

What Would It Mean to Blog on the Semantic Web?

David R. Karger¹ and Dennis Quan²

¹ MIT Computer Science and Artificial Intelligence Laboratory
32 Vassar Street
Cambridge, MA 02139 USA
karger@mit.edu

² IBM T. J. Watson Research Laboratory
1 Rogers Street
Cambridge, MA 02142 USA
dennisq@us.ibm.com

Abstract. The phenomenon known as Web logging (“blogging”) has helped realize an initial goal of the Web: to turn Web content consumers (i.e., end users) into Web content producers. As the Semantic Web unfolds, we feel there are two questions worth posing: (1) do blog entries have semantic structure that can be usefully captured and exploited? (2) is blogging a natural way to encourage growth of the Semantic Web? We explore empirical evidence for answering these questions in the affirmative and propose means to bring blogging into the mainstream of the Semantic Web, including ontologies that extend the RSS 1.0 specification and an XSL transform for handling RSS 0.9x/2.0 files. To demonstrate the validity of our approach we have constructed a semantic blogging environment based on Haystack. We argue that with tools such as Haystack, semantic blogging will be an important paradigm by which metadata authoring will occur in the future.

1 Introduction

The Web is arguably the most successful open information distribution mechanism in existence. Despite its success, a few issues were not worked out initially, such as easy publication and machine-readable metadata. Web logs (“blogs”) have emerged as a potential solution to the publication problem. The idea is based on the premise that publication occurs incrementally in discrete units—blog entries—and that users manage their own content (as opposed to newsgroups). A number of different software packages have grown around this simplified abstraction. Separately, the problem of machine-readable content is being attacked by the Semantic Web. Here the idea is that Web sites would host content in a language such as RDF, mostly devoid of human-readable artifacts such as prose, instead opting for more precise machine-readable specifications [7].

On the surface, these efforts seem to have little to do with each other, besides being both based on the Web. However, we argue that they are actually bound to converge at a common destination—a metadata-rich version of today’s Web. Even today, there is a lot of family resemblance between blogs and the Semantic Web model. Blogs enable the average user to talk about—i.e., annotate—resources on the Web and pub-

lish these annotations for others to see. A large portion of these blogs already have machine-readable “table of contents” files encoded in an XML format called RSS. Furthermore, blog entries themselves are first class—blog entries can be searched over, replied to, and referred to in other blogs.

In this paper we wish to examine two questions. First, if blogs could take full advantage of the RDF representation, what benefits would be realized? Second, do blogs form the basis of a nascent Semantic Web? In pursuit of answers to these questions, we characterize the notion of *semantic blogging*—the publication of RDF-encoded Web logs. Furthermore, to explore the practical benefits of semantic blogging, we have constructed a prototype semantic blogging tool on top of Haystack, our Semantic Web metadata exploration, visualization, and authoring tool [1, 8].

1.1 Approach

Our approach to semantic blogging focuses on adding unobtrusive extensions to the current blog creation interface paradigm—one of the key elements of the success of blogging—while producing machine-readable content for the benefit of blog content consumers. We argue that there is no added user effort involved in creating a semantic blog versus a conventional one, because the real work is done by the software in capturing semantics that are already being provided by the user. For example, when the user clicks on the “blog this” button on the Google toolbar in the Web browser to blog about the current page [19], instead of just recording a hyperlink, the system can record the fact that the current Web page is the *topic* of the blog entry. (Furthermore, one is not bound to using Google’s blogging service.) In addition to these and other semantics inherent to blogs today, we also provide a mechanism for more advanced users to embed arbitrary RDF in a blog using a forms mechanism similar to that seen in Annotea [6].

In either case, the consumers of blog content benefit from having information from multiple blogs and the rest of the Semantic Web being integrated together and displayed in a variety of ways. Instead of being restricted to viewing one blog at a time, users can view cross-blog reply graphs to track the flow of a conversation, without requiring that publication be controlled by a central authority, as with a newsgroup. When browsing Web or Semantic Web content, one can see opinions, instructions, or descriptions from blogs alongside the content itself, as is done in tools such as the Haystack Semantic Web browser and Annotea [6]. Possibilities for automation are also created from the precise recording of semantics such as movie or product ratings, interest levels, and statistics such as sports scores.

1.2 Outline of the Paper

The paper is organized as follows. First, we first explore how blogs are written and used today and compare and contrast the blog paradigm with other Web publication paradigms. Based on these observations, we propose ontologies that extend the RSS 1.0 specification [12] and enable one to capture the semantics of blog entries in RDF. Afterwards, we describe our prototype semantic blogging environment, which allows users to both read existing blogs and to maintain their own blogs. We describe an XSL transform that translates existing RSS 0.9x/2.0 feeds [13, 14] into RDF, allowing

us to take advantage of both existing and new RDF-enabled blogs. Finally, we put the pieces together and outline a scenario of how semantic blogging can enable more powerful forms of information retrieval and automation across blogs and other Semantic Web content.

