PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A

rsta.royalsocietypublishing.org

Discussion

Cite this article: Berners-Lee T, O'Hara K. 2013 The read–write Linked Data Web. Phil Trans R Soc A 371: 20120513.
http://dx.doi.org/10.1098/rsta.2012.0513

One contribution of 15 to a Discussion Meeting Issue 'Web science: a new frontier'.

Subject Areas:
human-computer interaction

Keywords:
Web science, linked data, Web architecture

Author for correspondence:
Kieron O'Hara
e-mail: kmo@ecs.soton.ac.uk

The read–write Linked Data Web

Tim Berners-Lee[1] and Kieron O'Hara[2]

[1]Computer Science and AI Laboratory, Massachusetts Institute of Technology, Vassar Street, Cambridge, MA 02139, USA
[2]Web and Internet Science, Electronics and Computer Science, University of Southampton, Highfield, Southampton SO17 1BJ, UK

This paper discusses issues that will affect the future development of the Web, either increasing its power and utility, or alternatively suppressing its development. It argues for the importance of the continued development of the Linked Data Web, and describes the use of linked open data as an important component of that. Second, the paper defends the Web as a read–write medium, and goes on to consider how the read–write Linked Data Web could be achieved.

1. Introduction

For some years, people have been asking whether and when the Semantic Web [1,2] was going to take off, yet in the guise of the Linked Data Web [3,4] we see it growing all the time. The Web continues to develop from a medium for publishing textual documents into a medium for sharing structured data (which was after all the point of the Semantic Web). The Linking Open Data project[1] has been monitoring the development of the Linked Data Web since 2007, which by 2011 had grown to a size of about 32 billion RDF triples. DBpedia was the first major effort to publish linked data, but now contributions are coming increasingly from companies, governments and other public sector bodies such as libraries, statistical bodies or environmental agencies. In parallel, Google, Yahoo! and Bing have established the schema.org initiative, a shared set of schemata for publishing structured data on the Web that focuses on vocabulary agreement and low barriers of entry for data publishers.

[1]http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData.

Royal Society Publishing
Informing the science of the future

In this paper, we consider some of the issues that will help the Linked Data Web continue to grow and develop its scientific and social utility. In the next section, we consider recent moves to populate the Web with linked open data. Then, we discuss how the Data Web can become a medium for writers as well as readers.

2. Linked open data

Once data is placed in a file using the W3C standards, it can already be processed, for example putting it in a spreadsheet, or (if it has a geolocation) on a map. It can also be linked in two ways. First if the data was about a particular object X, it could be linked with other data about X. Such a link helps with disambiguation if X's name denotes other things too; for instance, if your data was about Copenhagen the city, then you could link it to other data about the city, rather than data about Copenhagen the play, the song by the Wolverine Orchestra, the football team or the interpretation of quantum mechanics. Second, the properties that the data expresses can be linked to the same properties in other data; for instance, if the data gives the population of Copenhagen, we can specify that by 'population' is meant what it means in DBpedia. The links are made by using universal resource identifiers (URIs) across the two datasets. By following the links made in this way, Linked Data allows the user to gather large quantities of data about related things, or about related concepts [5].

The Web of Linked Data has certainly been growing. DBpedia, a collection of data, was formed by taking structured data out of Wikipedia, and making it available on the Web by linking it to other relevant data. The links have grown as more public datasets have been published in linked form, and by 2009 (see for example Berners-Lee's TED talk of that year[2] focusing on the need to publish and link data) the Linked Data Web had billions of pieces of data, with metadata stored by the Comprehensive Knowledge Archive Network (CKAN) archive [6], which in turn is administered by the Open Knowledge Foundation.[3]

The growth of the Linked Data Web shows how the Semantic Web principles aid the expression of data and the acquisition of new and related data, enabling the creation of much more value. For example, an app about football could exploit open linked data from the Web about stadiums, locations of matches, famous footballers, previous games in the competition or the weather. Adding these to the app will make it a source of much richer information.

