Proceedings of the 7th International Conference on Applied Informatics
Eger, Hungary, January 28–31, 2007. Vol. 2. pp. 237–245.

Browsing the Semantic Web

Peter Jeszenszky

Faculty of Informatics, University of Debrecen
e-mail: jeszy@inf.unideb.hu

Abstract

The Semantic Web is a vision that aims at the machine processibility of web content. Many researchers believe that it will be the next logical step in the evolution of the web. Although the Semantic Web is a vision and the current web is far from "being semantic", the underlying standards and technologies, that are still experimental in many cases, have enormous potential.

This paper presents a solution to add a unique feature to the popular Firefox web browser, the capability to extract XMP metadata from web resources, making a small step toward a "more semantical" web. The solution is based on the Piggy Bank Firefox extension that turns Firefox into a "Semantic Web browser".

Keywords: Semantic Web, browser, Firefox, XMP, metadata

1. Introduction

The Semantic Web is a vision that aims at the machine processibility of web content. Many researchers believe that it will be the next logical step in the evolution of the web. If the vision comes true, that will enable us to implement more intelligent information services than at present.

Computer processing of web content is an extremely difficult task because most of the currently available web content is intended for human consumption. Although CSS makes it possible to separate content and presentation, and is a very popular and widely used technique now, the task of locating relevant information in an arbitrary web page is still hopelessly difficult, and should require a true AI.

Although the Semantic Web is a vision and the current web is far from "being semantic", the underlying standards and technologies, that are still experimental in many cases, have enormous potential.

One of its cornerstones is RDF [16], a flexible and simple framework to represent knowledge on the web. In order to enable machine processibility of web content data must be available in RDF. That assumes the existence of machine readable web pages written in RDF. The extensive availability of RDF metadata is the main problem. Although certain Semantic Web applications, for example FOAF [6] are based on the use of such machine readable web pages, it is not likely that millions of users will publish RDF data on the web.

XHTML 2.0 [20] provides an elegant and easy-to-use annotation mechanism, called metadata attributes module to embed RDF in XHTML in such a way that does not put additional burden on web page authors. However, XHML 2.0 is a work in progress, and it is unknown when it will be become a stable standard and supported widely. Furthermore XHTML 2.0 is not backward-compatible with previous XHTML versions.

The availability of RDF data together with web ontologies will enable future Semantic Web applications, including web browsers to use reasoning, exploiting the web as an immense knowledge base. There are a few applications for the time being that may be called Semantic Web browsers since they utilize Semantic Web technologies and employ innovative techniques and solutions providing a novel browsing experience.

Although there are standalone GUI applications like IsaViz [8] and RDF-Gravity [15] that may be used to view already available RDF data they can not be considered as general purpose web browsers. These tools may provide search capabilities (IsaViz also serves as an RDF editor) but they are primarily intended to visualize the graph structure of data.

Even a conventional web browser may function as an RDF browser if RDF data is converted to HTML on the server side. Longwell [11] and Tabulator [19] follows this approach to add Semantic Web capabilities to existing web browsers. Longwell is a web application that provides a very sophisticated user interface to browse and search RDF data and is based on the faceted browsing UI paradigm. Unfortunately, RDF data must be available locally at the computer running the web application, that limits the possible uses. Tabulator is more flexible and accepts URLs pointing to RDF data. The implementation consists of a set of JavaScript files that utilize AJAX techniques. Complex searches can be performed via SPARQL [18] queries, however, this feature is intended to be used by Semantic Web experts. Both Longwell and Tabulator provide views to explore data from different aspects. For example, they can display geographical information on a map using Google Maps.

Other tools provide Semantic Web capabilities and enrich browsing experience being integrated into existing web browsers. The extensibility of the Mozilla [12] platform allows extra functionality to be added to Mozilla-based web browsers easily. That makes Mozilla an ideal platform to experiment with Semantic Web technologies. Not surprisingly, many of the currently available semantic browser tools are available as Mozilla Firefox extensions.

An example is Piggy Bank [13] that may be the most powerful general purpose semantic browser tool that is currently available. It is a unique Firefox [5] extension that uses screen scrapers to obtain RDF data from web pages. (This technique together with discussed later in Section 5.)

XMP [1] is an RDF-based framework that allows metadata embedding in application files providing a rich source of RDF metadata for semantic web applications. Although these tools they can not be considered as.

This paper presents a solution based on Piggy Bank that offers a brand new web browser feature, ie. the capability to extract embedded XMP metadata from the resources that are accessible from the current web page.

