Linked Data Authoring for Non-Experts

Markus Luczak-Rösch
Freie Universität Berlin
AG Corporate Semantic Web
Takustr. 9
D-14195 Berlin, Germany
luczak@inf.fu-berlin.de

Ralf Heese
Freie Universität Berlin
AG Corporate Semantic Web
Takustr. 9
D-14195 Berlin, Germany
heese@inf.fu-berlin.de

ABSTRACT

The vision of the Semantic Web community is to create a linked "Web of data" providing ubiquitous data access via machine understandable links between data resources. Due to the need of enormous expertise for producing and consuming linked data, the idea of linked data emerges only slowly and only a few data sources are currently available as linked data. In this paper we present Loomp to facilitate an increasing use of the Web data. Loomp enables non-experts to produce and publish semantically annotated content as easy as formatting text in word processors. Furthermore, Loomp simplifies the reuse of content with different Web applications.

Categories and Subject Descriptors

I.7.4 [**Electronic Publishing**]; H.4.1 [**Information Systems Applications**]: Office Automation—*Desktop publishing, Word processing*; H.3.2 [**Information Storage and Retrieval**]: Information Storage

General Terms

Semantic Web

Keywords

linked data, content authoring and publishing, Web of data

1. INTRODUCTION

Over the past years authoring content for the Web underran an interesting evolution which is well-known as the evolution from a Web 1.0 in the past towards the so-called Web 2.0 in the present. We state that the decisive catalyst for the evolution step from Web 1.0 to Web 2.0 was the broad success of blogging. Easy-to-use tools such as WordPress enabled a critical mass of people to change their behavior on the Web from consume-oriented to an interplay of consuming and publishing. In addition, people apply wiki systems as appropriate means for managing large knowledge bases collaboratively. Both developments show that ordinary Web users nowadays have a higher technical affinity and understanding of the network effects on the Web than in times of Web 1.0. Authoring content online has become a familiar task.

WWW2009, April 20-24, 2009, Madrid, Spain.

Likewise, we believe that the principles of linked data will change once more the perception of the Web and publishing content as linked data will become commonplace. At the moment, most linked data sources offer data that originally resides in relational databases. Wrappers convert this data into RDF format automatically. Besides the availability of database wrappers, we think that the broad success of linked data in everyday usage also depends on the availability of authoring tools that enable ordinary Web users – i.e. non-experts regarding to semantic web technologies – to publish linked data. Currently, there exists tool support for editing metadata or adding semantics to wikis, but to our best knowledge tools are missing that allow *non-expert users* to enrich content such as text and multimedia objects with detailed semantics.

In this paper we introduce Loomp as a system that allows every Web user to create semantically enriched content as easy as to format text in a word processor. A user can easily publish the same content to various applications such as Web browsers, blogs, and wikis. Additionally, Loomp features an integrated linked data server for publishing the content as linked data (Figure 1).

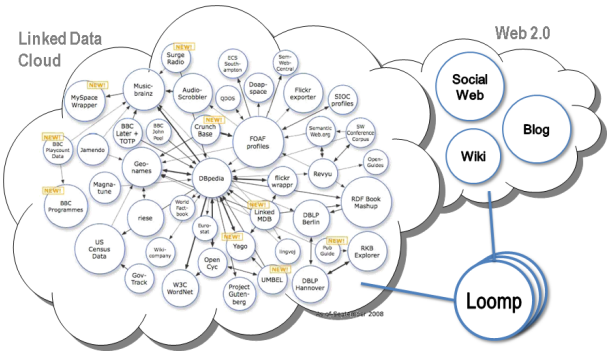

Figure 1: Providing manually created linked data to the linked data cloud and Web applications

The paper is structured as follows: As an illustrating example of using Loomp in a real world scenario, we report our results of interviewing journalists and editors in Section 2. In Section 3 we give an overview of the Loomp architecture and describe key points of Loomp in detail. In Section 4 we present related work and in Section 5 we conclude and outline our future work intentions.

