Web-Style Multimedia Annotations

Ryan Shaw (Yahoo! Research)
Erik Wilde (School of Information, UC Berkeley)

UCB iSchool Report 2007-014
August 2007

Available at http://dret.net/netdret/publications#wil07k

Abstract

Annotation of multimedia resources supports a wide range of applications, ranging from associating metadata with multimedia resources or parts of these resources, to the collaborative use of multimedia resources through the act of distributed authoring and annotation of resources. Most annotation frameworks, however, are based on a closed approach, where the annotations data is limited to the annotation framework, and cannot readily be reused in other application scenarios. We present a declarative approach to multimedia annotations, which represents the annotations in an XML format independent from the multimedia resources. Using this declarative approach, multimedia annotations can be used in an easier and more flexible way, enabling application scenarios such as third-party annotations and annotation aggregation and filtering.

Contents

1 Introduction

The current wave of Web 2.0 applications has been fueled by the fact that for the first time in the history of the Web, there is a robust programming model which can be used to distribute client-side code which is accessing a server. This technology is generally known as *Asynchronous JavaScript and XML (Ajax)*, which refers to the fact that the scripting code in a Web page can communicate with a server exchanging XML data.

This availability of the *Web as a platform* for programming has fostered creativity and produced many interesting Web-based applications, but it has also caused problems, because the practice of Web 2.0 programming often looks significantly different from the theory. In theory, Web 2.0 application can be mashed together and work reasonably well in state-of-the-art environments. In practice, mashups often break because interfaces are being changed unilaterally and in an incompatible way; runtime problems are often caused by new browser and/or platform versions, Ajax libraries, or a lack of interworking between different libraries.

This development is of course not a surprise, it is the standard development which has to be expected in a freely programmable environment where imperative code is being developed and information is exchanged through traditionally designed APIs. However, the Web in its pure form champions a quite different architecture, one where applications work based on declarative data, which is a more robust approach to develop loosely coupled systems than traditional APIs.

In this paper, we describe an approach to recreate some of the declarativeness of the Web which has been displaced by the current wave of Web 2.0 applications. We describe how multimedia annotations, with our main example being picture annotations, can be based on a declarative approach, so that the annotations become more widely available to other applications. In addition, our approach separates resources from the annotations pertaining to them, thereby enabling annotations to read-only resources. While a declarative approach such as ours cannot replace the full programmability of the current Web 2.0 model, it strives at representing the most fundamental information about Web resources, leaving it open to imperative code to add the more advanced parts of an application scenario.

The work described in this paper is a part of the *AJAXLink* project, which, as part of the *Declarative Web 2.0* [20] approach, targets a declarative representation of Web 2.0 content, contrasting the imperative approach which is dominating most of the current Web 2.0 efforts. AJAXLink is a general framework for separating content and links in a Web-style style, and the work presented in this paper is the part of AJAXLink that is concerned with multimedia (mainly image) content.

2 Related Work

The ability to add annotations has been recognized as key requirement for enabling collaboration around digital documents [21]. For multimedia documents, annotations in the form of keywords or structured metadata can also be critical for allowing effective search and organization of document collections. Such annotations can sometimes be added automatically by classification or clustering algorithms, but these algorithms themselves usually require some manually annotated documents as training data. Given these needs, a large number of tools for multimedia document annotation have been developed by researchers and software developers. A complete review of these efforts is beyond the scope of this paper, but for illustrative purposes we consider here the range of tools currently available for annotation of image regions.

On one end of the spectrum are Web 2.0 tools for associating free-text descriptions with regions of images. Examples include Fotonotes,[1] a JavaScript tool which stores XML descriptions of image regions in JPEG file headers, and the region annotation features offered within photo-sharing sites such as Flickr.[2] Such tools are simple and easy to use, but lack any semantics which might make the annotations more useful. One

[1] `http://fotonotes.net/`

[2] `http://flickr.com/`

exception to this is the photo annotation tool offered within Facebook,[3] a popular social networking site. By constraining photo annotations to those that identify specific people, Facebook is able to create two-way links among photographs and the Facebook users depicted in them.

