Data Republishing on the Social Semantic Web

Claudia Wagner[1,2] and Enrico Motta[2]

[1] Institute for Networked Media, JOANNEUM RESEARCH,
Steyrergasse 17, 8010 Graz, Austria
claudia.wagner@joanneum.at
[2] Knowledge Media Institute, The Open University,
Walton Hall, Milton Keynes, MK7 6AA, United Kingdom
e.motta@open.ac.uk

Abstract. Data Republishing is a recent Social Web phenomenon which can be observed in different areas of the Social Web. However, current Data Republishing tools don't work in the emerging context of the Semantic Web. In particular, these tools neither generate any semantic metadata which provide information about the republished content (e.g., provenance information) nor are they able to make use of existing semantic metadata annotating the original content being republished. In this work we introduce the concept of Semantic Data Republishing and describe how to implement it.

1 Introduction

1.1 Motivation

Data Republishing is a recent Social Web phenomenon which can be observed in different areas of the Social Web, such as the blogosphere, the microblogosphere or the social networking sphere. Data Republishing refers to the process in which a user, knowing that data are already published on the Web, rereleases them in a new context. Users for example republish data by *reblogging* external content on their blogs, by *retweeting* microblog posts from other users on their own microblog or by *posting* external content to their Facebook[3] wall. A new kind of republishing oriented Social Web application, so-called tumblelogs, has recently emerged from this trend. Tumblelogs are blogs with shorter posts and mixed media types which are usually less structured than classical blogs [19]. Users can quickly share their online discoveries by republishing multimedia content, found on the Web, on their tumblelogs. Tumblelog providers, such as tumblr[4] and soup[5], gain in importance thanks to their increasing number of unique visitors[6].

[3] http://facebook.com
[4] http://tumblr.com
[5] http://soup.io
[6] http://siteanalytics.compete.com/soup.io+tumblr.com/?metric=uv

Current Data Republishing tools, such as Tumblr Share[7] , ShareThis[8] or Zemanta Reblog[9], support users in republishing their online discoveries on Social Web applications. These tools allow users to select data on any web page, generate a new data item on their preferred target web application, transfer the selected data as text or binary data and use them as content of the new data item. However, a limitation of this approach is that no semantic metadata are generated - e.g., to expose the provenance of the copied data. That means that the information about the republishing process (i.e., who republished, when, from which source application, which fragments of data on which target application) is lost. Another drawback of current Republishing tools is that they are not able to make use of existing semantic metadata which may annotate the original data being republished. Consequently, these tools do not fully support the next generation of Social Web applications, so-called Social Semantic Web applications, which expose the semantics of their data in a machine-interpretable way by using ontology-based metadata.

In this paper we illustrate the need of a new kind of Republishing tool for the Social Semantic Web. We introduce the concept of Semantic Data Republishing and discuss requirements and functionalities of tools implementing this concept in section 2. An initial implementation of an example prototype implementing Semantic Data Republishing is presented in section 3. Finally, in section 4 and 5 we discuss related work stemming from the areas of data publishing and Data Portability on the Social Semantic Web and outline new opportunities for research and development made possible by it.

1.2 Data Republishing on the Social Semantic Web

Two different methods for Data Republishing across individual web sites can be distinguished: (1) Data Republishing by copying data values and (2) Data Republishing by copying data references.

(1) Data Mobility standards (e.g., RSS 1.0, RSS 2.0, Atom, OPML) facilitate Data Republishing by copying data values [9]. Thanks to Data Mobility initiatives structured data can be republished on individual websites without the need to implement application-specific Programming Interfaces (APIs).
(2) Linked Data Design Principles[10] provide data access by reference. Hence, data published according to these principals can be republished and reused by reference. Social Semantic Web applications can reference and dereference resources by using their URIs, access their machine-interpretable descriptions and republish data without the need to copy data values.

Both methods, i.e. Data Republishing by reference and Data Republishing by value, are important for different scenarios.