2 The Essence of Blogging

At its core, blogging incorporates three distinct, key concepts, and analyzing blogging with respect to these concepts helps to clarify how blogging (and by extension semantic blogging) compares with other approaches. The first concept is of enabling users to publish information in small, discrete notes, as opposed to large, carefully-organized Web sites. Blogger remarks that “blog posts are like instant messages to the web” [9]. This analogy is borne out by the observation that interfaces for popular blogging tools including Blogger, Moveable Type [29], and Radio Userland [10] resemble those used by Web-based e-mail systems such as Hotmail [28]. Looking purely from this perspective, one might also note that the same kinds of brief, subjective comments can be found in the feedback bulletin boards of Web sites such as Amazon.com, Epinions.com, and Slashdot.org. Similarly, systems such as Annotea, which allows users to attach RDF-encoded “sticky note” annotations to text on Web pages [6], accumulate content one note at a time. Of course, blogs are not restricted to being lists of comments and criticisms; a wide variety of content exists in blogs today [24].

The second key concept is decentralized, per-user publication. Blog entries very often take the form of annotations to pages or critiques of products, but blog entries are primarily kept together based on common authorship, not common subject. In other words, the content author is the one who controls publication, and the feeling of “ownership” is part of the blogging experience. Bloggers write on a variety of topics and categorize their content as they choose, in contrast to a discussion site or a newsgroup, where deviation from predefined topics is often frowned upon and can result in censorship. Additionally, bloggers have control over the structure of individual blog entries. In a discussion group or with an annotation server, users are sometimes forced to conform to preset rating schemes or other attribute sets.

Furthermore, blogs generally exist independent of a centralized server. In this sense, maintaining a blog is conceptually similar to maintaining a single-user version of Annotea that has been reformulated to publish a table of contents of recent annotations to selected Web sites. However, the lack of a central point of aggregation has its disadvantages. With the Annotea model, where annotations are stored on a centralized server, it is easy to determine what other annotations exist for a resource of interest. Similarly, with the Internet newsgroup model, it is a simple task to trace reply chains because messages are grouped by thread. Later in the paper, we discuss ways in which semantic blogging can be used to overcome these limitations.

The last key concept—exposing machine-readable listings—is a property that many blogs possess in order to allow individual blog entries to be aggregated together. A family of XML-based standards for describing the contents of a blog loosely affiliated by the acronym “RSS” plays an important role in the thesis of this paper. The Really Simple Syndication 0.92 [14] (and its related standards hereafter referred to as RSS 0.9x) and 2.0 standards originated by Dave Winer [13] are not

based on RDF but are by far the most widely adopted. The RDF Site Summary 1.0 standard [12] encodes essentially the same information and is based on RDF, but relatively few sites produce RSS files to this specification. There are other specifications for XML-based blog listing formats, such as Atom [17], which contain similar information. What is agreed upon is the basic notion of producing a machine-processable listing of blog entries in an XML format, and we build upon these XML formats in this paper.

Machine-readable listings are often used by blog readers, such as SharpReader [21] and NetNewsWire [20], as well as those built into blogging tools such as Radio Userland mentioned earlier. Like newsgroup readers and e-mail clients, blog readers typically offer users the ability to view lists of blog entries sorted by date, name, and other attributes—even lists that span multiple blogs. In order to emulate the threaded message displays found in other messaging tools, blog readers such as SharpReader find embedded hyperlinks in common between different blog entries and use them as the basis for determining threading. With semantic blogging we attempt to find cleaner ways to support the already-extant notion of inter-blog threading.

3 Bringing Out the Hidden Semantics in Blogs

Superficially, the existence of machine-readable listings is the strongest link between blogging and the Semantic Web. The connection, however, runs deeper. The discrete units being published in bulletin boards, in blogs, and with Annotea often have considerable internal structure: e-mails, instant messages, annotations, and bulletin board posts have a from field, a send date, and sometimes more specific information such as the product or other object being reviewed, a rating, etc. Put another way, the process of blogging inherently emphasizes metadata creation more than traditional Web publishing methodologies (e.g., direct HTML composition). The success of blogging points to its role as a socially viable means for encouraging users to publish certain forms of metadata. In this section, we elaborate on three specific forms of metadata that are inherent to the process of blogging. We also discuss possible benefits that would be realized from more precise, machine-readable metadata records.

3.1 Blogs as Annotations

Bloggers are free to write about anything they want in their blogs, from what they ate for breakfast that morning to the traffic problems on the commute home. Despite this freedom, bloggers frequently comment on things that exist on the Web. More specifically, bloggers spend much of their time creating focused annotations of Web resources. This observation is of interest to us to the extent to which these annotations can be modeled by taxonomies and ontologies.