The possibilities of that kind of semantic linkage are what motivate the tools on the other end of the spectrum, which allow the linking of image regions to RDF vocabularies [8, 11]. These tools promise to allow sophisticated semantic retrieval and browsing of images, at the expense of rather more complex interaction scenarios. Moreover, there is still work to be done to establish standards and best practices for image annotation using Semantic Web technologies [19].

The Semantic Web tools cited above are desktop-based tools that require photos to be copied and imported before they can be annotated, instead of allowing stand-off annotation of images within the context of the Web. The Web 2.0 tools have the same problem, requiring either write access to the image files or the use of a specific proprietary image repository. In this paper we present a solution that offers the simple interaction of Web 2.0 tools with some of the benefits of more semantic approaches, by treating multimedia annotations as typed links. Moreover, we show that an open, distributed, and extensible system for multimedia annotation can be created by embracing the architectural principles of the Web, proving that the *Declarative Web 2.0* approach provides benefits over more closed and imperative implementations.

3 Declarative Web 2.0

The *Architecture of the World Wide Web* [9] generally favors the *separation of content, presentation, and interaction*, defining as good practice that "a specification should allow authors to separate content from both presentation and interaction concerns." Another issue is that for hypertext, "a specification should provide ways to identify links to other resources ..." and that "a data format should incorporate hypertext links if hypertext is the expected user interface paradigm."

The current popularity of Web 2.0 applications relies heavily on imperative code, combining the scripting power of JavaScript with the server access of the `XMLHttpRequest` object [18]. While this, as any general-purpose programming environment, is a very powerful way to build new applications and explore new ways of combining applications, it also is a step away from the original architecture of the Web, introducing tight coupling, accessibility problems [17], and new security issues [15] into the Web.

Using a declarative approach to represent the dependencies between resources on the Web, we have proposed the *Declarative Web 2.0* [20], which employs the *XML Linking Language (XLink)* [4] to describe the interrelations between resources which are relevant for an application. While XLink is used for the representation of resource interrelations, our approach has a wider focus, supporting external linkbases, and specifies an access protocol for these linkbases. This allows applications on the Web to combine resources and links connecting them from different sources, something that has been known in hypermedia systems for a long time, but so far has not been possible on the Web.

The goal of the Declarative Web 2.0 approach is to turn (some of) the current imperative programming of Web 2.0 applications back into the declarative style of the Web, making the Web 2.0 more accessible, more secure, and less tightly coupled. Furthermore, because our approach is orthogonal to media types, it can be used for multimedia resource types as well, allowing functionality which so far has been limited to closed settings (such as photo annotations) to become openly applicable on the Web.

4 Considerations for Multimedia

While the general principles of hypermedia linking and declarative representation of hypermedia structures are independent from any particular media type, it is important to note that fragment identification is more

[3] `http://www.facebook.com/`

important for multimedia datatypes, in particular for time-variant datatypes than it is for text-oriented datatypes. Even though there have been some proposals [14, 16], and a number of closed community efforts for fragments identifiers within certain multimedia presentation frameworks, currently there are no standardized fragment identifier formats for multimedia datatypes such as images, audio, and video.

Furthermore, while Web clients have more-or-less standardized on HTML for the presentation of text and images, there is still no standardized support for presentation and rendering of Web audio and video. The W3C's *Synchronized Multimedia Working Group* has presented the *Synchronized Multimedia Integration Language (SMIL)* [1] as a standard presentation language for these media, but currently only Internet Explorer supports even a subset of SMIL [12]. Even if there were widespread support for SMIL, however, there would still be a problem with incompatible audio and video formats, since SMIL does not specify any standard individual media formats or codecs that clients must support. Fortunately, the ubiquity of the Adobe Flash plugin for Web browsers means that, in practice, browsers can at least be expected to be capable of rendering MP3 audio and *Flash Video (FLV)*.

