Sticky Notes for the Semantic Web

David R. Karger
MIT Laboratory for Computer Science
200 Technology Square
Cambrdge, MA 02139 USA
karger@theory.lcs.mit.edu

Boris Katz, Jimmy Lin, Dennis Quan
MIT Artificial Intelligence Laboratory
200 Technology Square
Cambrdge, MA 02139 USA
{boris,jimmylin,dquan}@ai.mit.edu

ABSTRACT
Computer-based annotation is increasing in popularity as a mechanism for revising documents and sharing comments over the Internet. One reason behind this surge is that viewpoints, summaries, and notes written by others are often helpful to readers. In particular, these types of annotations can help users locate or recall relevant documents. We believe that this model can be applied to the problem of retrieval on the Semantic Web. In this paper, we propose a generalized annotation environment that supports richer forms of description such as natural language. We discuss how RDF can be used to model annotations and the connections between annotations and the documents they describe. Furthermore, we explore the idea of a question answering interface that allows retrieval based both on the text of the annotations and the annotations' associated metadata. Finally, we speculate on how these features could be pervasively integrated into an information management environment, making Semantic Web annotation a first class player in terms of document management and retrieval.

Categories and Subject Descriptors
H.3.5 [**Information Storage and Retrieval**]: Online Information Services – *Web-based services*

General Terms
Human Factors

Keywords
Annotation, Semantic Web, RDF, user interface, question answering, natural language

1. INTRODUCTION
People make notes to themselves in order to preserve ideas that arise during a variety of activities, e.g., reading documents or attending meetings. The purpose of these notes is often to summarize, criticize, or emphasize specific phrases or events, and notes can also serve as reminders that are used later to improve recollection. Passing annotated documents between colleagues is a highly effective way to exchange ideas and to engage in collaboration. However, as any student who has shared his or her notes with a classmate knows, sharing notations on paper requires considerable effort: either the document must be photocopied, or the author must give up the original for a period of time.

Keeping documents in electronic form alleviates many of these problems. In fact, most work to date on Internet annotation has concentrated on the types of document- or topic-specific applications mentioned above. For example, Microsoft Research has studied the use of Microsoft's Office 2000 product for posting documents to the Web and engaging in online discussions [3]. We believe that these kinds of software packages go a long way towards supporting the exchange of ideas online, but there are still many areas of functionality in need of further improvement.

One powerful use of annotations is locating items that have been subjectively found by others to match certain criteria. For example, when buying a product on an e-commerce website, one can often find customer commendations or complaints on the webpage featuring the product. What if the specific product being sought has not yet been identified? One major advantage of going to a bookstore over buying online is the ability to speak with a sales associate and ask in person, "Can you suggest a mystery novel for my long flight to Tokyo?" Most websites are unable to offer such advice, not because they do not expose customer feedback, but because their interfaces are geared towards allowing users to *browse* annotations for their own sake, rather than using them to *find* objects given a specification.

2. APPROACH
Annotation functionality is not limited in scope to e-commerce; rather, the idea of using annotations to retrieve objects can be applied to all aspects of information management. Systemwide infrastructure is needed to support pervasive annotation and its use as a retrieval paradigm. In effect, we wish to create the digital equivalent of physical "sticky notes": the ability to attach annotations anywhere and to anything. As a motivating example of the deficiencies of current information management tools, consider where (if anywhere) one could record the comment "this report will be useful the next time I meet with a customer from Denmark."

To this end, we are developing the idea of annotation-based information retrieval within the context of the Haystack project [5]. The goal of this project is to develop a tool that allows users to easily manage their documents, e-mail messages, appointments, tasks, and other information. Haystack uses a semistructured data model to describe the connections between different documents in a user's corpus as well as the metadata concerning each document. Furthermore, Haystack's user interface exposes general tools for navigating the various kinds of information found in the user's corpus. As a result, integrating annotation functionality can

Copyright is held by the author/owner(s).
IUI'03, January 12–15, 2003, Miami, Florida, USA.
ACM 1-58113-586-6/03/0001.

be done uniformly throughout the environment for all types of documents.

The idea of using natural language annotations for information access was pioneered by the START system [7] [9]. Natural language descriptions of information segments can be analyzed by a natural language understanding system to provide question answering capabilities. The natural language annotation technology developed for START can not only be employed to describe textual segments but also multimedia content, database queries, and even arbitrary code fragments.

We seek to build on START's experience and develop ubiquitous annotation support into Haystack. Our annotation framework builds upon the Semantic Web, an extension of the World Wide Web that facilitates the exchange of machine-readable information [1]. At the heart of the Semantic Web is a technology known as the Resource Description Framework (RDF) [10], a portable XML-based representation of semantic networks or labeled directed graphs. RDF serves as the *lingua franca* of the Semantic Web, making it possible for programs to exchange ontologically encoded information, such as authorship, annotations, topic labels, content and customer satisfaction ratings, etc. over the Internet using a standard format. RDF also forms the basis of Haystack's data model, meaning that annotations created from within the Haystack environment are usable by other RDF-enabled software packages.

