Semantic Browsing

Alexander Faaborg and Carl Lagoze

Information Science, Cornell University, Ithaca, NY 14850 USA

Abstract. We have created software applications that allow users to both author and use Semantic Web metadata. To create and use a layer of semantic content on top of the existing Web, we have (1) implemented a user interface that expedites the task of attributing metadata to resources on the Web, and (2) augmented a Web browser to leverage this semantic metadata to provide relevant information and tasks to the user. This project provides a framework for annotating and reorganizing existing files, pages, and sites on the Web that is similar to Vannevar Bush’s original concepts of trail blazing and associative indexing.

1 Introduction

The organization of information has been an ongoing challenge for libraries. As libraries make increasing use of Web resources to populate their collections (exemplified by the work by our colleagues in the National Science Digital Library (NSDL) [1]), digital library developers need to find effective ways to organize digital objects stored across distributed servers on the Web. Just as brick and mortar libraries built shelves for books, digital libraries need virtual shelves to organize the elements of their collections.

The Semantic Web effort [2] of the W3C and its collaborators is developing the tools that facilitate the organization and classification of Web resources in a manner usable to both humans and machines. A result of this effort is the Resource Description Framework (RDF) [3] that provides the syntactic and semantic building blocks for organizing Web resources. While much effort has gone into RDF and related Semantic Web standards [4], the actual user interface of the Semantic Web is rarely mentioned. We directly address this issue.

We have implemented two tools that allow users to both build and use the Semantic Web. One is a user interface for quickly generating RDF annotations that overlay a personal and sharable semantic layer on top of the underlying Web. The other is a modified Web browser that leverages this semantic layer as a means of enriching and personalizing the browsing experience, locating related information, and providing contextually relevant tasks. We believe that end-user Semantic Web tools will greatly improve the usability of the Web in future digital libraries. They will enable users to organize their individual browsing experiences and help information intermediaries, such as librarians, tailor the Web experience for their target audiences. They will help integrate the Web into digital libraries rather than make digital libraries simply catalogs of Web resources.
Our work, and that of the broader Semantic Web community, helps move the Web closer to Vannevar Bush's seminal vision [5]. In 1945 Bush wrote the well-known article *As We May Think*, in which he describes the concept of blazing trails between documents. Many people consider his concept of trail blazing to be the precursor of the hypertext we find today on the World Wide Web. However, the Web in its present form falls short of Vannevar Bush's vision for a number of reasons.

First, the Web allows for only primitive trail blazing: hyperlinks under the control of the page author that have limited semantic expressiveness. Our tools allow users to overlay their own trails on top of the existing hyperlink structure. Furthermore, the expressiveness of RDF allows the user to endow these trails with rich semantic meaning.

Second, the current Web fails to realize Bush's visions of automatic processing and logic. Bush saw this logic processing as the means towards advanced capabilities such as associative indexing, in which documents immediately and automatically select other related documents. The machine-readable and semantically rich annotations created via our tools provide the foundation for agent-initiated associative indexing and other automated tasks on metadata.

It is our hope that by demonstrating semantic browsing from the perspective of end users, we will be able to establish the importance of the Semantic Web in the implementation of Web based digital libraries.

2 Trail Blazing Using Semantic Web Technologies

Vannevar Bush describes the concept of trail blazing as joining two documents, and "thereafter, at any time, when one of these items is in view, the other can be instantly recalled." [5] While hyperlinks provide a mechanism for joining documents on the Web, these links can only be placed on a Web page by its author, and not its readers. Our work is based on the notion that the utility of the Web as an information resource can be enriched if users themselves have the power to link together documents and otherwise annotate the Web.

One potential application of our work is in the education domain, where our NSDL project [1] has demonstrated teachers' need to weave together primary Web resources in order to make them more usable for the classroom experience [6]. For the remainder of this paper we will use an example from this domain, an imaginary high school biology teacher named Mrs. Thompson. It is important to note that our work is applicable in a variety of domains from information management inside a corporation to organizing the public Web.

