A truly Read-Write Web for machines as the next generation Web? ⋆

Sam Coppens, Ruben Verborgh, Miel Vander Sande,
Davy Van Deursen, Erik Mannens, Rik Van de Walle

Ghent University - IBBT
Department of Electronics and Information Systems - Multimedia Lab
Gaston Crommenlaan 8 bus 201, B-9050 Ledeberg-Ghent, Belgium
firstname.lastname@ugent.be

Abstract. In this paper, we provide a view on the future Web as a Semantic read-write Web. Given a number of prerequisites for enabling a fully read-write Web for machines, we predict the following. First, datasets and data in general will be driven by machine updates. Second, lots of human-created machine-readable schemas will become obsolete. Third, the Web of Services and the Web of Data will be fully integrated into a Web of read-write Data. Fourth, reasoning on the Web stays monotonic and will be optimized for versioned data. Additionally, two application areas are discussed where the read-write Semantic Web will have a deep impact: the Web of Things and advanced personalisation.

1 Introduction

There are indications for an industrial uptake of a number of Semantic Web technologies now that big players, such as Google, IBM and Facebook, have adopted some of these technologies [7]. Today, it seems that we are still in the phase of publishing existing or new data as structured data on the Web, so that it can be *read* by machines. The current approaches for publishing structured data on the Web usually provide just read-only interfaces to the underlying data. This is also reflected in today's Linked Data applications, which are limited to smart search engines allowing more complex queries, mashups integrating similar data, or classic search engines enhancing their results.
The Web started as a read-only version (i.e. Web 1.0), evolved to a read-write Web where humans can collaborate (i.e. Web 2.0), and is currently a Web powered by semantic technologies so that machines can read and understand the underlying data (i.e. Web 3.0). Given the evolution of the Web, a logical next step towards Web 4.0 seems to be a read-write Semantic Web, where machines can collaborate.
Several proposals exist to enable machine-writable structured data. For instance, Tim Berners-Lee himself proposed update protocols based on

⋆ The research activities as described in this paper were funded by Ghent University, the Interdisciplinary Institute for Broadband Technology (IBBT), the Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT), the Fund for Scientific Research-Flanders (FWO-Flanders), and the European Union.

WebDAV or SPARQL/Update [2] and implemented them in Tabulator. Lanthaler *et al.* tried to obtain a read-write Web of Data by integrating Web APIs into it (in the form of JSON services) [6]. Recently, W3C started a standardization activity to create a Linked Data Platform, with as main goal to define a RESTful way to read and write Linked Data [3].
We are convinced that enabling writable Linked Data on the Web will have a deep impact on the use of Linked Data and the Semantic Web in general. While machines are currently limited to read/query and simple reasoning tasks, a whole range of new applications will become available. It is important to note that a read-write Web for machines not only implies that machine-readable data can be manipulated by humans. More specifically, machines should be able to autonomous understand/discover the possible operations on the available machine-readable data.
In this paper, we discuss the possible impact of a truly read-write Web of Data on different aspects of the Semantic Web. Additionally, we discuss two application areas where a read-write Semantic Web will have important influences.

2 Prerequisites

In this section, we discuss a number of items that can be considered as prerequisites for a fully enabled read-write Semantic Web. We consider these items as existing during the discussion of the impact on parts of the Semantic Web and application areas.

Read-write Linked Data principles The current Linked Data principles [1] tell us how to make data available so that machines can read it. These principle have to be extended with information about how to make data accessible, so that machines can modify it (preferably in a RESTful way). If all goes well, W3C's Linked Data Platform Working Group will provide and standardize such principles [3].

Data provenance and trust Currently, most machine-readable data sets are created (scraped, transformed, etc.) from one existing source on the Web. Therefore, today it is in most cases quite trivial to determine the provenance of machine-readable data. With a fully enabled read-write Semantic Web, determining the data provenance will be less trivial. Therefore, it is important to have good support for modeling and determining the provenance of machine-readable data. Currently, the W3C Provenance Working Group is defining a data model for provenance on the Web and will provide guidelines to access information about the provenance of resources on the Web [4]. Provenance plays an important role in obtaining trust on the Semantic Web. In particular, it should make trust measurements (using whatever metric) transparent for machines (i.e. reproducible). This way, independent parties (i.e. so-called proof checkers or provenance checkers) can judge impartially whether the given data is trustworthy.