5 Implementation Variants

Our system consists of two parts: a Web service for storing and retrieving annotations using a standard protocol, and a client component for creating and rendering these annotations. It is expected that the client component will be running within a Web browser. When the user visits a Web document, the client component queries the Web service for annotations relevant to that document or any of the media objects (images, audio, or video) embedded within it. These annotations are then transformed into an format suitable for presentation and dynamically added to the document. The client also dynamically adds user interface controls which can be used to launch an editing interface for adding annotations to media objects embedded in the document, or to the document itself.

As stated above, we chose to encode our annotations as XLink extended links. Extended links (in their third-party usage) are intended to allow the creation of links among resources to which the link creator does not have write access, thus fulfilling our goal of creating an open annotation system. By the same token, extended links can be used to create links to and from multimedia formats that do not natively support linking, provided that a way of identifying media fragments has been established. Furthermore, using the `role` attributes available on XLink arcs and locators, these links can be associated with various vocabularies to express the semantics of the associations thus created.

Other formats for representing associations among resources are possible, of course, with the *Resource Description Framework (RDF)* [10] being an obvious candidate. RDF offers a richer language for expressing the semantics of relationships among resources than does XLink. But this expressivity comes at the price of more difficult processing using widely available tools and scripting languages. At the current time we do not feel that the benefits of using RDF syntax warrant this additional complexity. One compromise might be to use existing RDF vocabularies to define roles for XLink arcs and locators, an approach advocated by the microformats community.[4] This would allow usable RDF statements to be harvested from XLink linkbases [3]. Another way for supporting this harvesting of RDF would be the approach taken by *Gleaning Resource Descriptions from Dialects of Languages (GRDDL)* [2].

An added benefit of using XLink is that we can take advantage of its built-in vocabulary for describing various types of link traversal behavior. While technically link traversal is presentation-specific and should be separated from the semantics of annotations, in practice it may be convenient to supplement semantic descriptions of interactive annotations with hints regarding their intended behavior. Clients are still free to use local preferences and styling to determine the actual way annotation information is displayed through link traversals.

[4] `http://microformats.org/about/`

The presentation layer might be implemented in a variety of ways. To follow the precepts of Declarative Web 2.0, it would be best if our annotation language could be transformed into a declarative representation of our desired annotation display. Annotations of media with spatial extent (such as bitmap images or vector graphics) might be presented using *Scalable Vector Graphics (SVG)* [5], while annotations of media with temporal extent could be presented using SMIL. In this way the presentation and interactive behavior of annotations could be easily modified client-side through the use of stylesheets.

Unfortunately, client-side Web technology is still a rather wild mixture of partially implemented standards and semi-compatible technologies, making this a goal impossible to meet using current Web browsers. A more realistic approach is to use JavaScript to transform XLinks into whatever presentation languages are supported by the client environment, whether this is a declarative presentation language, imperative code, or some mixture of the two. For example, annotations on an image might be transformed into SVG for Mozilla-based browsers, which support native SVG display, and into the *Vector Markup Language (VML)* for Internet Explorer. Likewise, audio or video annotations could be transformed into HTML+TIME (a subset of SMIL) for Internet Explorer, and a series of imperative JavaScript calls to the Adobe Flash plugin for Mozilla-based browsers, which do not support SMIL.

It is hoped that over time more and more options for declaratively describing multimedia presentations will be available, resulting in fewer dependencies on complex imperative code. There are some signs that this is happening: for example, recent versions of Adobe Flash can dynamically load declarative descriptions of animation and interaction behavior, albeit in the proprietary *MXML* format. Microsoft is also introducing declarative user interface programming through its proprietary *Extensible Application Markup Language (XAML)*.