2. THE JOURNALISTS USE CASE

Nowadays we can find interesting use cases with underlying business models in literature that are based on linked data contained in webpages. The BBC for example developed a system that utilizes automatic enrichment of content to increase the visiting time on their website[1]. By providing links to information sources of their own website that are related to the currently shown website, a user does not need to search external sources for further information. In general, it is more complex to identify the business value of publishing linked data on the Web – especially the advantage over search engines does not hold as an argument. For example, we have to distinguish between the automatic generation of linked data based on conventional data sources and the manual enrichment of Web contents. In the first case, the creation and publishing of linked data does not cause any additional effort for the author. In the latter case, an author has to put more effort into the creation of Web content, because he has to face the additional task of annotation.

In this section we describe a use case in the domain of journalism which illustrates the added value of manual creation of linked data. This *journalists use case* is representative for the domain of content and knowledge intensive work in a heterogeneous environment. We personally interviewed journalists and editors of publishing houses which are typically working in the areas of print publishing, online publishing, and cross-media publishing. The self-conception of both groups is a bit overlapping, because sometimes freelancers writing articles call themselves editors. However, in this paper we distinguish these two groups on the basis of their tasks: journalists research and write articles and editors revise and publish the work of journalists- Journalists may be employed or work as freelancers. As a matter of fact, however, many freelance journalists have a contract with only a single publishing house.

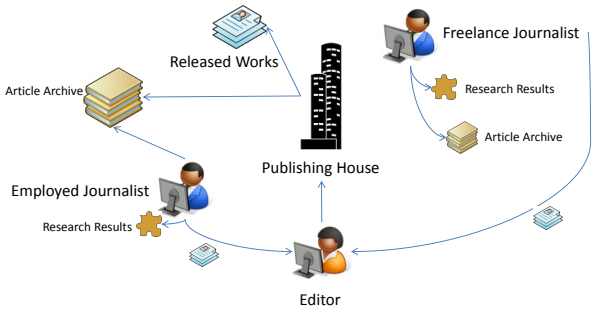

Figure 2: Representative setting of today's publishing houses

Figure 2 depicts a representative setting of journalists and editors in the context of today's publishing houses. Journalists research specific topics on demand and access various information sources for this purpose, e.g. websites, books, related articles, and human informants. Our personal interviews yielded that journalists note the results of this research using paper and pencil. Only very few journalists use digital devices for this task and even fewer apply information management systems. To transfer the finished article to the responsible editor at the publishing house the people use free text documents and email communication. In some cases, especially in the online publishing sector, journalist enter their articles directly into editorial managements systems or content management systems. Furthermore, they add appropriate categories and tags. Finally, an editor revises and releases the articles for his department.

During the last years several studies such as [9] recognized an increasing importance of online publishing and claimed for more professional journalists in online media. Classical publishing houses and media companies are the most important information provider on the Web. However, even these providers need to satisfy the demands of the consumer for cross-media publishing of contents.

In Figure 3 we give an overview of Loomp in the journalist use case setting. The application of Loomp is possible in two ways. First, as a personal information management system for the journalists to facilitate an easier search and reuse of former research results and former articles. Second, as an editorial management system for the publishing houses which acts as flexible means for cross-media publishing and personalized content aggregation.

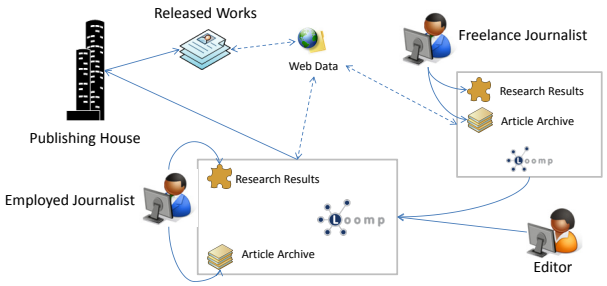

Figure 3: Setting of Loomp in the context of the journalists use case

Loomp helps journalists to manage their notes, interview logs, references, addresses, etc. The system supports users to enrich them semantically, e.g., by an automatic annotation assistant and an easy-to-use editor for manual annotations. Furthermore, authors can write and manage their articles with Loomp – the creation of semantic annotations is possible. Especially, Loomp helps to link an article to its information sources. Loomp provides human and machine readable representations of the content, so that it can easily be searched, reused, and published. In this case Loomp serves as a Web authoring tool.