3. Read–write data

Some think that the future of media lies with phone apps, but a major issue with these is that the content is not on the Web. There is a huge difference with a Web app, which allows users to share URIs to get directly to underlying data sources, exploiting the added value of putting content on the Web. There is nothing wrong with phone apps *per se*, but none of the information in a phone app can be linked to. They are not participating in the new world of people linking things together, so for example, search engines will not find their content. There are guidelines for the creation of Web apps, to enable their function on all kinds of devices, from the Mobile Web Initiative at the W3C.[4] In this section, we sketch an infrastructure for Web apps, and consider issues pertaining to 'webizing' data. The basic plan for webizing data is to take the identifiers in a system and turn them into URIs; not all systems would survive such a change, but those that do will become much more powerful.

When the Web was created, the idea was for a read–write Web. The original browser was actually an editor; if you had the right access, you could change the text or add links to other

[2]http://www.ted.com/talks/tim_berners_lee_on_the_next_web.html.

[3]http://okfn.org/.

[4]http://www.w3.org/Mobile/.

places on the Web. The idea of hypertext trails was an extremely important idea to facilitate collaboration [7], but the Web took off primarily as a publishing medium. Now there are more outlets for writers—one can write a blog or contribute to a wiki or OpenStreetMap—but there is a need for more modalities for interactivity. If the Web is genuinely to be a read–write Web, and if we are to pursue a Web of Data, then it is essential that the Data Web should not be read-only.

What would that look like? To begin thinking about this, consider the siloed data in social networks. If you go into one social networking silo, it is impossible to refer to friends in another one. For example, if I want to make a set of photographs in one social network available to a group of colleagues that I have carefully set up in another network, I cannot. The network page has a URI, so can be pointed to, but this is not sufficient. The issue is not that only some people can read the page (that is perfectly reasonable in social networking), but rather that the page is generated from powerful, useful data which could be a valuable resource if only we had access to it. For example, the network will be aware not only of, say, the tags of a photo, but also some semantic content. When the social network has prompted the user for information about the photo, it asked not for tags or other syntactic markers, but instead it (typically) will have asked *who is in the photo?* In other words, it will have posed a question that created *semantic* information enabling, for example, the generation of sets of photos not only of the user, but also of all the photos that his or her friends are in. The semantics enable these more complex queries.

Each site contains lots of data of this type, but the silos mean it cannot be linked. You cannot use your favourite photo software without logging in (and thereby being exposed to adverts, etc.), even though they are your photos and it is your metadata about which friends are who. So, for instance, if Alice tweets on Twitter, it goes into the Twitter system (even though Twitter allows access to tweets, they still have to be found within Twitter). This is somewhat frustrating, and drives recent thinking about how to implement a read–write Web of Data.

Work is being done to make social networking work in a Web-like fashion, so anybody can follow anybody across platforms. Hence, for example, the creation of identi.ca, a social networking and microblogging site similar in conception to Twitter, but that provides extra features including free export of personal data and data about friends based on the FOAF standard, and which runs off the status.net open source code base. If Alice tweets on identi.ca, the tweet goes into her Twitter feed. But even though such sites are moving in the right direction, they often use application programming interfaces, which are an extra layer of complexity and require the writing of some code—which remains an impediment to the webized free flow of data.

However, as we are talking about data, why not do the simple thing and use existing data standards? Everyone who stores tweets could actually surface them as linked data, using an architecture of the sort shown in figure 1. If someone looks up, say, Bob, they can be pointed at Bob's tweets. Identi.ca, which is designed to produce linked data, allows the storage of a canonical URI, so Bob could store a pointer to a canonical version of himself which could be reached by anyone accessing him via identi.ca. The pointer allows links, indicating that the 'Bob' on identi.ca is the same person as the one identified by the canonical URI, which then allows the material on identi.ca to be linked to Canonical Bob. In other words, the visitor to identi.ca is not kept in the walled garden, but is able to reach other views or other material on different sites about the person being investigated. From the canonical representation, it is possible that other personae of Bob are listed, accessible and linked to; hence from identi.ca Bob, Bob's other manifestations or representations are reachable.