Let us imagine that Mrs. Thompson, in preparation for her class, has located a Web site about the tundra biome and a Web site about the coniferous woodland biome, and she wants to guide her students to these sites. In the current Web, Mrs. Thompson might create a new document that comments on these pertinent resources intermixed with hyperlinks to them. This has a number of limitations. The first is that Mrs. Thompson's links only hook into the Web one level deep. In order to follow Mrs. Thompson's trail, her students must constantly return to
her Web page and follow the next link. Related pages are not joined as in Bush’s vision, but are simply referenced. It is not the case that “when one of these items is in view, the other can be instantly recalled.” [5] The second limitation is that Mrs. Thompson’s new Web page uses natural language to express the relationship among her chosen resources. The imprecise semantics of natural language limit the utility of her annotations to human browsing rather than machine understanding. By simply creating a new Web page, Mrs. Thompson’s attempt at trail blazing does not help to organize the Web as much as it just makes it a little larger.

The challenge that Mrs. Thompson faces in making the Web useful in her teaching is to provide some level of structure and organization to her students and exploit the educational value of hypertext [7]. As described above, the current Web limits her ability to structure the Web for her students to simple pointers that then leave her students at the mercy of the Web’s existing link structure, which may not be relevant to her educational goals. The standards of the Semantic Web, RDF and RDF schema, provide the building blocks for a more structured, organized, and semantically rich Web. The remainder of this section describes the utility of these Semantic Web standards in a series of progressively richer examples. In the subsequent section, we describe how our work applies these standards to enhance the Web experience.

The simplest application of the Semantic Web allows Mrs. Thompson to associate metadata with a Web page that she does not control, as illustrated in the figure below. This metadata might then be integrated into her students’ Web browsing experience in a manner similar to the way Annotea [8] clients show user annotations. For instance, when browsing to a Web site about biomes in the United States Rocky Mountains, her students’ Web browsers could note the pages Mrs. Thompson feels deserve a close reading as they came to them, while still allowing the students to browse freely.

The Web consists not only of pages, but the relationships between pages that are established by its navigational structure. While the meaning of these links is sometimes described by their anchor text, Mrs. Thompson may want to annotate links with information appropriate to her lesson plan and class needs, as illustrated below. For example, Mrs. Thompson would like the next page her students view after learning about the tundra to be the Web page about the coniferous woodland, because they may visit the biomes in this order on their field trip. Mrs. Thompson could annotate the hyperlink, and this information would then appear when her students viewed the tundra Web page.
The established links on the Web may not be sufficient for Mrs. Thompson’s lesson goals. For example, although the links on a Web site show that the pika, a small member of the rabbit family, can be found in the tundra, she may want to create and annotate an additional link to show that this animal can also be found in the coniferous woodland. Students browsing the Web site then would be able to follow Mrs. Thompson’s imposed link (shown below) as well as the established links on the Web site.

The utility of the Semantic Web comes from the fact that annotations (metadata) can be machine readable and interpretable. Concepts and taxonomies can be expressed through RDF-based ontology languages [4]. Metadata creators may then refer to the URIs of these taxonomies within their descriptions rather than using text strings. The use of these unique identifiers as references allows one of the most important types of trail blazing, the ability to automatically group disparate resources together based on their shared conceptual relationships. Vannevar Bush referred to this feature as associative indexing [5].

Mrs. Thompson may want to create her Web annotations using a subject taxonomy endorsed by the National Science Teachers Association (NSTA). Thus, instead of using strings like ‘tundra’ and ‘pika’ in her descriptions, she could use the respective URIs from the NSTA taxonomy. As we show later, this allows her students to navigate Web sites based on semantic concepts in addition to the sites’ hyperlinks. When two Web pages are annotated with the same URI, associative indexing places both Web pages on the same metaphorical shelf, as illustrated below.

Because the metadata used in all of the above annotations is expressed in a machine-readable format, it can be shared between parties and merged together. For instance, because Mrs. Thompson can use URIs to associate two Web sites about animal life in the Rocky Mountains, these pages can be automatically associated with a larger set of pages also about animal life in the Rocky Mountains that have been annotated with the same URIs by other individuals. This facilitates the information environment envisioned by the NSDL where users become contributors in a dynamic collaborative environment.
In the next section we will demonstrate how these types of semantic annotations can aid in locating relevant information and improve the functionality of a Web browser, alleviating some of the frustrations users currently have with the Web. First however, we will see how a user like Mrs. Thompson can create these types of annotations with our software.