Blog entries that talk about Web resources can be classified into a number of different types, and different annotation types can be modeled by differing levels of structure. For example, it may be useful to model criticism annotations using a numerical rating. At the moment, blog entries that embody criticisms may contain a numerical rating but that rating is rarely recorded in a machine-readable form, although there have been proposals based on non-RDF formats such as structured blog-

ging [25] or Review Module for RSS 2.0 [23]. Using an ontology to model such annotations would make it easier to do automated filtering and analyses, such as finding the average rating for a given resource; such analyses are already done by specialized data mining sites such as blogosphere.us [26], but they usually do not track anything more semantic than just the frequency with which specific sites are hyperlinked to. Also, because blogs are managed on a per-user basis, users have the flexibility to adopt such ontologies to mark up their annotations.

Additionally, in moving blogging to the Semantic Web, an obvious extension is allowing blogs to talk about arbitrary Semantic Web resources. This extension not only broadens the set of available topics but also allows resources that previously had to go unnamed (e.g., “teddy bear model #22321”) to be identified more precisely, improving search capabilities.

3.2 Blogs as Message Chains

Emphasizing a point from Section 2, blogs act as logs of messages, usually written to a general audience but at times focused towards specific parties. This phenomenon is most evident when an event of interest occurs, such as a product release. A flood of blog posts appear coincidentally, and the debate that ensues results in blogs containing entries that comment on other blog entries [5]. This structure is reminiscent of e-mail or newsgroups: a blog entry has a sender, a set of (sometimes targeted) recipients, a subject line, and often one reply-to entry (or more, although today noting that a blog entry is in reply to another blog entry may need to be done with human-readable prose and a hyperlink, depending on the blogging system in use).

3.3 Blogs as *Ad Hoc* Tables of Contents

Many blogs are devoted to a specific topic; alternatively, many bloggers intentionally divide their content into topics. The collections of commentary presented in these blogs often end up serving as a useful introduction to some esoteric subject. In other words, blogs can act as tables of contents for some field of interest. This combination of intentionally-recorded domain knowledge and a machine-readable listing format creates a number of possibilities for enhancing information retrieval. The main problem with a blog serving as a permanent archive of introductory material is that RSS listings produced by blog servers typically only include the n most recently published blog entries. One way of overcoming this problem would be to have blog servers also produce archival “back-issue” RSS files. We discuss alternative ways of dealing with this issue throughout the paper.

4 Ontologies for Enabling Semantic Blogging

Having identified the core semantics imparted by blog entries, we proceed to use Semantic Web tools, including RDF and OWL, to find ways to capture these semantics in a standard fashion. In this section we describe various ontologies and strategies for recording blog entries in RDF. We refer to blogs that have been represented in this

fashion as “semantic” blogs to highlight that important elements of blog entries that were once recorded in prose are now being described with machine-readable meta-data.

4.1 Building on RSS 1.0

There are many benefits to using RDF as the basis for recording machine-readable blog listings. Most important of these is that the notion of resource identity is built into RDF: resources are always referred to by their URIs. This provision simplifies processes such as notating what resource is being annotated or what blog entry is being replied to. Also, there is a well-defined mechanism for adding custom properties or attributes to blog entries. Furthermore, there are well-defined semantics for merging multiple RDF files. As a result, one can accumulate the RSS files generated by a blog over time and keep a historical record of the blog easily.

As noted earlier, RSS 1.0 is an already-extant RDF-based mechanism for publishing machine-readable blog listings. While RSS 0.9x/2.0 is the predominant format, many blogs also provide listings in RSS 1.0 format. Furthermore, most of the concepts in RSS 0.9x/2.0 map directly onto the RSS 1.0 ontology. In Section 5.1 we discuss an XSL transform for converting RSS 0.9x/2.0 files into RDF. These observations make RSS 1.0 a natural standard to build on.

4.2 Categorization

More and more, bloggers have begun to categorize their individual blog entries. Categorization allows a single blogger to distinguish multiple trains of thought or topic areas in a single blog. The RSS 0.92 and 2.0 standards support categorization, but apart from a URI naming a taxonomy, category labels themselves are just strings. The RSS 1.0 standard does not include any explicit support for categorization.

The Haystack ontology, developed for the Semantic Web Browser project discussed later [1], defines a class called **hs:Collection**, and one of the roles of a collection is to act as a category for classification purposes. A single predicate, **hs:member**, binds a collection to its members, regardless of type. In this way, collections named by URIs are used as the basis for category labeling and can hence be shared by multiple blogs. Universal naming also avoids name collisions between blogs that have coincidentally named categories (e.g., “Jaguar” in a zoologist’s blog and “Jaguar” in an operating system researcher’s blog are likely to mean different things) and facilitates the construction of mappings between different categorization schemes.

