ProvenanceJS: Revealing the Provenance of Web Pages

Groth, P.T.

published in
Lecture Notes in Computer Science
2010

DOI (link to publisher)
10.1007/978-3-642-17819-1_34

document version
Publisher's PDF, also known as Version of record

ProvenanceJS: Revealing the Provenance of Web Pages

Paul Groth

VU University Amsterdam
De Boelelaan 1081a, 1081 HV, Amsterdam, The Netherlands
pgroth@few.vu.nl

Abstract. Web pages are regularly constructed through combining content from multiple providers (e.g. photos from Flickr, quotes from the New York Times). As a result, it is often difficult for users and programmers to retrieve the provenance of a web page. Here, we present a JavaScript library, ProvenanceJS, that allows for the retrieval and visualization of the provenance information within a Web page and its embedded content. A key contribution is to demonstrate that provenance can be supported using widely deployed browser-based technologies.

There has been a rapid proliferation of content sharing on the Web. Sites such as Flickr, Slideshare.net, and YouTube make it easier to find and then integrate images, video, and documents into web pages. Additionally, the cultural of the Web, in particular the blogsphere, thrives on quoting and re-quoting information. Because of this mash-up culture and infrastructure, most web pages consist of content originating from multiple sources. Thus, when viewing a web page it is often difficult to determine where its content came from and how it was produced. This lack of provenance is seen as a critical issue in both the provenance and Web communities as highlighted by the start of the W3C Provenance Incubator Group and its recently produced report on requirements for provenance on the Web [3]. In particular, provenance is one of the most import features users rely on when determining whether to trust a Web page [4]. Indeed, Tim Berners-Lee envisioned an "Oh, yeah?" button within Web browsers that when clicked on would produce reasons why the user should trust the web page based on its provenance [1].

To move towards the realization of such an "Oh, yeah?" button that is widely distributed, we have developed a library, ProvenanceJS[1], that allows for the retrieval and visualization of the provenance of a web page. There are two key contributions stemming from ProvenanceJS:

1. Browser-based technologies are capable of retrieving and rendering provenance information without the need for additional software installation.
2. Embedding provenance information within content is a viable approach for ensuring that the provenance information is available.

[1] Source available at: `http://code.google.com/p/opmv/source/browse/#svn/trunk/js`

D.L. McGuinness, J.R. Michaelis, and L. Moreau (Eds.): IPAW 2010, LNCS 6378, pp. 283–285, 2010.

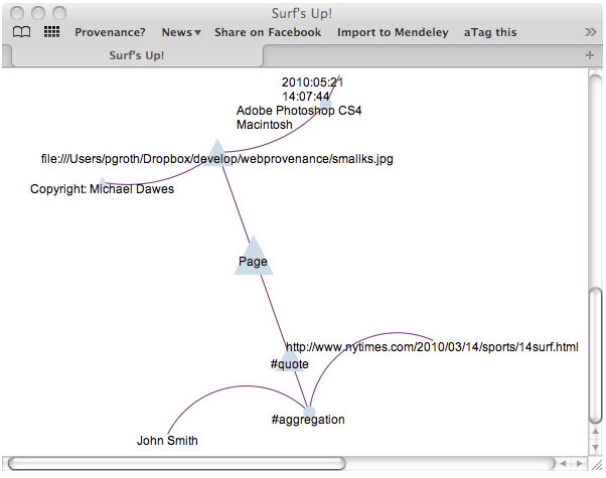

Fig. 1. A visualization of the provenance of a Web page

1 Provenance Metadata and Implementation

In order to make provenance apparent to the user, ProvenanceJS must retrieve provenance information from a Web page. It can acquire this information either from interrogating the page's metadata, extracting the metadata of the embedded content (e.g. an image), or by consulting an outside service that maintains the provenance. Because our aim was to develop a browser-based solution, we chose to focus on the first two sources.