6 Implementation

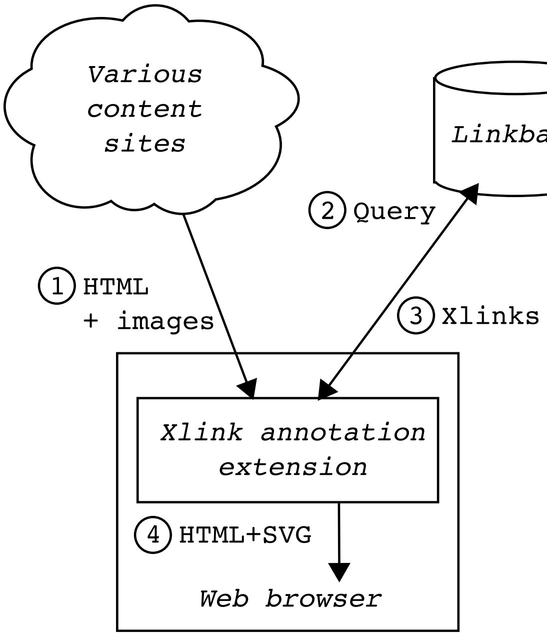
To prototype the system described above, we developed an extension for the Firefox Web browser. For the initial implementation we decided to focus on image annotation, taking advantage of Firefox's native SVG support. The extension communicates with a remote XLink linkbase using a REST [6] protocol similar to the *Atom Publishing Protocol (APP)* [7][5]. Users with the extension installed can view and follow hyperlinked annotations on images embedded in the Web pages they visit; and using a simple authoring interface, they can create and save new annotations that then are visible to any other user with the extension installed.

As shown in the figure, each time an HTML page is loaded by the browser, the extension scans the page for embedded images. For each image found (excluding very tiny images used for formatting and styling), the extension queries the linkbase for any XLinks that define arcs that start at that image, or some region within the image. As noted above, there is no established standard for identifying regions of images, so we use an ad-hoc convention of identifying rectangular regions using URI fragment identifiers of the form `#rect(x,y,w,h)`, where `x` and `y` specify the pixel coordinates of the top left corner of the region, and `w` and `h` specify its width and height in pixels. It is easy to imagine how such a scheme might be extended to arbitrary SVG paths to identify non-rectangular regions, as suggested in by Rutledge and Schmitz [16].

If any XLinks are returned by the query, the annotated rectangular regions described by the XLink locators are displayed as SVG elements superimposed on the the image. These annotations can be clicked on to trigger the specified actuation behavior (usually traversal of the link to the annotation endpoint).

Unfortunately, our implementation of link traversal is also an example of current limitations on declarative Web programming. Although SVG supports hyperlinking using XLink, it is not possible to design a good user experience of more complex link traversals using a purely declarative specification of link behavior. In the common case of images that are already contained within HTML anchors, the dynamic overlay of

[5] We do not yet use APP, but intend to switch over to APP in the next design cycle of the extension. After that switch, linkbases will be modeled as Atom collections, making them readily accessible to Atom-based processing methods, so that for example a linkbase can be represented as an Atom [13] feed.



hyperlinked region annotations creates some ambiguity about what should happen when the user clicks on the annotated region.

Ideally, clicking within the annotated region would actuate the link associated with the annotation, while clicking outside of the region would actuate the original link associated with the image. Achieving this kind of interaction requires some control over the way events are bubbled up through the document structure. Unfortunately, there is currently no way to declaratively specify how events should propagate through a compound SVG+HTML document, and thus link traversal must be implemented imperatively using JavaScript.

For the same reason, the annotation authoring interface is also implemented in JavaScript. To create new annotations, the user simply mouses over any image in any HTML page being viewed. This causes an "Add XLink" button to appear in the upper-left corner of the image. Pressing this button results in an SVG rectangle to be superimposed on the image, which can be dragged to whatever part of the image the user wishes to annotate. The dimensions of the rectangle can be interactively changed by dragging the rectangle's sides and corners. When the user has thus defined the region to be annotated, he simply presses Enter and is prompted for a URL to which the region should be linked. After entering the URL, a new XLink is created and posted to the linkbase.