4.3 Message Ontology

Blog entries, as discussed earlier, have a lot in common with other forms of electronic messages, such as e-mails and instant messages. A previous paper on modeling messages in RDF [3] proposed an ontology for messaging, and we have reused the portions of this ontology that are applicable to blogging. At the heart of this ontology is the **msg:Message** class, which is the base class for all messages. We define **rss:item** from the RSS 1.0 ontology as deriving from **msg:Message**. This allows us to reuse

the **msg:inReplyTo** predicate, which declares one message to be in reply to another message. (**msg:inReplyTo** is simply a sub-property of **ann:annotation**, which is used to indicate the resource being annotated by a message.)

An argument that was presented in the earlier paper on messaging was that conversations are often spread across multiple messaging channels, such as e-mail, instant messaging, and chat. Blogging has a similar phenomenon: not only do users send e-mails to bloggers in response to their blog entries or use the comment bulletin board feature attached to many blogs, but they also respond to others' blog entries in their own blogs. E-mails, bulletin board messages, and blog entries are all acting as messages within a greater conversation, and using a general messaging ontology enables us to capture these semantics in a uniform fashion across an entire conversation. More specifically, predicates such as **msg:to** and **msg:from** are used to characterize e-mails and instant messages, and in the case of a blog posting, we use a "blog audience" resource associated with the blog for the recipient (i.e., value of the **msg:to** property).

4.4 Encoding Custom Semantics

Our messaging ontology defines a **msg:body** predicate, which is used to associate a message with textual content. However, the value of the **msg:body** property need not be human-readable. By allowing an RDF file to serve as the body of a blog, we can enable arbitrary Semantic Web content to be included. While we do not see this sort of "pure" semantic blogging taking off immediately, we feel that this provision provides space for semantic blogging to progress in the future.

Haystack supports a form-based mechanism for allowing users to input a variety of different kinds of RDF metadata. Forms for entering specific kinds of semantics are likely to be generated in accordance with specific RDF Schemas or ontologies. Haystack includes support for schema-based form generation, similar to those found in Annotea [6] and Protégé [16]. Our forms mechanism is described in previous work [15].

5 A Semantic Blog Client

To demonstrate the benefits of semantic blogging, we constructed a semantic blog client built into Haystack that incorporates publishing, aggregation, and browsing capabilities. Haystack's paradigm is similar to that of a Web browser; users navigate to resources by typing in their URIs or by clicking on hyperlinks, and the toolbar exposes the familiar back, forward, refresh, and home buttons for ease of navigation. However, unlike a Web browser, which simply renders the HTML content of a resource, Haystack contains user interface templates called *views* that are used to transform machine-readable RDF metadata relating to the current resource into a human-readable, hyperlinked, on-screen presentation. We have extended Haystack with specialized views and user interface commands called *operations* that enable blogging functionality within Haystack. The technical details of how views and operations are defined are described in previous papers on Haystack [1, 8]. In this section we characterize these views and operations and show how they contribute to the end user experience.

5.1 Subscribing to Blogs

Most blogs are published as Web sites, allowing users with a normal Web browser to view and search them. Haystack includes a Web browser view, meaning that when a user browses to a Web resource (i.e., a Web page named by an HTTP URL), Haystack's functionality reduces to that of a traditional Web browser. As a result, users can browse the HTML versions of blogs from within Haystack. In addition, most blogs also provide an RSS file that can be used to subscribe to a blog, as noted earlier. By convention, an "RSS" or an "XML" button is placed on the blog's home page to signify the availability of an RSS file, and when the user clicks on the button, the browser navigates to the RSS file. When the user browses to an RSS file, Haystack detects that it is an RSS file and allows the user to subscribe to it using the Subscribe to RSS Feed operation. After the user subscribes to the blog, Haystack will automatically download the RSS file on a regular basis. Over time, Haystack accumulates a historical record of a blog, which can be useful when a blog contains helpful reference material that may want to be looked up later.

Today, relatively few blogs are encoded in RDF, and since Haystack deals primarily with RDF-encoded metadata, RSS 0.9x/2.0 files need to be converted into RDF before they can be subscribed to. The mechanism we have elected to use in handling the various forms of RSS currently extant on the Web today is to transform RSS 0.9x/2.0 files into RDF files based on RSS 1.0 and our extension ontologies discussed earlier. Being able to convert RSS 0.9x/2.0 files into RDF enables a number of powerful Semantic Web capabilities to be used over a large, pre-existing corpus of blogs, such as search, semantic exploration, etc.