On the side of the publishing houses editors use Loomp to revise and edit articles written by journalists. Furthermore, they can add more annotations to articles and possibly interlink them. Finally, they choose a publishing channel for the work (e.g. such as a blog, an RSS feed, a wiki, or print) and release them. At this point we regard Loomp as a distributed and collaborative Web content management system which facilitates cross-media publishing of semantically enriched information.

The benefits of Loomp in this context are manifold. Most important, Loomp features a semantic search engine and thus decreases effort of finding information that a user has created before. Because the system also keeps track of provenance information, authors can retrace the sources of an article. Since Loomp serves all content semantically enriched, consumers can modify the presentation of it according to

[1] `http://www.bbc.co.uk/blogs/radiolabs/`

their current information needs. For example, the reader may decide to highlight all names of persons in a text. As a consequence authors and editors are more or less freed from the effort of formatting texts in bold, italic, or underlined. Based on the semantic annotations the content provider can offer content and target group specific services, e.g., the BBC example of providing related information or accurately fitting commercials.

3. DESIGNING A LINKED DATA EDITOR FOR THE MASSES

Considering our use case example it comes clear that the task of building a linked data authoring tool for a broad range of Web users is a complex task. The design criteria have to respect that the target group has no theoretical understanding of what RDF and linked data is. Moreover, we expect that the common understanding of the Web as a network of human readable pages will not change in the near future, so that the value of a Web of data is not very familiar for ordinary users. To lower the barriers the compelling simplicity of Web 2.0 applications should be transfered to the task of creating linked data: light-weight, easy-to-use, and easy-to-understand. In the following list we describe design requirements on an authoring system which in our opinion are necessary to enable non-expert users to participate in the Web of data.

Intuitive user interface The system hides the complexity of creating linked data by providing an intuitive user interface. It follows common mindset and uses well-known procedures of system interaction to produce semantic annotations. For example, every computer user is nowadays able to select a text and to click on a button to format it italic.

Simple vocabularies Although Web users know the term URL or Internet address, they are currently rarely aware of namespaces. Thus, the system provides access to vocabularies without going into technical details. Each concept of a vocabulary has a meaningful label and is explained in simple terms and with examples of usage. The system supports widely accepted vocabularies and is able to map concepts of equal or similar meaning.

Reuse of content Often the same content is published in different formats, so the system has to be able to convert the content to common formats such as PDF and to interact with other (Web) applications such as blogs and wikis.

Support for linked data The system offers its content as linked data. In order to create linked data, the system has to provide support for searching resources and linking to them.

Data authority A user decides which data is publicly available.

Easy to install The requirements for installation and running the system are low, so that it can be installed in most webspaces. The need for configuration is reduced to a minimum.

Some of these requirements seem to be rather visionary, but we realize Loomp with these requirements in our mind. We strongly focus technical requirements as well as socio-economic requirements. Both, but mostly the latter, stress the goal to design an authoring tool for the Web of data which does not contradict human mindsets.

3.1 The Loomp System Architecture

The two basic types of resources that are managed by Loomp are fragments and mash-ups. A fragment is the smallest piece information in Loomp and describes a closed notional entity containing annotated text, multimedia content, or a SPARQL query. Mash-ups are composed of an arbitrary number of fragments. Both, fragments and mash-ups, are assigned a unique identifier (URI) and can be retrieved by dereferencing this URI. In the case of SPARQL queries we assign two identifiers, one for the query itself and one for the result of the query. As known from the RDF specification, the identifier can be used to make statements about them, e.g., add metadata such as the author and the creation date.

We designed Loomp as a typical LAMP[2] compatible Web application (see Figure 4). Loomp serves contents either in RDF (e.g. for linked data clients) or in XHTML/RDFa [1] (e.g for Web browsers).