It should be noted that our prototype implementation uses a predefined set of semantic roles for the XLink arcs and locators in order to simplify the authoring process. This is appropriate for the generic annotation scenario described here, but more specialized implementations for specific domains would need to have some way for annotation authors to specify the semantics of the XLinks they create. The specifics of these interactions are to be considered in future work.

7 Future Work

While a quality user experience still requires some scripting of interaction behavior, our prototype demonstrates that a declarative approach to open, distributed semantic annotation is possible using current technology. However, a distributed multimedia annotation system built around a single shared linkbase only scratches the surface of what might be done. In this section, we consider some avenues for future work, focusing on the possibilities of third-party semantic annotation and linking of multimedia documents.

One possibility is a system for subscribing to annotations created by specific authors or groups of authors. Just as many people currently subscribe to specialized blogs to read commentaries on and reactions to recent news articles, users could subscribe to specialized annotation feeds. For example, the National Academy of Sciences might publish an annotation feed for major newspapers, adding commentary and linked resources to science-related articles. Instead of individually viewing many independent commentaries on some Web document, annotation feeds would allow all the relevant commentary to be viewed in the context of the document being commented upon. Existing approaches to filtering and aggregating XML feeds could also be applied to annotation feeds, allowing users to craft personalized lenses for viewing the Web.

The utility of such an approach is especially apparent when the document being commented is non-textual. Rather than watching a 10-minute video, and then viewing a sequence of independent blog posts or video responses which address specific segments of that video, an annotation-aware client could produce a dynamic presentation intermingling the ordinal video and third-party commentary. Unwanted commentary (as defined by the viewer, rather than the media owner or hosting site) could be easily filtered out.

Furthermore, publishing of annotation feeds need not be limited to human authors. Multimedia content analysis tools could also publish to distributed annotation linkbases. Face recognition algorithms might be used to automatically publish an annotation feed linking famous faces within photographs to Wikipedia articles. An automated feed for language learners might add multilingual annotations to recognized everyday objects. Instead of using specialized portals, or waiting for their favored image publishing sites to adopt such technologies, users could add these functionalities directly to their browsers.

8 Conclusions

The declarative multimedia annotation approach described in this paper is an attempt to bring some of the current wave of Web 2.0 application programming back in line with the architecture of the Web. As with any declarative approach, it is limited in its expressiveness and can only represent the concepts which have been included in the declarative format. This means that for applications outside of this area, imperative programming is still required. However, our goal is to represent the most fundamental aspects of multimedia annotations (which resources or parts of resources are annotated or connected with other resources or parts of resources), making this information available as an XML-based format and through a REST protocol.

While the current fast pace of Web 2.0 developments will need more time to settle and to see what parts of application architectures should be provided in a more Web-style way, we believe that links between resources are a very good candidate for such a factoring out of reusable concepts[6]. The Declarative Web 2.0 approach helps to represent some of the most important data about resources into the realm of declarative formats, and we believe that this enables a wider reuse of this data in unforeseen application scenarios than the more closed approaches which have been used so far.

[6] The area of offline (i.e., local) data access is another area where this will probably happen after an initial period of experimentation, which has seen its first important contribution with the recent release of *Google Gears*.

References

[1] Dick Bulterman, Guido Grassel, Jack Jansen, Antti Koivisto, Nabil Layaïda, Thierry Michel, Sjoerd Mullender, and Daniel F. Zucker. Synchronized Multimedia Integration Language (SMIL 2.1). World Wide Web Consortium, Recommendation REC-SMIL2-20051213, December 2005.

[2] Dan Connolly. Gleaning Resource Descriptions from Dialects of Languages (GRDDL). World Wide Web Consortium, Proposed Recommendation PR-grddl-20070716, July 2007.