2. XMP

2.1. What is XMP?

XMP is an RDF-based metadata framework of Adobe Systems Incorporated, that provides the following: *a)* a data model, *b)* a storage model, *c)* metadata schemas *d)* and rules that describe how to embed XMP metadata in various application files.

The data model makes it possible to associate properties with resources in order to describe them. A resource may be a file, or a portion of it, that may be meaningful to a processing application in itself, and that is also a distinct logical component of the file structure.

The storage model provides an implementation of the data model, it uses a subset of the RDF/XML [14] syntax to represent XMP metadata. Metadata describing a particular resource are serialized as RDF/XML and may be embedded in the resource itself in an XMP packet. The packet is a wrapper around the RDF/XML data that is surrounded by easily recognizable delimiters. The XMP specification [3] provides for the way of embedding of these packets in many common file formats, such as GIF, JPEG, PDF, PNG, PostScript, TIFF and others.

XMP schemas are sets of predefined metadata elements that can be used by various applications to characterize a wide range of resources, such as electronic documents, digital images, audio and video files.

2.2. Why is XMP important?

XMP has obvious advantages:

- It provides a standard and file format independent way to annotate digital images and other resources.

- Embedded XMP metadata will be available to virtually every application. Even an application without the knowledge of the file format may scan a file for embedded XMP packets. (However, it is not recommended, as discussed later.)

- Metadata is shipped together with the embedding resource and will not lost.

It is not an exaggeration to say that XMP opens up exciting new possibilities for digital photography and image editing applications. If it will be widely used on the web, a rich source of metadata will be available to Semantic Web applications that may be utilized effectively.

3. Current XMP support

XMP is a flagship product of Adobe Systems Incorporated. Their software products, for example Adobe Acrobat, Adobe FrameMaker, Adobe GoLive, Adobe InDesign and Adobe Photoshop support XMP by default.

Unfortunately, popular open source image and photo editing applications (for example GIMP) do not support XMP for the time being. Similarly, none of the currently available open source office applications (e.g. OpenOffice.org) can embed XMP metadata in the documents they produce.

Adobe XMP Toolkit (XMP SDK) [2] is Adobe's open source XMP library that allows developers to add XMP functionality to their application programs. It is intended to be cross-platform and runs on multiple operating systems, namely on UNIX/Linux, Windows and Macintosh.

4. Piggy Bank

4.1. What is Piggy Bank?

Piggy Bank is a Firefox extension that turns the popular Firefox browser into a "Semantic Web browser". It means that:

- It employs innovative techniques and solutions providing a novel browsing experience. Piggy Bank provides a taste of the future, and forecasts that we may expect from browsers if the Semantic Web becomes a reality.
- It utilizes Semantic Web technologies.

Piggy Bank is developed as an open source software in the SIMILE [17] project, that is a collaboration of W3C, MIT Libraries and MIT Computer Science and Artificial Intelligence Laboratory (MIT CSAIL).

4.2. What is it good for?

Piggy Bank can extract "pure" information from a web page. Then the following options are available:

- Collected information can be saved and stored locally.
- Collected information can be uploaded onto communal information repositories called semantic banks. Semantic banks enable information to be shared with other people.
- Relevant information can be filtered out of the collected information flexibly.

- Novel display methods are available to visualize the collected information. For example, it is possible to display collected information items on a geographic map or on a timeline. The graph structure of collected information can be examined as well.

4.3. How does it work?

Piggy Bank can extract RDF from a web page in the following two cases:

- If the HEAD section of the HTML document contains LINK elements to RDF data, that is either in RDF/XML or in N3 format.
- The URL of the web page matches the URL pattern of an active screen scraper.

A screen scraper is a special JavaScript code that can transform the content of particular web pages to RDF. A screen scraper has an URL pattern. Web pages having an URL that matches that pattern can be processed by the scraper. Typically, the HTML structure of the web pages offered by a given web site is hardwired into a screen scraper that will enable it to extract all relevant information from a page. A screen scraper not only can process the current web page, but it may load and process related web pages also or invoke a web service to obtain auxiliary information to enrich the content. Screen scrapers must be installed in Piggy Bank in order to operate. A few screen scrapers are available at the Piggy Bank web site to collect pure information from popular web sites, such as Flickr or the ACM Portal. Another option is to write an own scraper.

Piggy bank uses Longwell [11] to display the collected information.