[7] `http://www.tumblr.com/goodies`
[8] `http://sharethis.com`
[9] `http://zemanta.com/reblog`
[10] `http://www.w3.org/DesignIssues/LinkedData.html`

If data are republished by copying their values both, the source and the target application, store an individual instance of the same data. These instances can then be changed individually. Therefore this technique is suited for situations in which users want their republished data to be independent of the original data (e.g., because users do not want the republished data to change, if the original ones change or because the original data may not be available for long).

If data are republished by reference, the source and the target application point to the same data instance. In this scenario, if the source or target application modifies the data, the data being displayed change on both applications. Therefore this approach ensures that in situations where data are likely to be modified (e.g., in the context of a wiki page) the republished data and the original data are kept in sync.

The advantage of exploiting Semantic Web technologies in the context of the current republishing phenomenon of the Social Web is that the two aforementioned republishing methods can be integrated to combine the advantages of both approaches. In particular by applying Semantic Web technologies to Data Republishing data can be cached on the target application to increase the availability of republished data and can in addition be updated at certain intervals by using the semantic metadata of the republished data to formulate queries.

2 The Design of a Semantic Republishing Tool

2.1 Requirements

A Semantic Republishing tool should allow users to select content from any web site and republish it on their preferred Social Web or Social Semantic Web application (e.g., their blog, their Facebook wall). To exploit the full potential of the Social Semantic Web in the context of the current Republishing trend we have identified the following main requirements for Semantic Republishing tools:

1. **Semantic Republishing tools must be able to detect and republish semantic metadata together with the data they annotate.** Semantic metadata must be republished together with the original data they annotate to allow users to benefit from additional third party services and tools which leverage semantic metadata of the data currently being processed. These additional services need semantically described structured data in order to be able to interpret the data and provide services upon them.
   For example browser tools, such as Firefox Operator[11], leverage semantic metadata found on the currently viewed web site and provide services (such as "Export contact to MS Outlook address book") upon the data which are annotated by the processed metadata.
2. **Semantic Republishing tools must be able to generate new semantic metadata exposing information about the provenance of the republished data.** If data are republished in a new context, new semantic

[11] `https://addons.mozilla.org/de/firefox/addon/4106`

metadata must be created which expose information about the provenance and the republishing process in a machine-interpretable way. Consequently, Semantic Web search engines can use this information to answer sophisticated data queries (such as *select all users who republished this section of this article* or *select all comments about a certain youtube*[12] *video related with the original video or with posts embedding the video*). Furthermore, it is important that Semantic Republishing tools expose detailed provenance metadata to boost the transparency and information accountability on the Web (see section 2.3). Finally, the exposure of detailed machine-interpretable provenance metadata allows implementing synchronization services which keep the republished data and the original ones in sync.

3. **Semantic Republishing tools must be able to interpret semantic metadata associated with the data to republish.** Existing semantic metadata can expose information about the content, the structure, the privacy settings and usage restrictions of the data they annotate. Hence, existing semantic metadata annotating the original data must be interpreted by Semantic Republishing tools in order to support users during the Republishing process (e.g., suggest tags of original data to reuse or suggest how to republish original data according to their licenses).
4. **Semantic Republishing tools must be easy to use for end-user.** To minimize usage barriers the interface of the Semantic Republishing tools must be similar to interfaces of already widely used traditional Republishing tools.

2.2 Metadata Modelling

We use the SIOC[13] ontology (namespace prefix `sioc`) together with the DCMI Metadata Terms[14] (namespace prefix `dcterms`), the Dublin Core Metadata Element Set[15] (namespace prefix `dc`), the FoaF[16] Ontology (namespace prefix `foaf`) and the RDF Site Summary 1.0 Module Content[17] (namespace prefix `content`) to describe republished data items in a machine-interpretable way. A republished data item is exposed as a resource of type `sioc:Post` and identified by a URI (e.g., `http://example.com#rebloggedItem_443af`) to enable any third party to make reference to this item in other RDF statements. The `sioc:content` property is used to expose the plain text content and the `content:encoded` property is used to expose the (X)HTML content of a republished item. The `dc:source` property relates the republished item with the resource from which it originates. The `dcterms:created` property exposes the date and time when the republished content has been published for the last time.

