'Semantic Web 2.0' - write-enabling the Web of Data

Hert, M; Reif, G; Gall, H C

Abstract: The Semantic Web today is mainly a read-only Web of Data. Many of the data sets that contribute to the Semantic Web are not stored as native RDF, but generated on demand via wrappers. Despite the fact that user contribution is the key success factor in the Web 2.0, current wrapper approaches and standardization efforts still focus on read-only data access. In this paper, we argue that the Semantic Web should learn from the evolution of the Web 2.0 and consider write-enabled semantic data wrappers.

Posted at the Zurich Open Repository and Archive, University of Zurich
ZORA URL: https://doi.org/10.5167/uzh-42660

Originally published at:
Postprint available at:
http://www.zora.uzh.ch

Posted at the Zurich Open Repository and Archive, University of Zurich.
http://www.zora.uzh.ch

Originally published at:
'Semantic Web 2.0' - write-enabling the Web of Data

Abstract

The Semantic Web today is mainly a read-only Web of Data. Many of the data sets that contribute to the Semantic Web are not stored as native RDF, but generated on demand via wrappers. Despite the fact that user contribution is the key success factor in the Web 2.0, current wrapper approaches and standardization efforts still focus on read-only data access. In this paper, we argue that the Semantic Web should learn from the evolution of the Web 2.0 and consider write-enabled semantic data wrappers.
Semantic Web 2.0 – Write-enabling the Web of Data

Matthias Hert, Gerald Reif, and Harald C. Gall
Department of Informatics, University of Zurich
{hert, reif, gall}@ifi.uzh.ch

Abstract. The Semantic Web today is mainly a read-only Web of Data. Many of the data sets that contribute to the Semantic Web are not stored as native RDF, but generated on demand via wrappers. Despite the fact that user contribution is the key success factor in the Web 2.0, current wrapper approaches and standardization efforts still focus on read-only data access. In this paper, we argue that the Semantic Web should learn from the evolution of the Web 2.0 and consider write-enabled semantic data wrappers.

The Linking Open Data project\(^1\) is one of the most prominent success stories of applying Semantic Web technologies. A large number of data sets is made available on the Web of Data and is interlinked following the Linked Data principles [1]. This way, Linked Data represents a mainly read-only Web database. The Web of Documents, in contrast, evolved from a read-only medium to the so-called Web 2.0 where user contribution and therefore write access to Web resources play an essential role. Sites such as Wikipedia and Flickr\(^2\) are evidence for the success of the Web 2.0. As a result, the current Web of Data is at a disadvantage compared to the Web of Documents w.r.t. write access.

Tabulator Redux [2] is a Linked Data browser with write capabilities. It addresses user interface concerns as well as how modified data is persisted. Tabulator Redux proposes a network protocol to propagate modified data back to the individual sources. However, this approach does not address the fact that most of the data in the current Web of Data is not stored as native RDF. Instead, the RDF data is generated on demand via wrappers from existing non-RDF data sources. Updating these data sources would require a translation of the modified RDF data into its native format. Two of the most relevant sources of such non-native data are Web 2.0 APIs and relational databases (RDBs).

Web 2.0 APIs restrict data access to a predefined set of operations. Several approaches exist that expose these API operations as Linked Data (e.g. Flickcurl\(^3\) for Flickr), but it is in the nature of such proprietary APIs that their usage differs for each provider. Hence, the community project pushback\(^4\) was started

---

1 http://esw.w3.org/SweoIG/TaskForces/CommunityProjects/LinkingOpenData
3 http://librdf.org/flickcurl/
4 http://esw.w3.org/PushBackDataToLegacySources
with the goal of introducing a unifying wrapper infrastructure for Web 2.0 APIs. It will provide read and write access via RDF-annotated HTML forms, so-called RDForms, and mappings to the native interfaces of the Web 2.0 APIs.

Today, most structured data is stored in RDBs. Exposing this data to the Semantic Web for read-only access has been the focus of many approaches (see [3] for a survey). The World Wide Web Consortium (W3C) has also recognized the importance of mapping relational data to the Semantic Web by starting the RDB2RDF Working Group (WG).\(^5\) Its goal is to unify the existing approaches in a standard RDB-to-RDF mapping language. However, existing approaches as well as the current proposal of the RDB2RDF WG are not suitable for write access to the data as their mappings are, in general, too expressive. Existing mapping languages employ SQL views on the relational schema to define mappings. While this results in a high expressiveness, it also means that these RDF views are, in general, not writable due to the view update problem [4].

OntoAccess [5, 6] is a first approach that avoids this problem by introducing the mapping language R3M. It is based on the ideas described in [7] with several enhancements such as mapping join tables to object properties instead of classes. R3M may therefore be less expressive than existing languages, but R3M was designed to expose normalized and well designed RDB schemata (e.g. ones generated by object-relational mapping tools) to the Web of Data for read and write access and not to fix degenerated RDB schemata.

In summary, demand for write access to Semantic Web data already exists and will increase with the evolution of the Web of Data to a 'Semantic Web 2.0' where user contribution plays an essential role. To foster this evolution, the requirement of write access must be considered explicitly in upcoming approaches and standardization efforts.

References

\(^5\) http://www.w3.org/2001/sw/rdb2rdf/