5. XMP in Firefox

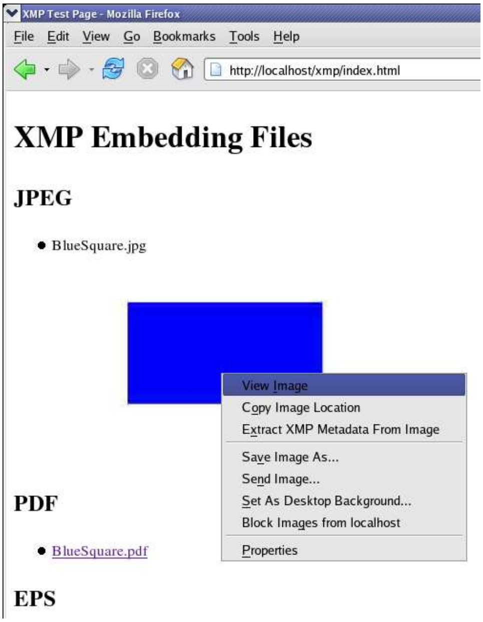
5.1. The main idea

Since the author is an enthusiastic fan of both Firefox and metadata standards including XMP, it was a fairly obvious, and at the same time a very desirable goal for him to add XMP support to his favourite web browser.

A Firefox extension is presented here that provides the following functionality:

- All the embedded XMP metadata can be extracted from the images and the resources that are linked to current web page with one click for browsing.
- XMP metadata can be extracted for browsing from individual images and links also.

To the best of our knowledge none of the currently available web browsers offers such a function, thus it is a completely new browser feature. It is probable that such a functionality should play an important role in the popularization of XMP also.


XMP Test Page - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://localhost/xmp/index.html
View Image
Copy Image Location
Extract XMP Metadata From Image
Save Image As...
Send Image...
Set As Desktop Background...
Block Images from localhost
Properties

XMP Embedding Files

JPEG

- BlueSquare.jpg

PDF

- BlueSquare.pdf

EPS

Figure 1: Right clicking on an image in Firefox. Select "Extract XMP Metadata From Image" to view embedded XMP metadata

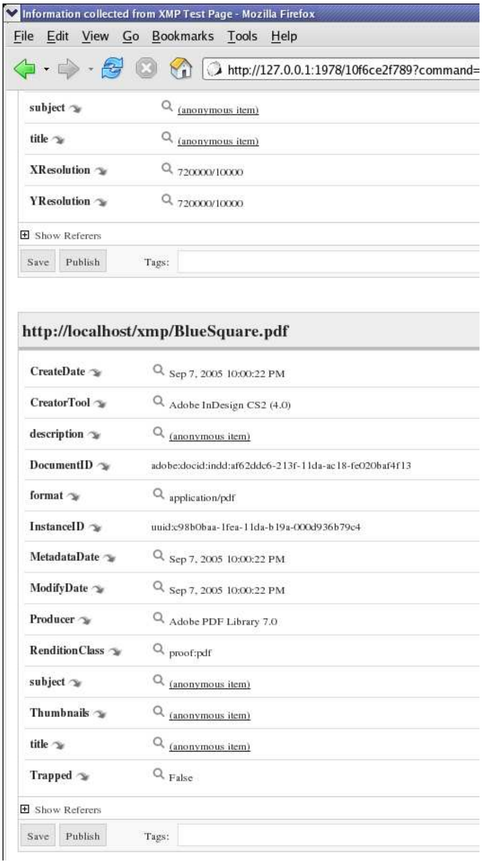
5.2. Implementation

The idea fits in very well with the philosophy of Piggy Bank. Furthermore, Piggy Bank can serve as a basis for the implementation. Actually, Piggy Bank has been extended to provide the above functionality.

Only few additional user interface elements were necessary to be added to the extension. Notably an XMP menu (containing the "Extract XMP Metadata From This Page" and "XMP Options" items) to the "Tools" menu, an XMP options dialog and two appropriate items ("Extract XMP Metadata From Image", "Extract XMP Metadata From Link") to the context popup menu that appears when the user right clicks on an image or a link.

Currently, XMP extraction functionality is available to Piggy Bank as a RESTful web service. The web service accepts an URL in a HTTP [7] GET request and extracts XMP metadata that is returned as an RDF/XML document. It is a convenient solution to completely separate XMP logic from Piggy Bank code. Since screen scrapers may also utilize web services, this approach fits well with Piggy Bank, too. A great advantage of the solution is that it does not require XMP software to be installed on the client side, implementation details are hidden behind the web service.