[3] Ron Daniel. Harvesting RDF Statements from XLinks. World Wide Web Consortium, Note NOTE-xlink2rdf-20000929, September 2000.

[4] Steven J. DeRose, Eve Maler, and David Orchard. XML Linking Language (XLink) Version 1.0. World Wide Web Consortium, Recommendation REC-xlink-20010627, June 2001.

[5] Jon Ferraiolo, Jun Fujisawa, and Dean Jackson. Scalable Vector Graphics (SVG) 1.1 Specification. World Wide Web Consortium, Recommendation REC-SVG11-20030114, January 2003.

[6] Roy T. Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, Irvine, California, 2000.

[7] Joe Gregorio and Bill de Hóra. The Atom Publishing Protocol. Internet Draft draft-ietf-atompub-protocol-15, May 2007.

[8] Christian Halaschek-Wiener, Jennifer Golbeck, Andrew Schain, Michael Grove, Bijan Parsia, and James Hendler. PhotoStuff — An Image Annotation Tool for the Semantic Web. In Yolanda Gil, Enrico Motta, V. Richard Benjamins, and Mark A. Musen, editors, *Poster Proceedings of the 4th International Semantic Web Conference*, Galway, Ireland, November 2005.

[9] Ian Jacobs and Norman Walsh. Architecture of the World Wide Web, Volume One. World Wide Web Consortium, Recommendation REC-webarch-20041215, December 2004.

[10] Graham Klyne and Jeremy J. Carroll. Resource Description Framework (RDF): Concepts and Abstract Syntax. World Wide Web Consortium, Recommendation REC-rdf-concepts-20040210, February 2004.

[11] Mathias Lux, Jutta Becker, and Harald Krottmaier. Semantic Annotation and Retrieval of Digital Photos. In Johann Eder and Tatjana Welzer, editors, *Short Paper Proceedings of the 15th Conference on Advanced Information Systems Engineering*, volume 74 of *CEUR Workshop Proceedings*, pages 85–88, Klagenfurt, Austria, June 2003. Technical University of Aachen (RWTH).

[12] Debbie Newman, Aaron Patterson, and Patrick Schmitz. XHTML+SMIL Profile. World Wide Web Consortium, Note NOTE-XHTMLplusSMIL-20020131, January 2002.

[13] Mark Nottingham and Robert Sayre. The Atom Syndication Format. Internet proposed standard RFC 4287, December 2005.

[14] Silvia Pfeiffer, Conrad D. Parker, and Andre T. Pang. Specifying Time Intervals in URI Queries and Fragments of Time-Based Web Resources. Internet Draft draft-pfeiffer-temporal-fragments-03, March 2005.

[15] Paul Ritchie. The Security Risks of AJAX/Web 2.0 Applications. *Network Security*, 2007(3):4–8, March 2007.

[16] Lloyd Rutledge and Patrick Schmitz. Improving Media Fragment Integration in Emerging Web Formats. In *Proceedings of the 8th International Conference on Multimedia Modeling*, Amsterdam, Netherlands, 2001. November.

[17] Fred Sampson. Sense and Accessibility. *interactions*, 14(3):10–11, 2007.

[18] Anne van Kesteren. The XMLHttpRequest Object. World Wide Web Consortium, Working Draft WD-XMLHttpRequest-20070618, June 2007.

[19] Jacco van Ossenbruggen, Raphaël Troncy, Giorgos Stamou, and Jeff Z. Pan. Image Annotation on the Semantic Web. World Wide Web Consortium, Working Draft WD-swbp-image-annotation-20060322, March 2006.

[20] Erik Wilde. Declarative Web 2.0. In Weide Chang and James B. D. Joshi, editors, *Proceedings of the 2007 IEEE International Conference on Information Reuse and Integration*, pages 612–617, Las Vegas, Nevada, August 2007.

[21] Robert Wilensky. Digital Library Resources as a Basis for Collaborative Work. *Journal of The American Society for Information Science and Technology*, 51(3):228–245, 2000.