The XSL Transformations language (XSLT) is a standard means for transforming one XML format into another, and we have implemented our translation process in XSLT. The complete source for our XSLT can be found on our Web site¹. The majority of the XSLT converter is straightforward and simply converts RSS 0.9x/2.0 syntax into well-formed RDF. The primary challenge is coming up with a URI for blogs and blog entries. Fortunately, the permalink feature of RSS 0.9x/2.0 allows a blogger to assign a permanent URL name to a blog entry, but its use is optional. When it exists, our XSLT uses it as the blog entry's URI; otherwise, it defaults to the value of the link element. Many features of semantic blogging are missing from RSS 0.9x/2.0, but it is possible to use techniques such as scraping for hyperlinks as to fill in the list of resources being annotated or replied to as an approximation.

5.2 Viewing Blogs

Once a user has subscribed to a blog, it appears on the news home page in Haystack. This screen is like the front page of a newspaper in that it shows the recent news articles from all of the subscribed feeds. By using the preview pane controls on the toolbar, the user can introspect individual articles: when the user clicks on a link on the news pane, the associated article appears in the pane on the right.

In addition, individual blogs can be viewed separately. The list of entries in a blog (sometimes known as a channel) maps onto Haystack's collection concept introduced earlier. There are a number of collection views built into Haystack, and as a result,

¹ <http://haystack.lcs.mit.edu/rss.xslt>

blogs can be visualized in many different ways, including as a list, a calendar, or a multi-column table. In addition, some specialized views, such as the Explore relationships between collection members view, are provided to aid in certain forms of information exploration. In this case, the Explore relationships between collection members view displays a labeled directed graph, drawing the members of the collection (in this case, the articles in the blog) as nodes and allowing the user to select which arcs (i.e., predicates) to show between the articles. The built-in Reply graph arc set is selected in Fig. 2 and shows the **msg:inReplyTo** relationships between the messages in the display.

5.3 Organizing Blog Content of Interest

In Section 4.2 we discussed the use of collections as a categorization mechanism for bloggers. These collections are first class resources and browseable in their own right from within Haystack. In addition, a user reading a blog can also create his or her own collections and use them to file blog entries of interest. Looking at an article, the user can invoke the File away operation on the left hand pane. This operation reveals a customizable list of collections on the right hand pane into which the article can be filed. Each collection has a checkbox next to it, making it easy to file an article into multiple collections at once [11]. Collections can also be hierarchical; adding a new collection can be accomplished by right-clicking on the parent collection and selecting Add new item/Create a collection from the context menu.

5.4 Creating a Semantic Blog

We have implemented support for allowing users to blog from within Haystack. As noted earlier, blogging tools expose forms that resemble an e-mail editor for creating new blog entries. In Haystack, we have adapted the existing mail composition and transmission functionality to be able to post messages to servers (including possibly a local server built into Haystack, as in the case described below) that speak protocols like the Blogger XML-RPC protocol [30].

To implement our enhanced semantic capabilities, we have extended the Blojsom Open Source blog server, which runs on a standard Java Servlet-compliant Web server such as Apache Tomcat. Blojsom stores blog entries in text files on the file system. (One benefit of the file system approach is that a user can work on a local copy of the blog and use CVS to synchronize it with a blog server; Haystack includes built-in support for CVS.) These files contain a header that includes a one-line title for the blog entry and a series of metadata declarations. We have introduced five metadata tags, meta-inReplyTo, meta-annotates, meta-fileInto, meta-rdfBody, and meta-uri² in order to support our semantics, and have written a new Haystack messaging driver that allows Haystack to serialize blog entries in this form.

² This declaration permits the file to state the URI to be assigned to the blog entry and allows, in theory, for a blog entry to be posted to multiple blogs but retain the same URI. The `rss:link` property, which provides the URL of the content for the blog entry, can be specific to the blog server without disrupting this scheme.

Users gain access to blogging functionality through a combination of views and operations. First, a user clicks on the Blogging link in Starting Points. Haystack then shows a list of the user's existing blogs. To connect to an existing blog server via XML-RPC or to create a new semantic blog, the user can choose either the Connect to Blog Server or Create New Semantic Blog operations from the left hand pane, respectively. After filling in the appropriate configuration parameters, the user is taken to a screen that shows the blog entries that exist for that blog. Creating a new blog entry is accomplished by selecting the New text blog entry operation from the left hand pane.

The key benefits of blogging in Haystack versus current blogging tools (or even an e-mail client) are twofold. First, Haystack can produce specialized forms that capture various levels of semantics. At one end of the spectrum are simple message editors such as those seen in today's blogging tools and Web-based e-mail products. At the other extreme are specialized forms, such as those proposed by structured blogging [25] or used by Annotea [6], which can be derived from RDF Schemas automatically by Haystack.

The other benefit is that the semantics surrounding the reason for a blog entry's creation can be captured immediately. When the user is viewing a Web page, a Semantic Web resource of interest, or even another blog entry, and wishes to blog about it, he or she can click on semantically-specialized operations such as "Blog a reply" or "Blog a complaint". In contrast, those semantics are often lost in the transition between the program the annotated resource is being viewed in (e.g., the Web browser) and the blogging tool.