3 Tools to Build and Browse the Semantic Web

We have developed two applications that allow users like Mrs. Thompson to leverage the Semantic Web. Site Annotator is used to author semantic metadata about Web resources, and the Web Task Pane modifies Internet Explorer to integrate this semantic metadata into the browsing experience. Larger screen shots and additional information about these applications can be found at: http://www.cs.cornell.edu/lagoze/Projects/SemanticBrowsing/

3.1 Site Annotator

Site Annotator, shown above, has an address bar similar to a Web browser. When a user enters a URL into this address bar, the Web site is shown in a tree view and list view, as if the user is viewing the file system of their computer. We chose the file system metaphor because it provides an easy way to select multiple Web resources at the same time. Site Annotator is able to display a file system view of a Web site by parsing each Web page’s HTML. The file system
view of a Web site is accessed through HTTP in Site Annotator and should not be confused with the Web site's actual file structure, which might be accessed through FTP. Thus, if a file exists on a Web server but none of the other files link to it, Site Annotator will not be able to find it.

The purpose of Site Annotator is to help the user author subject-predicate-object statements about Web resources. The user interface for creating a subject-predicate-object statement appears as three blocks arranged in a triangle (the foreground window in the figure above). The subject and object blocks are on the bottom of the triangle and the predicate block connects them. Web resources selected in the main window appear in the statement’s subject block, which is dark green, and the statement’s predicate and object blocks appear in white because they require the user’s input. After the user enters information into the required predicate and object blocks and creates the statement, all three blocks turn green and the window closes. These statements can later be exported as RDF triples encoded in XML. Because of the importance of using URIs from Semantic Web ontologies when creating annotations, the application allows the user to import multiple RDF schemas through the Load RDF Schema button and reference them via URIs in their annotations.

Returning to our example of Mrs. Thompson, one of the ways she wanted to annotate the Web included leaving notes for her students to find as they browsed to particular pages. This can be accomplished by selecting the appropriate page as the subject block, selecting “has note” in the drop down list box in the predicate block, and then typing the note she wishes to leave into the object block. The real power of using machine-readable metadata however lies not in notes expressed in a natural language, but in using ontologies to group and express relationships between Web resources while trail blazing. Mrs. Thompson can create a statement about a Web page expressing that the pika can be found in the tundra. To be useful, the predicate and object blocks (found in and tundra) in this statement need to be expressed as URIs that come from a Semantic Web ontology. Many organizations are working to create ontologies for the Semantic Web [9]. If Mrs. Thompson would like to group several animals into the tundra biome, she simply needs to keep the Annotate Resource window open, and each Web resource she selects will be added to the subject block. However, statements that animals can be found in the tundra do not need to be made at the same time, or even by the same person. They simply need to use the same URIs to express found in and tundra. These URIs place the animal pages on the same metaphorical shelf. Later we will see how grouping Web resources with URIs allows Mrs. Thompson’s students to navigate based on semantic concepts.

The Create Semantic Link window is similar to the Annotate Resource window, but both the subject and object files must be selected in the main window. The
Create Semantic Link window allows Mrs. Thompson to create semantic associations between existing Web resources. This window resembles the basic user interface designed by Vannevar Bush when he described the concept of trail blazing: “Before him are the two items to be joined, projected onto adjacent viewing positions... The user taps a single key, and the items are permanently joined.” [5] Mrs. Thompson could use the Create Semantic Link window to indicate to her students which Web page she wants them to view next, to annotate an existing hyperlink with additional information, or to create an entirely new hyperlink between Web resources. For instance, she could instruct her students that more information about the pika can be found on another Web page.

The previous example demonstrates human intervention to create an RDF statement. There are cases however, when Site Annotator has enough knowledge to automatically generate metadata. We have been working on making Site Annotator generate RDF statements in the background while the user focuses on complex annotations like biomes. For instance, Site Annotator will automatically author semantic metadata, expressed as RDF triples, about page titles, hyperlinks, link text, and MIME types. This automatic metadata generation is only a step toward our long-term goal of identifying more complex types of relationships on the Web. Two examples include automatically detecting implicit structures in a Web site, like an article that spans multiple pages, and automatically detecting groups of files, like the structure of an image gallery.