From a page's markup, ProvenanceJS extracts RDFa metadata. RDFa is a widely adopted standard for embedding structured data within web pages. ProvenanceJS recognizes RDFa published using the Open Provenance Model Vocabulary [9]. This vocabulary is an RDF realization of the Open Provenance Model (OPM) [8] with a number of extensions and is being actively developed to help address the needs of data.gov.uk. Using this vocabulary, publishers can markup their data with explicit statements about the provenance of the various parts of their page.

While explicit provenance metadata within Web pages is advantageous, many times it is not practically feasible to provide it explicitly. To address this concern, ProvenanceJS aims to extract provenance metadata from a page's content. For example, ProvenanceJS can extract information from the EXIF metadata found within JPEG images.

ProvenanceJS is implemented entirely in Javascript using the Javascript InfoVis Toolkit, rdfQuery, and exif.js. In addition to the extraction of the metadata described above, it provides an API for building and manipulating OPM Graphs and visualizing those graphs. A bookmarklet ('Provenance?') is included, which visualizes the current web page's provenance. An example is shown in Figure 1. Triangle nodes are artifacts. Circle nodes are processes. It shows how the quote

on a page was generated by an aggregation process controlled by John Smith. In addition, it depicts that the image was modified by Adobe Photoshop and that the copyright of the image belongs to Michael Dawes. The bookmarklet is a first step towards a true "Oh, yeah?" button.

2 Related Work and Conclusion

Moreau provides an extensive review of the provenance literature from the perspective of the Web [7]. A number of authors have considered provenance on the Semantic Web. In particular, Bizer et al. present a Semantic Web based policy framework for information quality [2]. It included an implementation of the "Oh, yeah?" button. However, this implementation required a browser plug-in. We see ProvenanceJS as building on-top of such existing Semantic Web approaches. Margo and Seltzer showed how by treating user interaction with a Web browser as provenance, novel search functionality could be realized [6]. The closest work to ProvenanceJS is the Provenance-Embedding Document approach [5]. This approach uses Javascript to extract provenance from RDFa metadata. Our work differs in that we support the extraction of provenance from embedded content and use a community driven provenance vocabulary.

ProvenanceJS can be used to retrieve and visualize the provenance of a web page using only browser-based technology, namely Javascript. Additionally, provenance metadata from page markup and embedded content can be integrated to provide a full view of provenance.

References

1. Berners-Lee, T.: Cleaning up the User Interface (1997), `http://www.w3.org/DesignIssues/UI.html`
2. Bizer, C., Cyganiak, R.: Quality-driven information filtering using the WIQA policy framework. Web Semantics: Science, Services and Agents on the World Wide Web 7(1), 1–10 (2009)
3. Cheney, J., Gil, Y., Groth, P.E., Miles, S.: Requirements for Provenance on the Web (2010), `http://www.w3.org/2005/Incubator/prov/wiki/User_Requirements`
4. Gil, Y., Artz, D.: Towards content trust of web resources. Journal of Web Semantics 5(4), 227–239
5. Jones, H.C.: XHTML documents with inline, policy-aware provenance. M. eng., Massachusetts Institute of Technology (2007)
6. Margo, D.W., Seltzer, M.: The Case for Browser Provenance. In: 1st Workshop on the Theory and Practice of Provenance, TaPP 2009 (2009)
7. Moreau, L.: Foundations of Provenance on the Web. Foundations and Trends in Web Science (2009) (submitted)
8. Moreau, L., Clifford, B., Freire, J., Futrelle, J., Gil, Y., Groth, P., Kwasnikowska, N., Miles, S., Missier, P., Myers, J., Plale, B., Simmhan, Y., Stephan, E., den Bussche, J.V.: The open provenance model — core specification (v1.1). In: Future Generation Computer Systems (July 2010)
9. Zhao, J.: Guide to the Open Provenance Model Vocabulary (2010), `http://purl.org/net/opmv/guide`