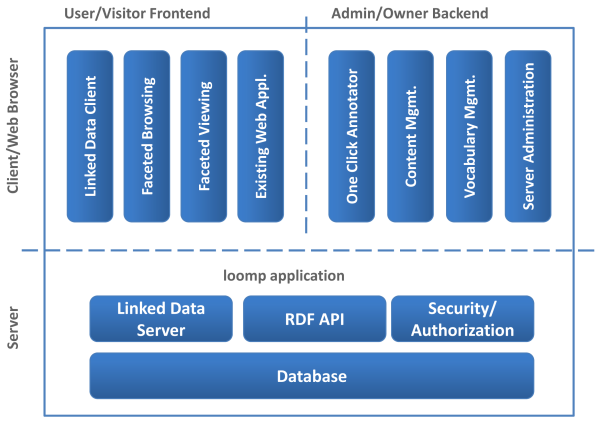

Figure 4: Overview of the Loomp system architecture

On the server side the main components are a database for storing the data, a linked data server for providing access to the data as linked data, an RDF API for accessing the data by the Loomp application, and a security/authorization component for granting access to the data. The linked data server and RDF API components are realized with the RAP Pubby library [14].

On the client side we distinguish between a frontend and a backend. The term frontend comprises all clients that retrieve data from the Loomp application without authorization, e.g., access to all publicly available content by linked data clients and by Web browsers. The boxes *Faceted Browsing* and *Faceted Viewing* represent websites that exploit the semantic annotations of the content for navigating to related content and for changing the appearance of the content (see Section 3.3). Loomp also features a plug-in mechnism to allow (read and write) access to its content from *Existing*

[2]LAMP: Linux, Apache, MySQL, PHP

Web Applications. Thus, for example, it is possible to view the annotated content as simple HTML pages, blog entries, or wiki pages. The term backend comprises clients that have to authorize before they can access data – typically, these clients are allowed to modify content, e.g., by the One Click Annotator and the content management component (see Section 3.2). Using the *Vocabulary Mgmt.* component an experienced user may add vocabularies and modify them.

3.2 One Click Annotation and Content Management

To enable users to annotate fragments as easy as formatting text in word processors, we develop the *One Click Annotator* (see Figure 5). The One Click Annotator extends the TinyMCE[3] online HTML editor to support RDFa annotations in a WYSIWYG way. It adopts the look & feel style which is well-known from word processors for applying style sheets to text. On the left side of the annotation toolbar a user selects the concept for annotating a piece of text, on the right side she chooses a vocabulary from a drop-down menu. The effort for annotating text semantically substitutes the effort of formatting text bold or italic. For example, the user selects an email address in the text and clicks the button *Email*. In a next step we plan to integrate an automatic annotation recommender. Nevertheless the user has the authority to reject suggested annotations. In the background the One Click Annotator inserts RDFa annotations into the XHTML. If a user saves her changes the XHTML/RDFa content is send to the server which in turn extracts RDF statements and stores them into an triple store. In detail, a fragment is stored as an XHTML/RDFa representation as well as a batch of extracted and assigned RDF metadata.

Figure 5: Using the One Click Annotator for annotating text semantically

A mash-up consists of a sequence of fragments. The user interface for modifying mash-ups exploits modern web technologies to allow drag-and-drop and in-place editing of its fragments. A user can extend a mash-up by creating a new fragment at the desired place of the mash-up or searching and dragging an existing fragment to it.

Loomp makes use of the semantic annotations for searching fragments and mash-ups. For example, if a search term has been annotated with different concepts, then the result items are grouped and displayed according to these concepts. In a second step, a user can refine the search by retrieving the remaining fragments of a group. A fragment can be contained in many different mash-ups, it is even possible that a mash-up contains the same fragment more than once. To comply with the linked data principles a user can link fragments and mash-ups to other resources on the Web, e.g., Loomp references resources in DBPedia to support unique identifiers.

[3] `http://tinymce.moxiecode.com/`

3.3 Consumer-oriented Presentation

Nowadays, if an author decides to emphasize a text phrase that seems to be important to her, she may format it using an italic font or select a different font color. A consumer of the content is unable to change the appearance in order to facilitate the accomplishment of a specific task, e.g., a consumer would like to highlight phrases (i.e. all names of persons belonging to a working group) that are important to decide on the relevance of a webpage for conducting a search on a topic.