In fact, annotation support has already been explored in the context of the Semantic Web. Projects such as Annotea [6] and CREAM [4] are developing frameworks for creating and exchanging RDF-encoded annotations between Semantic Web clients. However, we believe that using annotations for information access and providing natural language support for these annotations are crucial elements missing from previous work. Furthermore, natural language technology enables users to query information stores using everyday language without resorting to specialized and often unintuitive query languages.

Our approach can also be compared to the image search technology espoused by search engines such as Google. Here, pieces of anchor text on web pages (e.g., text within **<a>** tags or **alt** attributes of **<img>** tags) serve as annotations for images. In such a paradigm, image annotations are, in essence, byproducts of webpage creation. We instead posit that annotations should be treated as first class objects, and user interfaces for creating these annotations must not treat annotation as a secondary, special-purpose activity as is the case when assigning an **alt** attribute to an image.

We believe that using annotations as a retrieval paradigm represents a new approach to creating, browsing, and accessing information on the Semantic Web. The vision presented here is as follows: Users can use a client such as Haystack to retrieve and view information in RDF about documents, multimedia, or products. They can then compose comments, descriptions, and criticisms in the form of annotations attached to arbitrary objects. These annotations can be sent to a shared Semantic Web server, and interested clients can query these servers, with either RDF-based metadata queries, natural language queries, or a combination thereof, to find information that suits users' needs.

To realize this paradigm, we are working to integrate the START natural language engine with Haystack's information store. The remainder of this paper focuses on a data model for allowing Haystack and START to interoperate and a user interface that prominently and pervasively exposes annotation functionality. Together our data model and user interface work to make natural language search an integral part of the user experience.

3. DATA MODEL

The basic RDF data model consists of a series of nodes in a graph, which represent objects, and arcs connecting nodes, which represent relationships between objects. An arc (also called a predicate) in conjunction with the two nodes it connects is collectively termed a statement in RDF parlance and is the unit of information in the RDF model. Furthermore, nodes and predicates are named by uniform resource identifiers (URIs), which in conjunction with the XML Namespace standard [2], allow object identifiers to be globally unique. These standards also allow predicate vocabularies to be defined, which give standard names to relationships such as "has name", "published by", etc. These predicates often correspond to concepts that can be easily expressed in natural language; in fact, START has been using a ternary expression representation of language for nearly two decades [8], which simplifies translation to and from RDF statements.

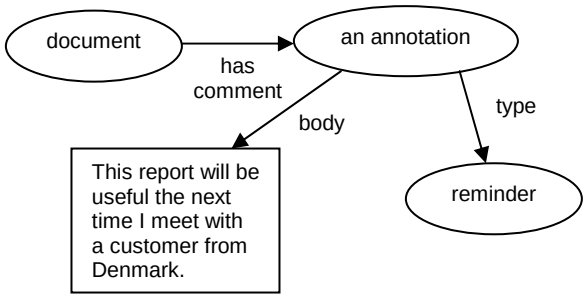

Figure 1: Example of annotation ontology

We have applied these fundamental RDF concepts to the problem of modeling annotations. Figure 1 depicts an example document and an associated annotation. We propose three core elements for an annotation:

1. The annotation predicate, which specifies the relationship between the annotated object and the annotation. Other possible predicates include "has description", which connects a document to a factual summary or synopsis, and "has reply", which connects a document (or perhaps another annotation) to an annotation that serves as a response to the original document.
2. The annotation body, which consists of natural language text. This text is analyzed by START and forms the basis of the system's natural language querying capabilities. Because the annotation text is parsed into syntactic structures, linguistically sophisticated machinery such as synonymy/hyponymy, ontologies, and structural transformation rules can all be brought to bear on the question answering process. Linguistic techniques allow our approach to achieve capabilities beyond simple keyword matching, e.g., handling complex syntactic alternations involving verb arguments.
3. The type of annotation. Possible types might include "reminder", "suggestion", "problem", "complaint", "idea", and "plan". Annotation types provide one way for people to classify their annotations and can aid in the browsing and retrieval of relevant objects.

Annotations can be saved into RDF stores, such as those used by Haystack for storing personal corpora, or the kind used by Annotea [6] for storing shared information. A natural language search engine would either require access to such an RDF store (in order to properly index and search it) or require that the natural language annotations be extracted and stored internally, as is the case with START.