6 A Scenario

To concretize the benefits of our approach, consider the following fictitious scenario. John Doe is a bioinformatician who is thinking about whether to attend the Semantic Bio 2005 conference. He goes to the conference Web site and sees an overwhelming 20 page listing of all of the papers to be presented. Like many futuristic (and fictitious) conference sites, this site also offers an RDF version of the program. Using Haystack, John downloads the program and uses Haystack to browse it.

John takes a look at the multi-day Inference Track category to see just those papers. One in particular catches his eye—one called "Using the COOL Algorithm for Protein Folding Analysis". Clicking on the article, he sees the paper's abstract and references. John, like many people, has friends who are avid bloggers and maintain blogs that review the latest findings in their particular fields. These friends have already seen this paper and blogged about it; their blog entries appear alongside the article. Refer to Fig. 1.

Glancing through the subject lines, it isn't clear which blog entry came first and which came last. John clicks Extract Relevant Discussion, selects the Explore Relationships between Collection Members view, and selects the Message Reply Graph arrow set. This allows him to see which blog entries are in response to which others.

Some of the blog entries appear emptier than others. He clicks on one of them and finds it is a blog article from the initial author. He clicks on the RSS link on the page and subscribes to the entire blog. When he goes back to the relationship diagram, the missing nodes are filled in. (This process might be automated in the future.) See Fig. 2.

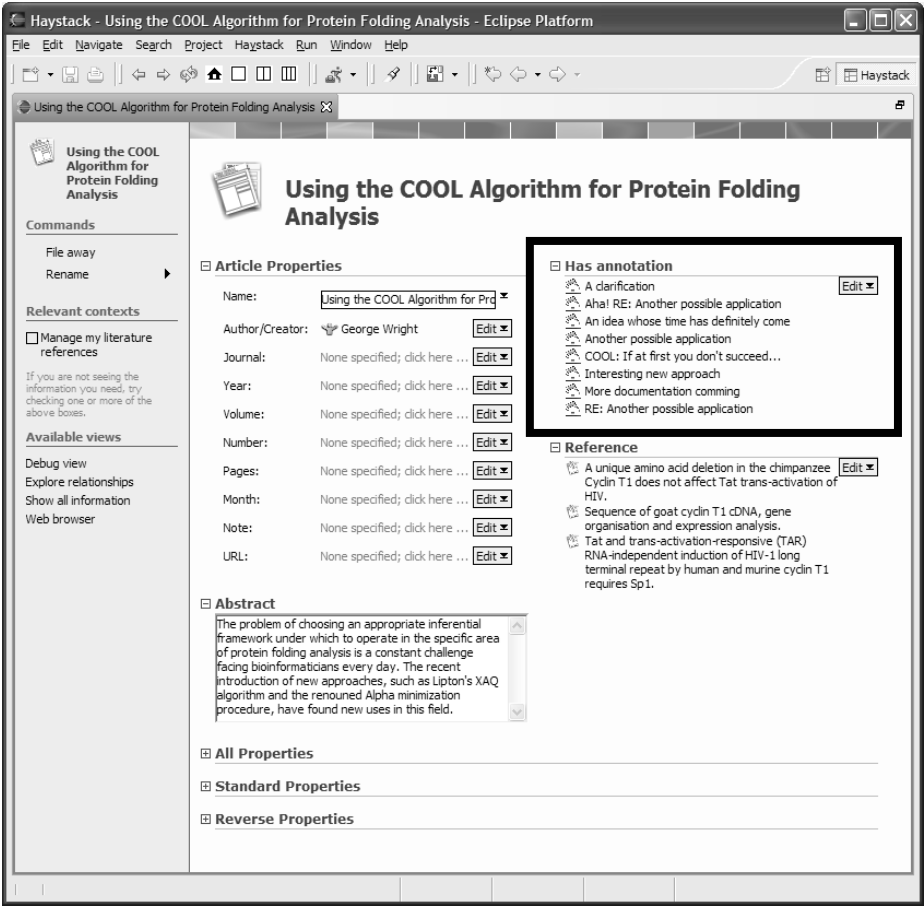


Fig. 1. Screen integrating information about a conference paper and annotations from semantic blogs (boxed area).

Furthermore, he can tell at a glance that two of the initial blog entries are commendations while the other is a criticism; however, looking down the line of conversation, he can see that the critic was eventually convinced and issued a commendation. Seeing all three of his friends in agreement is unique, and John decides to invest some time in reading through the articles. He creates a bookmark collection on the right hand docking pane and drags and drops some of the more interesting blog entries into it.