The architecture in figure 1 has a 'double bus' arrangement. The top 'bus' is the Web, with URIs using HTML and other protocols, whereas below are the data standards of the Linked Data Web. The top layer supports the URIs for webpages in a Web application. These pages dip into the Web of Data to gather the requisite data content, allowing the Data Web to act as an infrastructure for Web apps. The problem with phone apps is that you cannot get at the underlying data, but with the arrangement shown in figure 1 not only do you get the app's particular view of the data, but also you get access to the underlying data, the calendar, the photo store or the event stream (events are very powerful organizers of data, as argued by Jain [8], connecting people, places and times), leading to the arrangement shown in figure 2.

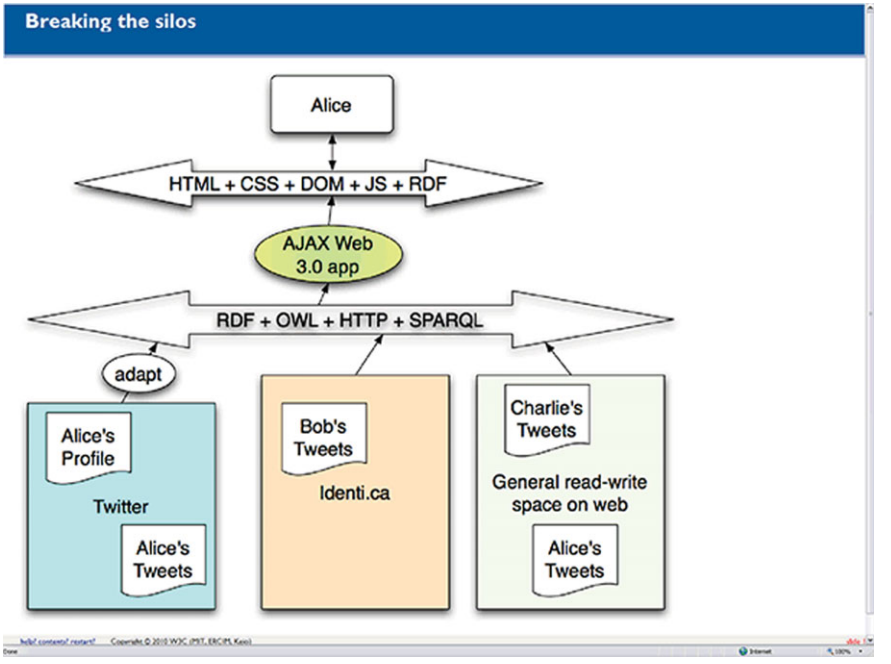

Figure 1. A Web architecture to break the silos. (Online version in colour.)

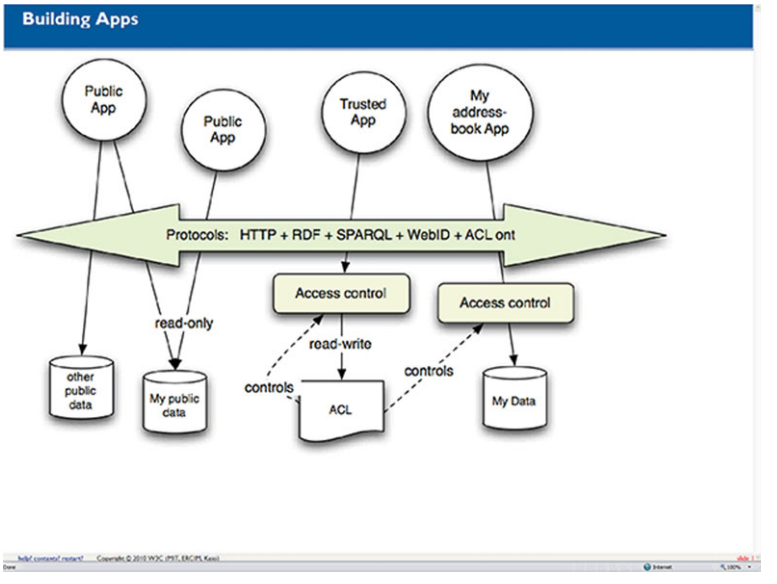

Figure 2. An architecture for read–write access-controlled linked data. (Online version in colour.)

