the users fulfilled their tasks, we made several key observations regarding how users function in this kind of knowledge environment.

The “sentence-centric” phenomenon. People tended to imitate the exact sentence structure of an item when adding its information to the KB, even including superfluous words. This indicates that one key problem novices will face in creating Semantic Web data will be transitioning from thinking in terms of sentences to thinking in terms of subject-predicate-object triples. One interesting exception, however, was that users sometimes switched to the passive voice in order to place the already existing resource on the left-hand side of a statement. There are varying explanations for this, but it indicates that users do have the ability to depart from the superficial sentence structure and comprehend the deeper meaning of the knowledge therein.

Resources vs literals. When entering data, a user must choose between modeling the object of a triple as a literal value or a resource. We often witnessed users struggling with this decision, voicing aloud that they were ambivalent. Additionally, when questioned in the interviews as to the perceived difference between the two, users exhibited many misconceptions and contradictory notions.

Units. When adding values to datatype properties, many users attempted to add units to their measurements (e.g., several users tried to add the word yards when adding a value for the totalRushingDistance property for American football players). The test system did not provide a way to specify units, which struck multiple users as frustrating and counterintuitive. We were mildly surprised at this, assuming that most users would (as we often do) simply assume an appropriate default unit.

Domain and Range information. Our prototype included a type system that implicitly created domain and range assertions while users were creating instance data. Unlike most systems, where users must explicitly specify the schema before they create instance data, our prototype infers the schema from the instance data as the user adds it. The user can thus freely create new predicates within the context of a resource, and these new predicates will intelligently be used later to prompt the user for data about other resources. Overall, this method of entry was effective, and made it simple for users to create new predicates without having to think at the “schema level” directly.

References

1. J. Nielsen and T.K. Landauer, “A mathematical model of the finding of usability problems,” *Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems*, Amsterdam: ACM, 1993, pp. 206-213.
2. N.F. Noy, W. Grosso, and M.A. Musen, “Knowledge-acquisition interfaces for domain experts: An empirical evaluation of Protege-2000,” *Proceedings of the 12th Internal Conference on Software and Knowledge Engineering*. Chicago, July 5, 2000.
3. S. Staab, A. Maedche, and S. Handschuh, “Creating metadata for the semantic web: An annotation framework and the human factor,” Technical Report 412, Institute AIFB, University of Karlsruhe, 2001.