4. USER INTERFACE

The expressiveness of our data model can only be realized if it is exposed to the user properly. The best way to provide ubiquitous support for annotations is to make the annotation process an intrinsic part of the user interface. Figure 2 shows a screenshot from Haystack depicting the annotation pane. Users can create description and comment annotations from this pane as well as view annotations from others. Annotations can also be created by right-clicking on any object on the screen. The interface supports the standard notion of threaded annotation discussions, built up from annotations that serve as replies to other annotations. Support for annotation type and other metadata associated with annotations has also been incorporated.

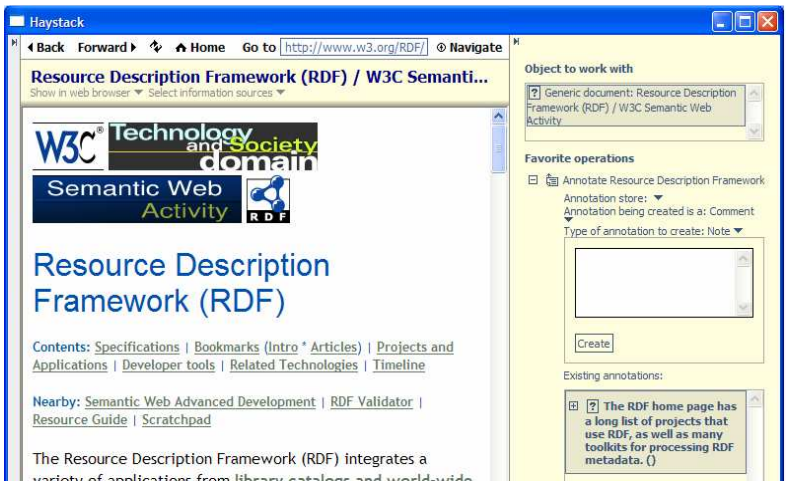

Figure 2: Screenshot of Haystack annotation pane

We have developed the infrastructure support needed for pervasive annotations and are currently developing support for querying. Queries may include elements from both the natural language text as well as other annotation metadata, such as the annotation type, annotation predicate, date of creation, annotator, etc. We envision the system being able to answer queries such as the following:

- What was it I wanted to remember about meeting Danish customers?
- Who complained about the service at this restraunt?
- Show me that idea John had about improving turnover.

These queries can further span multiple RDF stores and annotation servers across the Semantic Web, as a result of Haystack's support for integrating disparate information sources.

5. CONCLUSION

In this paper we have proposed that online annotation systems are useful not only as tools for collaboration, but also as effective means for retrieving documents and finding items of interest on the Semantic Web. We outlined an ontology for describing annotations in RDF and described our initial efforts to integrate support for creating and searching these annotations into Haystack. We believe that natural language support is a crucial element of any annotation framework. By leveraging the START natural language system, we can allow users to locate relevant information by simply specifying a query in everyday language. However, more work is warranted to demonstrate and test the usefulness of our systems to users.

6. ACKNOWLEDGMENTS

This work was supported in part by the MIT-NTT collaboration, the MIT Oxygen project, DARPA contract number F30602-00-1-0545 administered by the Air Force Research Laboratory, a Packard Foundation fellowship, and IBM.

7. REFERENCES

[1] Berners-Lee, T., Hendler, J., and Lassila, O. The Semantic Web. Scientific American (May 2001).

[2] Bray, T., Hollander, D., and Layman, A. (eds.) Namespaces in XML. Available at http://www.w3.org/TR/REC-xml-names/.

[3] Cadiz, J., Gupta, A., and Grudin, J. Using Web annotations for asynchronous collaboration around documents, in Proceedings of CSCW 2000 (December 2000).

[4] Handschuh, S., Staab, S., and Maedche, A. CREAM—creating relational metadata with a component-based ontology-driven annotation framework, in Proceedings of K-CAP 2001 (October 2001).

[5] Huynh, D., Karger, D., and Quan, D. Haystack: a platform for creating, organizing and visualizing information using RDF. Semantic Web Workshop, WWW2002 (May 2002). Available at http://haystack.lcs.mit.edu/papers/sww02.pdf.

[6] Kahan, J. and Koivunen, M. Annotea: an open RDF infrastructure for shared web annotations, in Proceedings of WWW10 (May 2001).

[7] Katz, B. Annotation the World Wide Web using natural language, in Proceedings of RIAO 1997 (June 1997).

[8] Katz, B. Using English for indexing and retrieving, in Proceedings of RIAO 1988 (March 1988).

[9] Katz, B., Lin, J., and Quan, D. Natural language annotations for the Semantic Web, in ODBASE 2002 Proceedings (October 2002).

[10] Lassila, O. and Swick, R. (eds.). Resource Description Framework (RDF) model and syntax specification. Available at http://www.w3.org/TR/1999/REC-rdf-syntax-19990222.