Loomp aims at supporting consumer-oriented presentation of content. Typically, the content managed by Loomp is delivered in XHTML/RDFa format and, thus, it contains semantic annotations. Using Loomp the appearance of the content is defined by cascading style sheets (CSS). By separating content from appearance an author who uses Loomp has still the possibility to exert influence on the appearance of the content by changing an existing style sheet or providing a user-specific one.

In contrast to current Web pages Loomp also allows consumers to change the appearance of the content according to her current needs. By means of a toolbar a consumer can format semantically annotated phrases, e.g., she can highlight the names of members of a specific working group with a yellow background. We call this feature *faceted viewing*.

4. RELATED WORK

In [11] Heath et al. divided the creation of linked data into the following steps: i) select vocabularies, ii) partition the RDF graph into "data pages", iii) assign an URI to each data page, iv) create HTML variants of each data page, v) assign an URI to each entity, vi) add page metadata and more links, and vii) add a semantic sitemap. With Loomp we follow these steps for creating linked data. Using the One Click Annotator a user selects from a set of vocabularies that reuses existing ontologies (i). Considering Loomp we distinguish between fragments and mash-ups which are automatically assigned URI in the background. The content is published in HTML format beside other (ii–v). The user may also add meta data to fragments and mash-ups (vi). Last but not least the Loomp server generates a sitemap of all publicly available content.

Tools for creating semantically enriched content include semantic wiki and semantic tagging engines. Examples for semantic wikis are OntoWiki [4], Ikewiki [15], or Semantic MediaWiki [12]. These wikis extend traditional wikis by functionalities that enable users to add annotations to a wiki page and to specify relationships between pages based on ontologies. In our opinion semantic wikis are far from being usable by non-experts. Besides the effort to learn a special syntax to write and to annotate content, a user has to cope with technical terms such as resource, different kinds of relationships, and namespaces. In contrast semantic tagging engines such as faviki[4] exploit the well-known user interaction procedure of tagging to annotate content. In the background faviki calls functions of the Zemanta Semantic API[5] to retrieve suggestions for tags, e.g., Wikipedia terms.

[4] `http://www.faviki.com`
[5] `http://www.zemanta.com/`

Zemanta as well as OpenCalais[6] are examples for services that automatically annotate content. In Loomp we use these services to suggest annotations to users.

In [8], the author present a JavaScript API for modifying RDFa directly on the client side and synchronizing the changes with the server. While our OneClickAnnotator is suitable for extensive changes of annotations of a text, this JavaScript library is a useful supplement for smaller changes of annotated texts.

The Tabulator linked data browser [5] allows users to edit data directly on the Web of data. However, since it requires a Firefox plug-in in its current stage of development we see it as a proprietary tool. In the context of OpenLink Data Spaces[7] provides a complete platform for creating a presence on the Web of data, e.g., calendar, weblog, or bookmark manager. However, they focus on describing the data entities semantically while we enrich the content itself.

In addition, many wrappers have been developed for websites and relational databases to generate linked data. For example, in [6] Bizer et al. describe one of many examples for writing an API wrappers to mash-up and interlink data as RDF. Auer et al. present an approach to create linked data based on crawling and processing data from Web pages in [3]. As a more direct possibility to publish linked data several tools support the mapping of relational databases to RDF, e.g., D2RQ [7], RDB2RDF [13], and Triplify[2]. All these approaches have in common that they only support an indirect way of creating linked data, e.g., an author cannot directly annotate the content.

On his website [10] the author presents the idea of writing RDF data directly to non-RDF data sources. With loomp we pursue a similar goal but from our viewpoint the data should recide on the user's server and not on the application server, if the user wishes so. Using the loomp plug-in an external application can directly retrieve the user data from his server.