[12] `http://youtube.com`
[13] `http://rdfs.org/sioc/spec/`
[14] `http://dublincore.org/documents/dcmi-terms/`
[15] `http://dublincore.org/documents/dces/`
[16] `http://xmlns.com/foaf/spec/`
[17] `http://web.resource.org/rss/1.0/modules/content/`

2.3 Related Privacy and Usage Rights Issues

In the context of Data Republishing privacy and usage policies related with the data being republished must be taken into account. Privacy policies specify the confidentiality of data during transmission and also after receipt of data [10] and usage policies specify how and under which conditions clients are allowed to use data. With current widely-used Republishing tools users can either republish all data for which they have reading permissions without taking privacy and usage policies into account or cannot republish private or usage restricted data at all. Usage rights and privacy settings related with the selected data cannot be taken into account by these tools, because the settings are usually neither published in a machine-interpretable way nor are these tools able to interpret them. Consequently, traditional Data Republishing tools cannot support users in republishing data without compromising privacy and usage policies of data.

We believe that Semantic Data Republishing tools can help to overcome this problem and support privacy and right data usage by taking one of the following approaches:

(1) Interpreting privacy and usage policies related with the data being reused to guide users through the republishing process -i.e. support users in republishing data without compromising privacy policies or usage restrictions.
(2) Preserving privacy and usage policies of the original data when they are republished to enforce them for the republished data as well.

First, to allow Semantic Republishing tools interpreting policies of data web applications must expose not only their data in a machine-interpretable form, but also the related privacy and usage policies. If policy metadata are embedded in web pages to relate data with their policy descriptions, Semantic Republishing tools will be able to extract and interpret them. Consequently, users will be informed about policies related with the data they want to republish and will be warned if they are going to violate policies by republishing data. The Creative Commons Rights Expression Language (ccREL) [1], the standard recommended by Creative Commons (CC) for machine-readable expression of usage rights, is a successful example of publishing lightweight usage rights encoded in XHTML+RDFa. The proposed interpreting and guiding functionality of Semantic Republishing tools will however not prevent users from abusing data and compromising privacy and usage settings, but boost user's awareness of data privacy and 'good' data usage. This user's awareness combined with a transparent republishing process can ensure privacy through fair, appropriate and transparent use of information [20]. Semantic Republishing tools expose the provenance and republishing history of data in a machine-interpretable form. Consequently, users who violate usage and/or privacy policies related to data being republished can then be held accountable.

Second, to allow source and target application to share data and their policies Semantic Republishing tools must make the relation between the original data and the republished data explicit. Existing policy frameworks, such as REIN [11] or Protune [7], can be used on source and target applications to share the

policies of the original data and reason over them. Both frameworks are based on Semantic Web technologies and can be used for representing and processing distributed policies. However, in the context of Data Republishing the same data can be accessed on the source and target application. Therefore the source and target application must both be able to enforce the policies of the original data or the target application must redirect the client request to the source application which can consequently enforce the policies of original data for the republished data as well.

3 Implementation of a Semantic Reblog Tool

To demonstrate our ideas we have implemented a first example of a Semantic Data Republishing tool, namely a Semantic Reblog tool for the OpenSource Blogging Software WordPress[18]. The Semantic Reblog prototype consists of a client side bookmarklet and a server side reblog script. This section gives some insight into implementation issues and describes the Semantic Data Republishing process.

3.1 Extraction of semantic metadata

The Semantic Reblog tool extracts semantic metadata which annotate the current user selection (see step 1 and 2 in figure 1). On the client side the Semantic Reblog bookmarklet uses jQuery RDF plug-ins[19] to extract semantic metadata which are embedded in the selected (X)HTML region of the current web site. If no semantic metadata related with the selected (X)HTML region can be found, the Semantic Reblog server component parses the whole (X)HTML page searching for links to external RDF files which describe the page's data. The Semantic Reblog server component extracts triples from the external RDF files as well and checks if the selected data values belong to any object values of the extracted triples. The Semantic Reblog server component uses ARC2[20] to parse and extract semantic metadata. It must be noticed that the results of this server side extraction process can be ambiguous and that therefore the results of the client side extraction which takes positional information as well into account are usually more precise.