Data permissions With a writable Semantic Web, humans and machines will have concerns about what will happen with their data: who will be able to read it, is the data reusable, what parts of my data are writable by others, etc. In fact, the problem of data permissions in a machine-writable Semantic Web is threefold. First, machines should be able to determine the source of the data they want to write. Additionally, they should know whether they are allowed to reuse this data (in case the data is coming from an existing source). Second, machines should be able to determine if they are allowed to write their data in the targeted place. Finally, they should be able to discover/enforce who has read/write access to their data once it is written.

Functional descriptions In order to give machines access to a read-write Semantic Web, they should be able to determine which operations can be executed on a given resources (i.e. what functionality offers a resource). Therefore, next to provenance and data permission information, resources should also publish descriptions providing more information about possible functional operations (i.e. the API). A functional description model for the read-write Semantic Web should possess the following characteristics:

- support REST APIs;
- explain the functionality of the API in a machine-processable way;
- allow composition of any number of APIs.

Technologies such as RESTdesc [9] and Linked Data Services (LIDS, [8]) are valuable starting candidates for such a functional description model.

3 Semantic Web Impact

In this section, we discuss the expected impact of a fully enabled read-write Semantic Web on different aspects of the existing Semantic Web (i.e. datasets, schemas, Web services, and reasoning).

3.1 Datasets

Today, datasets within the Semantic Web are mostly static dumps of information (especially within the Linked Open Data cloud). Moreover, these datasets are not perfect: they contain errors, are not up to date in some cases, and lack links to other datasets. Enabling write access for machines on datasets implies that machines will be able to correct, update, and extend these datasets. Note that, in an ideal world, the human-readable Web is a possible view on the machine-readable Web. This means that, when machines are able to keep the underlying datasets up to date, the human-readable Web will be up to date as well. This way, datasets and data in general on the Web will be driven by machine updates. Indeed, lots of machine agents will, for their specific needs, interpret data and add/correct information where necessary.

3.2 Schemas

Schemas (or ontologies) are today even more static than datasets. Domain experts discuss carefully each concept and relationship in such a schema. This brings a number of problems such as the gap between a domain expert and an ontology engineer, schema updating is not trivial, and a lot of schemas are application-specific. In the future, we believe that machines (who will become more and more intelligent and autonomous) will be able to update and correct schemas according to their needs. Machines will be smart enough to build their schemas on the fly (i.e. at run-time), based on the incoming data and tailored to the target application.

3.3 Web Services

Recently, the number of Web APIs has been increasing at a tremendous rate, since they offer a practical alternative to the more heavy-weight Web services. Ideally, we want Web APIs to behave like Linked Data, where data from different sources can be combined in a straightforward way. In a read-write Semantic Web, there will be no distinction anymore between a Web resource and a Web API: each Web resource will have its API. Today, people are shifting from heavy-weight RPC-style Web services to more lightweight Web APIs (often claimed to be RESTful). We believe that, in the next 10 years, there will be a shift from function-oriented Web APIs to resource-oriented Web APIs (i.e. truly RESTful APIs). In the end, classic (heavy-weight) Web services as we know them today will disappear from the Web. However, their functionality will still be present behind the scenes (thanks to a RESTful design) when portions of information are being read or written.

3.4 Reasoning

Most reasoning tasks (whether DL- or rule-based) today are executed on static data sets. With an enabled read-write Semantic Web, structural data will be more and more dynamic. While for some people this might be a reason to use nonmonotonic logics to solve future Web inferencing tasks, the reasoning on the Web will always be monotonic [5]. In order to keep realizing the latter, changes in the data should be carefully archived, including timestamps (cfr. discussion on provenance in Sect. 2). This way, inferencing and proof checking tasks will always be run on a 'snapshot' of the Semantic Web, keeping the reasoning monotonic in a non-monotonic environment. Therefore, we believe that Semantic Web reasoners keep being monotonic but will be optimized to deal with timestamp-based (i.e. versioned) data.

4 Application Areas

Next, we discuss two application areas where we believe that the read-write Semantic Web will play a crucial role: the Web of Things and advanced personalisation.

4.1 Web of Things

The Web of Things vision implies that everyday devices and objects all become information resources on the Web. Information coming from these 'things' has been identified as a major driver in many domains, from smart cities to environmental monitoring to real-time tracking. Other applications are scanning barcodes or RFIDs of real-world things and get information about it (on condition that there exist a mapping between the scanned code and a URI on the Web). Some things such as temperature sensors are read-only, other things can be fed with input (e.g. a light that can be switched on and off). If we model and handle these things in exactly the same way as we model and handle (Semantic) Web resources (i.e. using read-write Linked Data principles), we obtain a fully integrated Web with a uniform interface. This way, machines will not only be able to collaborate with each other based on classic Web data, but also based on in- and output of real-world Things connected to the Web.