John decides to delve deeper into the approach. Scanning through the references in the paper, he finds a reference to the author's ontology. He browses to it in Haystack and clicks on one of the terms labeled "Analysis Engine". Just looking at the properties it supports does not give John a clear enough picture. However, the author has written a blog entry about it; by clicking on it, John is taken to the entire blog. A quick glance through the blog reveals that the author had been bombarded with requests for more explanation about his ontology, and he responded by blogging about

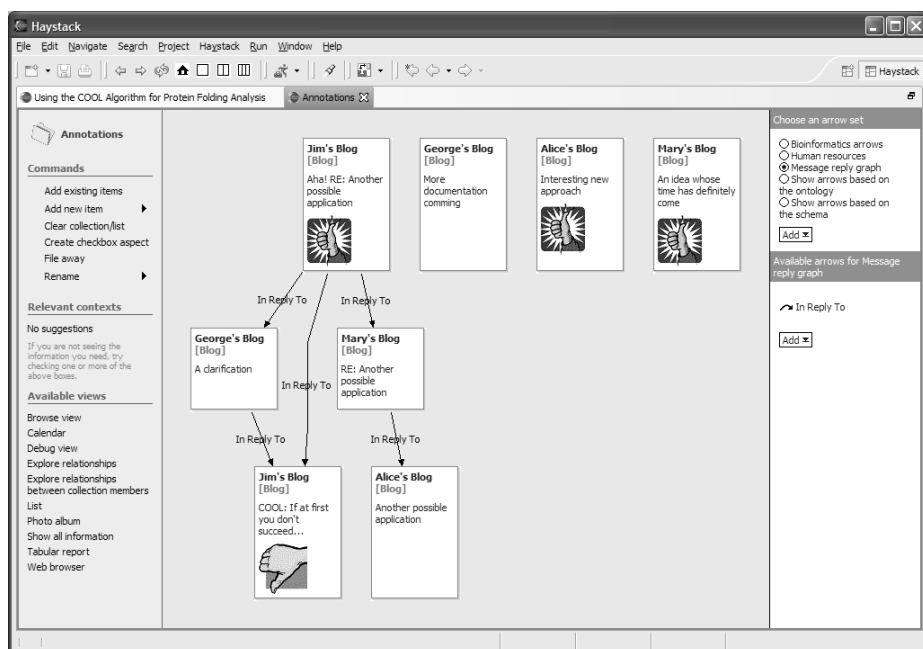


Fig. 2. Messages from multiple blogs displayed together as a reply graph.

various parts of the ontology over time, resulting in what is in essence a tutorial of how to use the ontology. John then browses through the blog, clicking on links to ontology elements embedded within the blog from time to time. In the end, John is himself convinced of the benefits of the approach and makes an entry in his own blog in concurrence, commenting on its applicability to his own research. In particular, he points out the one important argument that was critical to convincing him by dragging that blog entry from the bookmark collection he created earlier and dropping it in the In regards to field, making his blog entry an annotation for that argument.

7 Discussion

In the above scenario, we presented several examples of potential benefits for the integration of blogging and the Semantic Web. From the Semantic Web's perspective, blogging can be seen as a user-friendly metaphor for encouraging semantic annotation. Blogs already provide reviews, commentary, and human-readable instructions for many domains, and when directed towards Semantic Web resources, such blogs may be useful for documenting how to best make use of Semantic Web ontologies, schemas, and services. The contents of semantic blogs' annotations may also contain machine-readable metadata, such as numerical ratings, that could be consumed by Semantic Web agents to automatically determine aggregate ratings or other summary information.

In the future, there may be blogs that end up being completely semantically encoded. For example, one can imagine a semantic blog that notifies people of seminars, meetings, or other events run by an activities coordinator. Similarly, blogs that provide reviews (of movies, products, etc.) or that record statistics (e.g., scores from sports games) may someday be primarily encoded in RDF. Haystack enables users to input and publish such information using our extended version of RSS 1.0. Already, one sees evidence of desire for such support in sites such as Ripe Tomatoes [27].

From the blogger's perspective, certain semantics can be more cleanly recorded. The reply-to chains, which are usually embedded in prose, can be more explicitly specified as metadata, and bloggers would benefit from the same discussion visualization approaches that e-mail users have had for years. Also, our approach to displaying message threads does not rely on blog servers keeping in sync with each other via protocols such as TrackBack [18]; because the client creates the visualization, semantic blogs do not need to keep a record of what replies have been made to them.

Categories can be more easily shared not only between specific blogs, as could be the case if multiple bloggers worked in the same domain, but also among blogs in general. Simple category or type indications, such as commendation or complaint, are examples of more universally-shared categories that can be used by blog readers. Such categories could be exposed as visual cues (e.g., a thumbs up or thumbs down icon), which would improve the user experience during a quick visual scan through a set of blogs.

Also, the semantic annotation of blogs can be used to improve the user experience when dealing with various forms of information. A system such as Haystack can take advantage of these annotations in order to help users make sense of their information. For example, Haystack can help not only in bringing together the content from different blogs but also in integrating blog content with other information sources, such as conference programs. In this sense, blogs are acting as editorial "glue" that helps the reader to make sense of large bodies of information. Related scenarios can be made involving e-commerce product catalogs, online taxonomies, travel accommodation directories, etc.