3.2 Generation of semantic metadata

The Semantic Reblog server component pastes the selected data into the tinyMCE editor[21] which is used as visual user editor by WordPress (see step 3 in figure

[18] `http://wordpress.org`
[19] `http://code.google.com/p/rdfquery`
[20] `http://arc.semsol.com`
[21] http://tinymce.moxiecode.com/

1). As described in section 2.2 the republished data are automatically annotated with semantic metadata exposing their provenance. All semantic metadata are embedded in the (X)HTML of the post's content and are serialized in XHTML+RDFa.

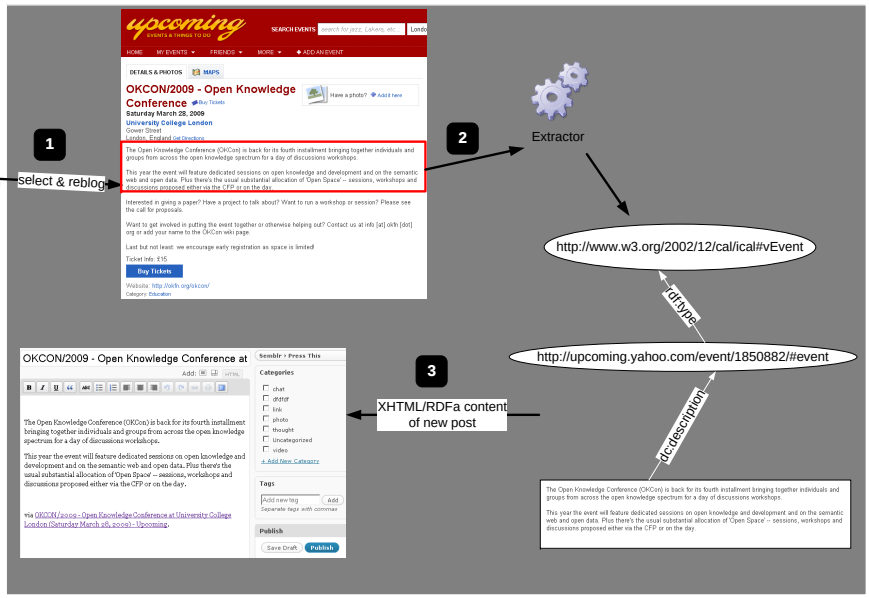

Fig. 1. Semantics-aware reblog process: extract and edit data and semantic metadata

3.3 Republishing data and semantic metadata

The Semantic Reblog tool preserves existing semantic metadata embedded in the selected content of the source site and republishes them together with the newly created semantic metadata and the data being annotated by them. Two different approaches have been identified for preserving semantic metadata embedded in (X)HTML snippets during the republishing process:

(1) **RDFa Serialization:** The RDF graph which has been extracted from the selected fragment of the source site can be serialized as XHTML+RDFa snippet. The disadvantage of this approach is that the parts of the selected (X)HTML content which are not semantically annotated get lost.

(2) **Snippet *Semantification*:** Cutting individual (X)HTML snippets from a semantically enriched (X)HTML pages can lead to (X)HTML snippets which contain meaningless, local and/or incomplete semantic metadata.

During the *semantification* process the semantic metadata embedded in the selected (X)HTML snippet are transformed into a valid semantically enriched (X)HTML snippet by reusing the semantic metadata extracted from the source site. The *semantified* (X)HTML snippets are serialized as XHTML+RDFa and are stored in a post's content.

Finally, the Semantic Reblog tool makes the republished and newly generated data and semantic metadata accessible for further web applications (see figure 2). The *semantified* (X)HTML snippet together with the newly generated semantic metadata are displayed in a user's reblog editor and can be edited by the user (see step 1 in figure 2). The editor can either be used in the visual edit mode in which the (X)HTML mark-up is hidden or in the HTML mode in which the data and their mark-up are displayed. The user can push the *publish* button to publish the post (see step 2 in figure 2). A newly created post is displayed on the user's blog. To make the embedded, reblogged resources accessible the WordPress SIOC Exporter[22] which models the content of a blog semantically and serializes it as RDF/XML document has been extended. The extended WordPress SIOC Exporter[23] is used to export resources embedded inside a post's content (e.g. reblogged data items) and relates them with the blog post via the `sioc:embeds` property of the SIOC ontology (see step 3 in figure 2) . Finally, the Semantic Web index service Sindice[24] is pinged to ensure that the republished semantically annotated data are indexed (see step 4 in figure 2).