5. CONCLUSION AND OUTLOOK

In this paper we presented a Web application for creating, managing, and publishing semantic data, namely Loomp. With Loomp we make an important contribution to the success of linked data. In contrast to existing editors, our main focus lies on an intuitive user interface, that enables every Web user to produce semantically enriched content and to distribute it across various media easily. Furthermore, we reduced the system requirements to operate a Loomp server to a minimum (LAMP system) and integrated a linked data server which provides all public content as linked data to increase the awareness and usage of linked data .

An initial version of Loomp has recently been released which illustrates the basic functionalities, e.g., content management and publishing. As a major feature, we will exploit existing web services to propose annotations automatically which can be accepted or rejected by a user. In our future work, we will also address the integration and the support of third party applications such as blogs, wikis, and word processors.

[6] `http://www.opencalais.com/`

[7] `http://virtuoso.openlinksw.com/wiki/main/Main/Ods`

Acknowledgments

This work has been partially supported by the "InnoProfile-Corporate Semantic Web" project funded by the German Federal Ministry of Education and Research (BMBF).

6. REFERENCES

[1] B. Adida and M. Birbeck. RDFa primer – bridging the human and data webs. W3C Working Group Note, Oct. 2008.

[2] S. Auer. Triplify. Project Website, Retrieved January 9, 2009, from http://triplify.org.

[3] S. Auer, C. Bizer, J. Lehmann, G. Kobilarov, R. Cyganiak, and Z. Ives. DBpedia: A nucleus for a web of open data. In *Proceedings of ISWC/ASWC 2007*, volume 4825 of *LNCS*, pages 715–728. Springer Verlag, Nov. 2007.

[4] S. Auer, S. Dietzold, and T. Riechert. OntoWiki - A Tool for Social, Semantic Collaboration. In I. F. Cruz et al., editors, *The Semantic Web - ISWC 2006*, volume 4273 of *LNCS*, pages 736–749. Springer, 2006.

[5] T. Berners-Lee et al. Tabulator redux: Writing into the semantic web. Technical report, ECS, University of Southampton, 2007.

[6] C. Bizer, R. Cyganiak, and T. Gauss. The RDF Book Mashup: From Web APIs to a Web of Data. In S. Auer, C. Bizer, T. Heath, and G. A. Grimnes, editors, *SFSW*, volume 248 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.

[7] C. Bizer and A. Seaborne. D2rq - treating non-rdf databases as virtual rdf graphs. In *ISWC2004 (posters)*, November 2004.

[8] S. Dietzold, S. Hellmann, and M. Peklo. Using javascript rdfa widgets for model/view separation inside read/write websites. In *Proceedings of the 4th Workshop on Scripting for the Semantic Web*, 2008.

[9] P. Glotz and R. Meyer-Lucht. Zeitung und Zeitschrift in der digitalen Ökonomie – Delphi-Studie. Project Website, Retrieved January 10, 2009, from http://www.unisg.ch/org/mcm/web.nsf/wwwPubInhalteGer/Online Publishing Delphi-Studie.

[10] M. Hausenblas. pushback - Write Data Back From RDF to Non-RDF Sources. `http://esw.w3.org/topic/PushBackDataToLegacySources`, March 2009. retrieved on 3rd March, 2009.

[11] T. Heath, M. Hausenblas, C. Bizer, R. Cyganiak, and O. Hartig. How to publish linked data on the web, October 2008. Tutorial at the ISWC2008, retrieved on 10th February, 2009.

[12] M. Krötzsch, D. Vrandecić, and M. Völkel. *The Semantic Web - ISWC 2006*, chapter Semantic MediaWiki, pages 935–942. Lecture Notes in Computer Science. Springer Verlag, 2006.

[13] A. Malhotra. Progress report from the rdb2rdf xg. In C. Bizer and A. Joshi, editors, *International Semantic Web Conference (Posters & Demos)*, volume 401 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.

[14] R. Oldakowski et al. RAP: RDF API for PHP. In *Proceedings of SFSW 2005*, May 2005.

[15] S. Schaffert. Ikewiki: A semantic wiki for collaborative knowledge management. In *Proceedings of STICA'06*, Manchester, UK, June 2006.