4.2 Advanced Personalisation

Everybody can share any piece of information with anybody from anywhere, anytime. However, the more information is being published on the Web, the more scarce the attention for the information will become. This information overload can be managed by filtering only that information that lies in the interests of the end-user. The information becomes then more targeted and less abundant and, as a consequence, will get a lot more attention.

With the emerging Social Web, many end-users of the Web already have a user profile in one or more social networks, e.g., Facebook, Twitter, NetFlix, etc. These user profiles hold a lot of information on the interests of the end-user. However, these profiles are only applicable within the closed silo of the social network, while the real power of personalisation can be obtained by recommendations based on combining the profile information beyond the social network boundaries. This is where Semantic Web agents and writable Linked Data comes into play. Indeed, the end-user can then manage and publish its own profile information as writable Linked Data and machines can read and update this profile information (note that for instance Facebook could offer this service in the future).

Social networks (and actually any machine agent) will be able to use and update this profile information (at least if they are authorised to do so). These updates may include updates of static information of the end-user (e.g. age, gender, address, current job), updates in the end-user's social graph (e.g. new friends), and updates in the user's activity stream (e.g. Facebook likes, click behavior on NetFlix, tweets, etc). The benefit of this writable profile information is that an end-user would only require one profile (which he manages himself) that has much richer information (aggregated over the application boundaries) and every machine agent will be able to use this (or parts of) profile information.

5 Conclusions

Considering the evolution of Web generations, we discussed in this paper the impact of a fully enabled read-write Web for machines. Taking into account a number of prerequisites for enabling a read-write Semantic Web, we investigated the possible impact on different aspects of the Semantic Web. We can conclude with the following statements (each linked to a different aspect of the Semantic Web):

- datasets will be driven by machine updates;
- machine-readable schemas created by humans will become obsolete;
- the Web of Services and the Web of Data become a Web of read-write Data;
- reasoning on the Web stays monotonic and will be optimized for versioned data.

Additionally, two application areas were selected that will get a boost from the read-write Semantic Web: the Web of Things and advanced personalisation. For the former, the line between dataset resources, Web APIs, and Things will disappear for machines; for the latter, self-organizing user profiles will drive personalisation applications.

References

1. T. Berners-Lee. Linked Data. `http://www.w3.org/DesignIssues/LinkedData.html`, June 2009.
2. T. Berners-Lee. Read-Write Linked Data. `http://www.w3.org/DesignIssues/ReadWriteLinkedData.html`, October 2011.
3. A. Bertails, S. Hawke, and I. Herman. Linked Data Platform (LDP) Working Group Charter. `http://www.w3.org/2012/ldp/charter.html`, May 2012.
4. Y. Gil, J. Cheney, P. Groth, O. Hartig, S. Miles, L. Moreau, and P. Pinheiro da Silva. Provenance Interchange Working Group Charter. `http://www.w3.org/2011/01/prov-wg-charter`, April 2011.
5. P. Hayes and E. Franconi. Why must the web be monotonic? W3C www-rdf-logic mailing list thread, starting at `http://lists.w3.org/Archives/Public/www-rdf-logic/2001Jul/0065.html`, July 2001.
6. M. Lanthaler and C. Gutl. Aligning Web Services with the Semantic Web to Create a Global Read-Write Graph of Data. In *Proceedings of the 2011 IEEE 9th European Conference on Web Services (ECOWS 2011)*, pages 15–22, Lugano, Switzerland, September 2011.
7. J. Sequeda. The Semantic Web Has Gone Mainstream! Wanna Bet? `http://semanticweb.com/the-semantic-web-has-gone-mainstream-wanna-bet_b27329`, March 2012.
8. S. Speiser and A. Harth. Integrating Linked Data and services with Linked Data Services. *Lecture Notes in Computer Science – The Semantic Web: Research and Applications*, 6643:170–184, May 2011.
9. R. Verborgh, T. Steiner, D. Van Deursen, S. Coppens, J. Gabarró Vallés, and R. Van de Walle. Functional descriptions as the bridge between hypermedia APIs and the Semantic Web. In *Proceedings of the Third International Workshop on RESTful Design*, pages 33–40, Lyon, France, April 2012.