3.4 Usage Scenario

To illustrate the benefits of our Semantic Reblog tool, an example scenario is described:

Tim is a typical Social Web users and one of his hobbies is taking pictures and sharing them on the Social Web application Flickr[25]. He likes discussing his pictures with other users interested in photography. Tim browses the Web and stumbles across one of his pictures which has been republished on the tumblelog of someone he does not know. The republished picture has been commented on the tumblelog and Tim is happy that he found such nice comments about his picture. Tim starts being interested in who else might have republished his picture. In particular, he would like to find all comments about his picture, no matter on which application they have been published. That means that Tim wants to find as well comments about postings which have republished his picture. Tim uses RepuSearch which is the fictive Semantic Web search engine specialized in querying the republishing-sphere. RepuSearch provides a simple search form allowing users to specify what they are searching for and formulates SPARQL queries in the background. Tim copies the URI of his picture into the main

[22] `http://sioc-project.org/wordpress/`
[23] `http://clauwa.info/download`
[24] `http://sindice.com/`
[25] `http://flickr.com`

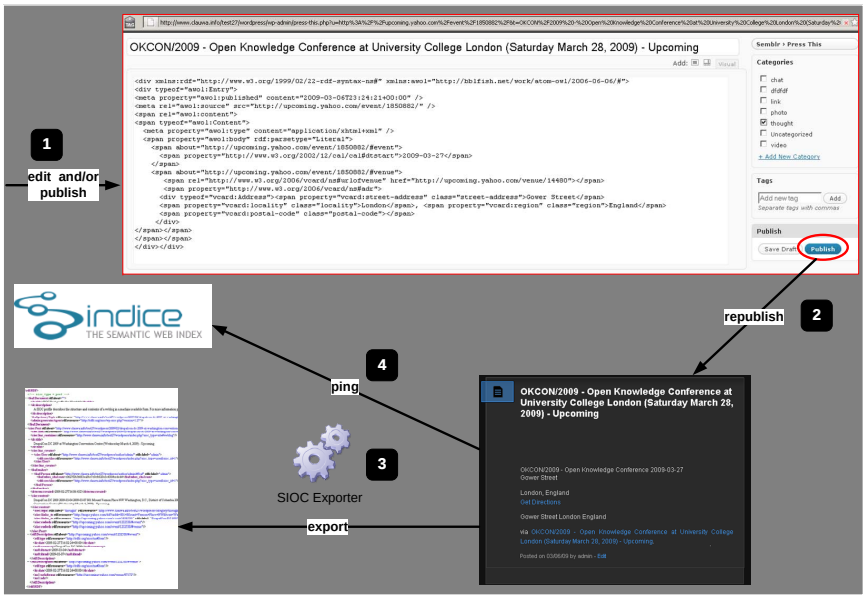

Fig. 2. Semantics-aware reblog process: republish and disseminate data and semantic metadata

search box, specifies that he wants to find comments about his picture which have been created in the last month and pushes the *search* button. RepuSearch displays as a result a list of comments created in the last month which refer to resources (e.g. posts) embedding Tim's picture.

Based on our work a scenario like this can be realized in the future Web where Semantic Web search engines exist allowing and supporting users in querying the Web like a huge database.

4 Related Work

There has been a significant amount of research in publishing and interlinking data on the Social Semantic Web. Social Semantic Web applications, such as semantic blogs [14] [12] [5] [18], semantic wikis [16] or semantic microblogs [15], allow average users to publish and interlink their data in a machine-interpretable way. Our work distinguishes from aforementioned work by focusing on republishing data. The requirements and challenges of publishing and modelling already published data slightly differ from publishing unpublished data and additional topics such as data privacy and usage rights arise in this context (see section 2.3).