After Site Annotator is finished generating metadata in the background and the user is finished trail blazing, the user can export the generated metadata to an RDF file by clicking the Save RDF button.

This semantic metadata can be published on the Web in a variety of ways. If the user has control of the Web site, the metadata can be placed on the server in an index.rdf file. This is functionally the opposite of a robots.txt file in that it gives parsing agents as much information as it can about the site, rather than instructing the agents to stay away. Other options include placing the semantic metadata inline with the HTML. Future changes to HTTP might provide access to the RDF metadata as part of a page request. We currently only support placing the semantic metadata in an index.rdf file, which is the simplest of these solutions.

If the user does not control the Web site, future options for sharing their metadata may include placing it in a directory service like dmoz.org, or anywhere online where a service like Google will be able to locate it. Of course, any public facility for annotating Web resources may lead to misuse and spam, so the creation of a global Semantic Web will also require a Web of Trust [10]. The Semantic Web may first emerge in controlled communities like corporate intranets and digital libraries where the reliability of metadata can be guaranteed.
Users will upload RDF triples into a database or place them online to be found by a spider agent and the collection of these statements will literally form the Semantic Web. Until a Web of Trust is constructed, the public Web may only contain small discrete packets of reliable semantic metadata. However, sharing knowledge in this manner is still compelling. Imagine if an expert like Stephen Hawking could publish his personal organization of physics on the Web in an .rdf file for anyone to download.

At present, the .rdf files created by Site Annotator may be shared via email and over the Web, and then loaded in the Web Task Pane as described below. In the case of Mrs. Thompson, her students’ Web browsers could be automatically set to look for semantic metadata created by her.

### 3.2 The Web Task Pane

The Web Task Pane, shown above, is a tool for semantic browsing, implemented as an explorer bar for Microsoft Internet Explorer. The user interface for the Web Task Pane is similar to the task panes found in Microsoft Windows XP and Office XP, which dynamically react to what the user is viewing. As the user browses the Web, the Web Task Pane will continually download and manipulate semantic metadata in the background, displaying information, tasks, and navigational controls in the user interface.

We believe that the ability to navigate based on semantic concepts can improve the overall usability of the Web. Vannevar Bush’s vision noted that the human mind “operates by association. With one item in its grasp, it snaps instantly to the next that is suggested by the association of thoughts, in accordance with some intricate web of trails carried by the cells of the brain.” [5] In *Hypertext and Cognition*, Jean-François Rouet discussed the human computer interaction involved with Web browsing, saying, “Progression in a hypertext requires active
decision making on the part of the reader. In order to make coherent selections, the reader has to possess a mental representation of how the information is organized." [7] Usability on the Web breaks down when a user identifies a relationship in their mental model, but then cannot locate an appropriate hyperlink.

The first section of the Web Task Pane, Page Properties, allows the user to navigate the Semantic Web. In the above screen shot, the Web Task Pane has downloaded RDF metadata from the Semantic Web (which was created with a tool like Site Annotator) and analyzed it to learn that the pika lives in the tundra biome. It then performed associative indexing to locate other animals that also live in the tundra. Page Properties allows the user to navigate between Web resources based on semantic concepts. Allowing users to navigate on semantic concepts increases the overall density of the Web. To prevent the user from being overwhelmed with information, Page Properties contains a tree interface that the user must actively expand.

The above screen shot shows the result of associative indexing using semantic metadata, displaying links to a set of pages that are metaphorically shelved using the same predicate and subject URIs: $x$ is *found in the tundra*. Other types of metadata, however, particularly those relating to a Web site's structure and files, can be used by the Web Task Pane to provide the user with a more advanced Web browser.

The second section of the Web Task Pane is called Site Tasks. The tasks that appear here are similar to tasks available in a normal Web browser like downloading or printing. Unlike current Web browsers however, these tasks go beyond the Web page the user is currently on. Often a Web browser's lack of understanding results in the user performing repetitive actions like a machine. Examples include printing each page of a multi-page article individually, or right clicking on each image in a gallery and selecting “Save As.” The Web is replete with similar structures, and users perform a similar set of tasks. Because of this, the Web Task Pane is able to use semantic metadata to provide its users with contextually relevant tasks. Currently the Web Task Pane has the ability to save a sequence of pages, and download all of the images in a set. We are now in the process of expanding the number of tasks it can automate for the user based on semantic metadata.