The XMP extractor web service operates as follows:


Information collected from XMP Test Page - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://127.0.0.1:1978/10f6ce2f789?command=

subject	(anonymous item)
title	(anonymous item)
XResolution	720000/10000
YResolution	720000/10000

⊞ Show Referers
Save
Publish
Tags:

http://localhost/xmp/BlueSquare.pdf

CreateDate	Sep 7, 2005 10:00:22 PM
CreatorTool	Adobe InDesign CS2 (4.0)
description	(anonymous item)
DocumentID	adobe:docid:indd:af62ddc6-213f-11da-ac18-fe020baf4f13
format	application/pdf
InstanceID	uuid:c98b0baa-1fea-11da-b19a-000d936b79c4
MetadataDate	Sep 7, 2005 10:00:22 PM
ModifyDate	Sep 7, 2005 10:00:22 PM
Producer	Adobe PDF Library 7.0
RenditionClass	proof:pdf
subject	(anonymous item)
Thumbnails	(anonymous item)
title	(anonymous item)
Trapped	False

Figure 2: Browsing the extracted XMP metadata in Firefox

1. The web service gets an URL in a HTTP GET request. XMP metadata will be extracted from the resource identified by the URL.

2. The web service initiates a HTTP HEAD request using the URL to query the MIME type and the length of the resource.

    - If the resource is not found or the XMP extractor web service does not support its format determined by the MIME type, then an appropriate response is returned indicating the error.

    - It is also an error if the content length of the resource exceeds a configurable limit.

3. The web service begins to retrieve the content of the resource and scans it for embedded XMP metadata.

- If the XMP packet is found, it is postprocessed then returned as an RDF/XML document.
- If XMP metadata is not found, an appropriate response is returned.

The web service is implemented in Java using the JAX-WS API [9, 10] and deployed to Apache Tomcat [4]. Java was a natural choice, as it is the "native language" of the author. Moreover, it is not tied to any particular vendor's operating system or platform.

6. Conclusions

The paper introduces a brand new feature that works with the Firefox web browser, the ability to display XMP metadata embedded in resources that are accessible from the current web page.

The implementation is based on the Piggy Bank Firefox browser extension, that turns Firefox into a "Semantic Web browser", and also relies on a web service. The web service accepts an URL and scans the identified resource for embedded XMP metadata that is returned. Using the presented browser extension images and hyperlinks can be processed by the webservice with a mouse click, and returned XMP metadata is viewable in Piggy Bank. The novelty of the presented work is that is uses Piggy Bank to obtain RDF from non-textual resources (for example, from images on a web page).

References

[1] Adobe XMP `http://www.adobe.com/products/xmp/`

[2] Adobe XMP SDK `http://www.adobe.com/devnet/xmp/`

[3] Adobe XMP Specification `http://www.adobe.com/devnet/xmp/pdfs/xmp_specification.pdf`

[4] Apache Tomcat `http://tomcat.apache.org/`

[5] Firefox `http://www.mozilla.com/firefox/`

[6] FOAF `http://www.foaf-project.org/`

[7] Hypertext Transfer Protocol – HTTP/1.1 `http://www.ietf.org/rfc/rfc2616.txt`

[8] IsaViz `http://www.w3.org/2001/11/IsaViz/`

[9] Java API for XML Web Services (JAX-WS) `http://java.sun.com/webservices/jaxws/`

[10] JAX-WS Reference Implementation `https://jax-ws.dev.java.net/`

[11] Longwell `http://simile.mit.edu/longwell/`

[12] Mozilla `http://www.mozilla.org/`

[13] Piggy Bank `http://simile.mit.edu/Piggy_Bank`

[14] RDF/XML Syntax Specification `http://www.w3.org/TR/rdf-syntax-grammar/`

[15] RDF-Gravity `http://semweb.salzburgresearch.at/apps/rdf-gravity/`

[16] Resource Description Framework (RDF) `http://www.w3.org/RDF/`

[17] SIMILE Project `http://simile.mit.edu/`

[18] SPARQL Query Language for RDF `http://www.w3.org/TR/rdf-sparql-query/`

[19] Tabulator `http://www.w3.org/2005/ajar/tab`

[20] XHTML 2.0 `http://www.w3.org/TR/xhtml2/`

Peter Jeszenszky
Faculty of Informatics, University of Debrecen,
4010 Debrecen, P.O. Box 12
Hungary