SemiBlog [13] [3] illustrates how users can annotate their blog posts with existing metadata from desktop applications. SemiBlog and our prototype both focus on reusing existing semantic metadata. However, unlike semiBlog our Semantic Reblog prototype reuses semantic metadata of web resources. On the contrary semiBlog reuses metadata stored on desktop applications.

The Snippet Manager [6] and PiggyBank [8] are tools which allow users to collect, manage and share information found on the Web. Both tools are centralized services, which store the information snippets of a user in his or her personal semantic bank or knowledge base. Users can share information with other users by granting them access to parts of their knowledge base or semantic bank. On the contrary our Semantic Reblog prototype is not a centralized service, but generates semantically enhanced information snippets which are stored on distributed web applications. Furthermore, the Semantic Reblog prototype allows republishing information snippets or modified versions of them in a new context.

The work by Bojars et al. [4] shows how Semantic Web technologies can be used to ensure portability of user-specific data and content. In particular they propose to use FoaF and SIOC ontologies to model user information and user-generated content in a machine-interpretable way. Current application specific SIOC Importers and Exporters[26] demonstrate how data can be migrated from one Social Web application to another. After the portability process of a certain resource (e.g., a blog post) the target site holds a replica of the original resource. Our work distinguishes from their work by addressing another scenario in which a user does not want to generate and republish a replica of the original resource. On the contrary a user wants to generate a new resource which embeds and/or discusses the original resource or parts of it.

Semantic Clipboards aim to realize Data Portability across all kinds of applications. The Semantic Clipboard idea was first presented by [2] and describes how Semantic Web technologies can be used for moving structured content across application boundaries. The source and destination application negotiate the format of the data to be transferred and the clipboard itself either holds a copy of the RDF description of data or a reference pointing to the data's RDF graph. A first implementation of the Semantic Clipboard is presented in [17] and allows copying RDF metadata from any source application to any desktop applications. Other implementations of the Semantic Clipboard idea such as the RDFa Clipboard[27] and Semsol's Web Clipboard[28] exist as well. As the Semantic Clipboard idea aims to solve a very generic problem our work can be seen as an easy-to-use, lightweight and pragmatic solution for a specific problem in the context of the current republishing trend of the Social Web. Furthermore, Semantic Clipboards are made to act on an ideal Semantic Web where all published data are described in machine-interpretable way. Only semantically annotated data can

[26] http://rdfs.org/sioc/applications/

[27] http://www.w3.org/2006/07/SWD/RDFa/impl/js/rdfa-clipboard/

[28] http://bnode.org/blog/2006/06/12/web-clipboard-adding-liveliness-to-live-clipboard-with-erdf-json-and-sparql

be clipped and reused by using Semantic Clipboards. Our Semantic Reblog prototype however is designed to be used on the current Web where not all data are semantically annotated, but can be republished.

5 Conclusions and Future Work

In this paper we discussed the current republishing phenomenon on the Social Web, highlighted related privacy issues and illustrated the benefits of using Semantic Web technologies for Data Republishing. We introduced the concept of Semantic Data Republishing and described requirements and potentials of a new generation of semantics-aware Republishing tools. We presented a first implementation of such a tool, namely a Semantic Reblog tool, which allows users to republish data and their semantics, found on the Web, and annotates them with ontology-based metadata exposing their provenance. However, a number of issues still need to be addressed including how Semantic Republishing tools should handle the republishing of private and/or usage restricted data and how current Social Web applications can export their privacy policies in a machine-interpretable way to make them reusable for other applications. We plan to address these issues in future versions of our Semantic Reblog tool. Furthermore, we plan to improve the user interface of our Reblog tool to separate the semantic annotations from the editable (X)HTML code and hide them from the user.