In figure 2, apps—Web apps, trusted apps on a computer, public apps or apps using a person's credentials on his or her machine—sit on top. With systems such as these, with their assumptions about social context as well as technology, it is clearly important to control access. Hence, this is properly thought of as *read–write access-controlled linked data*, or, put another way, *socially aware cloud storage*. The structure is analogous to UNIX storage burst open onto the Web, functioning at Web scale.

The technology required is largely in existence. For identification, there is WebID, the use of a URI to identify a person in such a way that basic profile information about them can be looked up on the Web as Linked Data. For authentication, one protocol is WebID/TLS, formerly

known as FOAF + SSL, a decentralized secure authentication protocol that links (as its former name implies) the security created by the Secure Socket Layer with the profile information supported by FOAF. Authorization uses RDF. The write part of the data Web is covered by Web Distributed Authority and Versioning (WebDAV), an extension of HTTP which allows remote users to collaborate. SPARQL/Update will express changes or updates of data stores without having to transfer the entire file that has been changed. What is needed to supplement this suite of technologies is a library of 'widgets' (generic and portable pieces of software) to support the writing of apps, and application building tools.

The point of this architecture is to allow users to write their own applications, bringing in the data they need easily and straightforwardly, for instance moving from their own calendars to the calendars of colleagues, maps and geographical information, room booking systems, events pages and so on. On this model, someone creating a social networking site would not follow the usual practice of building a giant store to hold everyone's data. Instead, they would sell an app cheaply to bring data together. The user would pay a small amount for access-controlled storage in the cloud, and keep control of the data himself or herself.

Access control would be done entirely by read–write linked data. Web groups would be identified by a URI, which could then be used to drive access. Even quite unusual groups—for example, the set of people at a particular conference—could be given a URI, and then it would be simple to set access to a set of documents only for those people referred to by the URI. The list of names might be public or secret (so that only a restricted set of people would know who had access to the data), but that type of secrecy is independent of the question of *who has* access. The URI could be used anywhere, so any social networking site on this open model could specify that certain documents could only be accessed by members of the group identified by the URI; authorization need no longer be relative to groups defined on particular sites. This is one of the keys to breaking open social network data silos.

This is only a brief review, and clearly there are issues to be resolved with this kind of open architecture. We need to ensure that protocols about how to write to this read–write Web are robust against attempts to undermine it, for example by identifying and suppressing spam. Security in general, especially in the context of personal data, will need to be very carefully thought out. But ultimately, the architecture for read–write access-controlled linked data is a matter of webizing the Unix storage system (i.e. replacing the string identifiers with URIs), building on existing technologies. This is not an impossible task, and such an architecture should help open apps and commodity pricing of data storage emerge.

References

1. Berners-Lee T, Hendler J, Lassila O. 2001 The semantic web. *Sci. Am.* **284**, 34–43. (doi:10.1038/scientificamerican0501-34)
2. Shadbolt N, Berners-Lee T, Hall W. 2006 The Semantic Web revisited. *IEEE Intell. Syst.* **21**, 96–101. (doi:10.1109/MIS.2006.62)
3. Bizer C, Heath T, Berners-Lee T. 2009 Linked data—the story so far. *Int. J. Semantic Web Inform. Syst.* **5**, 1–22. (doi:10.4018/jswis.2009081901)
4. Heath T, Bizer C. 2011 *Linked data: evolving the web into a global data space*. San Rafael, CA: Morgan & Claypool. See http://linkeddatabook.com/editions/1.0/.
5. Berners-Lee T. 2006–10 Linked data. W3C. See http://www.w3.org/DesignIssues/LinkedData.html.
6. Dietrich D, Pollock R. 2009 CKAN: apt-get for the debian of data. In *26th Chaos Communication Congress, Berlin, Germany, 27–30 December 2009*. See http://events.ccc.de/congress/2009/Fahrplan/events/3647.en.html.
7. Bush V. 1945 As we may think. *Atlantic Magazine*. See http://www.theatlantic.com/magazine/archive/1945/07/as-we-may-think/303881/.
8. Jain R. 2013 EventWeb: towards social life networks. *Phil. Trans. R. Soc. A* **371**, 20120384. (doi:10.1098/rsta.2012.0384)