Furthermore, Haystack allows blogs to be viewed in different ways, depending on the task at hand. Standard blogging tools allow users to view the articles in blogs and to group articles together in various ways. Less explored are the benefits of making use of the relationships between blogs. The example we cited earlier is Haystack's Explore relationships between collection members view, which allows the user to see the flow of a conversation even if it spans multiple blogs. One major consequence is that a group of people that wants to publish a conversation needs not do so through a single blog or a centralized mechanism, such as a newsgroup. Because URIs are used to name the blog entries, the various blogs can be aggregated by the blog reader and displayed together, while individual users are allowed to maintain the identities of and control over their own blogs—one of the key social motivational factors underlying blogging.

Acknowledgements. The authors would like to thank Kushal Dave for his helpful comments on the state of the art in blogging and Martin Wattenberg for useful pointers to relevant blogging literature.

Prefixes Used in This Paper

hs: <http://haystack.lcs.mit.edu/schemata/haystack#>
ann: <http://haystack.lcs.mit.edu/schemata/annotation#>
msg: <http://haystack.lcs.mit.edu/schemata/mail#>
rss: <http://purl.org/rss/1.0/>

References

1. Quan, D. and Karger, D. How to Make a Semantic Web Browser. Proceedings of WWW 2004.
2. Karger, D. and Quan, D. Collections: Flexible, Essential Tools for Information Management. Proceedings of CHI 2004.
3. Quan, D., Bakshi, K., and Karger, D. A Unified Abstraction for Messaging on the Semantic Web. Proceedings of WWW 2003.
4. Karger, D., Katz, B., Lin, J., and Quan, D. Sticky Notes for the Semantic Web. Proceedings of IUI 2003.
5. Kumar, R., Novak, J., Raghavan, P., and Tomkins, A. On the Bursty Evolution of Blogspace. Proceedings of WWW 2003.
6. Kahan, J. and Koivunen, M. Annotea: an open RDF infrastructure for shared web annotations. Proceedings of WWW10.
7. Berners-Lee, T., Hendler, J., and Lassila, O. The Semantic Web. *Scientific American*, May 2001.
8. Quan, D., Huynh, D., and Karger, D. Haystack: A Platform for Authoring End User Semantic Web Applications. Proceedings of ISWC 2003.
9. Blogger. <http://www.blogger.com/>.
10. Radio Userland. <http://radio.userland.com/>.
11. Quan, D., Bakshi, K., Huynh, D., and Karger, D. User Interfaces for Supporting Multiple Categorization. Proceedings of INTERACT 2003.
12. RDF Site Summary 1.0 Specification. <http://web.resource.org/rss/1.0/spec>.
13. RSS 2.0 Specification. <http://blogs.law.harvard.edu/tech/rss>.
14. RSS 0.92 Specification. <http://backend.userland.com/rss092>.
15. Quan, D., Karger, D., and Huynh, D. RDF Authoring Environments for End Users. Proceedings of the International Workshop on Semantic Web Foundations and Application Technologies 2003.
16. Noy, N., Sintek, M., Decker, S., Crubezy, M., Ferguson, R., and Musen, M. Creating Semantic Web Contents with Protege-2000. *IEEE Intelligent Systems* 16 (2), 2001, pp. 60-71.
17. The Atom Syndication Format 0.3. <http://www.atomenabled.org/developers/syndication/atom-format-spec.php>.
18. TrackBack Technical Specification. <http://www.movabletype.org/docs/mttrackback.html>.
19. Google Toolbar. <http://toolbar.google.com/>.
20. NetNewsWire. <http://ranchero.com/netnewswire/>.
21. SharpReader. <http://www.sharpreader.net/>.
22. Technorati. <http://www.technorati.com/>.
23. Review (RVW) Module for RSS 2.0. <http://www.pmbrowser.info/rvw/0.1/>.
24. Nardi, B., Schiano, D., Gumbrecht, M., and Swartz, L. "I'm Blogging This": A Closer Look at Why People Blog. Submitted to *Communications of the ACM*. <http://www.ics.uci.edu/%7Ejpd/classes/ics234cw04/nardi.pdf>.

25. Paquet, S. Towards Structured Blogging. <http://radio.weblogs.com/0110772/stories/2003/03/13/towardsStructuredBlogging.html>.
26. blogosphere.us. <http://www.blogosphere.us/trends.php>.
27. Rotten Tomatoes. <http://www.rottentomatoes.com/vine/register.php>.
28. Hotmail. <http://www.hotmail.com/>.
29. Moveable Type. <http://www.movabletype.org/>.
30. Blogger API. <http://www.blogger.com/developers/api/>.