References

1. Hal Abelson, Ben Adida, Mike Linksvayer, and Nathan Yergler. ccREL: The Creative Commons Rights Expression Language, March 2008.
2. Tim Berners-Lee. Semantic Clipboard. http://www.w3.org/DesignIssues/SemanticClipboard, January 2004. accessed: 12.2.2009.
3. Uldis Bojars, John G. Breslin, and Knud Möller. Using Semantics to Enhance the Blogging Experience. In *ESWC*, pages 679–696, 2006.
4. Uldis Bojars, Alexandre Passant, John G. Breslin, and Stefan Decker. Social Network and Data Portability using Semantic Web Technologies. In *2nd Workshop on Social Aspects of the Web (SAW 2008) at BIS2008*, pages 5–19, 2008.
5. Steve Cayzer. Semantic blogging and decentralized knowledge management. *Commun. ACM*, 47(12):47–52, 2004.
6. Steve Cayzer and Paolo Castagna. How to build a Snippet Manager. In Stefan Decker, Jack Park, Dennis Quan, and Leo Sauermann, editors, *Proc. of Semantic Desktop Workshop at the ISWC, Galway, Ireland, November 6*, volume 175, November 2005.
7. Juri Luca De Coi, Daniel Olmedilla, Piero A. Bonatti, and Luigi Sauro. Protune: A Framework for Semantic Web Policies. In Christian Bizer and Anupam Joshi, editors, *International Semantic Web Conference (Posters & Demos)*, volume 401 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.
8. D. Huynh, S. Mazzocchi, and D. Karger. Piggy Bank: Experience the Semantic Web inside your web browser. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(1):16–27, 2007.

9. Kingsley Idehen and Orri Erling. Linked Data Spaces & Data Portability. Linked Data on the Web workshop (LDOW2008), 2008.
10. Lalana Kagal, Tim Berners-Lee, Dan Connolly, and Daniel J. Weitzner. Using Semantic Web Technologies for Policy Management on the Web. In *AAAI*, 2006.
11. Lalana Kagal, Massimo Paolucci, Naveen Srinivasan, Grit Denker, Tim Finin, and Katia Sycara. Authorization and privacy for semantic web services. In *IEEE Intelligent Systems*, pages 50–56, 2004.
12. David R. Karger and Dennis Quan. What Would It Mean to Blog on the Semantic Web? *The Semantic Web ISWC 2004*, pages 214–228, 2004.
13. Knud Möller, John G. Breslin, and Stefan Decker. semiBlog - Semantic Publishing of Desktop Data. In *14th Conference on Information Systems Development (ISD2005), Karlstad, Sweden*, pages 855–866, Karlstad, Sweden, August 2005.
14. Ikki Ohmukai and Hideaki Takeda. Semblog: Personal Knowledge Publishing Suite. In *Proceedings of WWW 2004 Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics*, New York, USA, 2004.
15. Alexandre Passant, Tuukka Hastrup, Uldis Bojars, and John Breslin. Microblogging: A Semantic Web and Distributed Approach. 2008.
16. Alexandre Passant and Philippe Laublet. Towards an Interlinked Semantic Wiki Farm. In *SemWiki*, volume 360 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.
17. Gerald Reif, Gian Marco Laube, Knud Möller, and Harald Gall. SemClip - Overcoming the Semantic Gap Between Desktop Applications. In *Semantic Web Challenge*, volume 295 of *CEUR Workshop Proceedings*, 2007.
18. Aman Shakya, Hideaki Takeda, Vilas Wuwongse, and Ikki Ohmukai. SocioBiblog: A Decentralized Platform for Sharing Bibliographic Information. In Pedro Isaas, Miguel Baptista Nunes, and Joo Barroso, editors, *Proceedings of the IADIS International Conference WWW/Internet 2007*, volume 1, pages 371–380, Vila Real, Portugal, October 2007. International Association for Development of the Information Society, IADIS Press.
19. Alexander Stocker, Johannes Müller, and Klaus Tochtermann. Leichtgewichtiges Bloggen im Umfeld von Unternehmen: Microblogs und Tumblelogs. *e-commerce*, 2009.
20. Daniel J. Weitzner, Harold Abelson, Tim B. Lee, Joan Feigenbaum, James Hendler, and Gerald J. Sussman. Information accountability. *Commun. ACM*, 51(6):82–87, 2008.