The third section of the Web Task Pane is called Site Navigation. Here the Web Task Pane uses the same semantic metadata as the Site Tasks section, but for the purpose of providing the user with more advanced navigational controls. Providing a unified navigational interface increases Web usability because the user does not have to visually search for hyperlinks to navigate common structures on the Web. User advocate Jakob Nielsen believes that “at a minimum,
Web browsers need to have better support for structural navigation.” [11] The controls in the Site Navigation section include the ability to go up one level in a hierarchical structure, to view the Web site’s site map, and next and previous controls for quickly navigating between resources that form a sequence.

4 Implementation Details

Site Annotator is a standalone Java application. Each subject-predicate-object statement the user specifies using the Annotate Resource or Create Semantic Link windows forms an RDF triple. Site Annotator integrates with the RDF visualization tool IsaViz, which was created by Xerox PARC [12], so that at any time the user can press a button and view the semantic metadata they have created graphically. To encode the statements the user creates into RDF metadata serialized in XML, Site Annotator uses the Jena API that was created by HP Labs [13].

The Web Task Pane was written in C# for the Microsoft .NET framework. It integrates with Microsoft Internet Explorer using COM interopability. When the user loads a Web page, Internet Explorer notifies the Web Task Pane of the new URL. The Web Task Pane then attempts to locate an index.rdf file on the server containing semantic metadata. It also allows for the user to manually load .rdf files. The downloaded RDF metadata, which is serialized in XML, is manipulated using the .NET framework’s implementation of DOM.

5 Related Work

The Mrs. Thompson example used throughout this paper is only hypothetical. However, Shipman et al. have researched the use of Vannevar Bush style trail blazing in actual classroom environments [14]. The Web Task Pane allows users to navigate based on semantic concepts, or virtual shelves. Carr et al. have also studied conceptual linking [15]. The Web Task Pane also provides users with richer navigation options. Preliminary work on the use of RDF for site maps has been conducted by Dan Brickley [16].

Our software applications are related to the suite of RDF tools that have been developed over the last few years. The best reference to these tools is Dave Beckett’s Resource Description Framework (RDF) Resource Guide [17]. There are two RDF authoring tools in that list that are most closely related to our work: the RDF Instance Creator [18] and the IsaViz Visual Authoring Tool for RDF [12]. Site Annotator uses IsaViz for the display of RDF models.

Other Semantic Web Annotation tools include the OntoMat-Annotizer [19], the SHOE Knowledge Annotator [20], SMORE [21], MnM [22], and Trellis [23]. Semantic Web tools that extend Mozilla include the Mozilla sidebar [24], which is a client for Annota [8], and the COHSE Annotator sidebar [25]. Semantic Web tools that extend Internet Explorer include our work, and the work of the Annotation System project group of the National Centre for Software Technology in Bangalore [26]. The NCST’s Annotate Browser explorer bar is similar
6 Conclusion

The Semantic Web is a new enough research area that most people, upon hearing the phrase “building the Semantic Web,” assume the word “building” refers to its infrastructure, its syntax and protocols. We are interested in creating tools that allow end users to literally build the Semantic Web, to give it semantics. Users like Mrs. Thompson do not know about URIs, ontologies, RDF, or XML. If the Semantic Web is to become a reality, building it is going to need to be more like trail blazing with colorful blocks, and less like using XML Spy. Site Annotator, while providing a graphical interface for trail blazing with Semantic Web technologies, may still not be high-level enough for users like Mrs. Thompson. To facilitate a collaborative Semantic Web, users will need Semantic Web enabled annotation tools built into their browsers, and authors will need Semantic Web enabled tools built into their Web development software. With Semantic Web research quickly progressing, the next step in creating these applications will not be in developing their infrastructure, but in designing their user interface.

Acknowledgments

We would like to acknowledge the contributions of Geri Gay and Michael Spivey, who are on Alexander Faaborg’s honors thesis committee. This work was supported in part by NSF Grant 9905955.

References