

Annotea Shared Bookmarks

Marja-Riitta Koivunen, Ralph Swick, and Eric Prud'hommeaux

World Wide Web Consortium

MIT Laboratory for Computer Science

marja@w3.org, swick@w3.org, eric@w3.org

ABSTRACT

Traditional bookmark systems provide inadequate support for Web users with variety of devices including smart phones and their browsers. In addition, they offer very little support for sharing the bookmarks and topics between groups of users working together.

Annotea annotations provide a means for users to share communication about Web documents by attaching external annotation metadata to the documents. Users can easily find annotations related to a document by using clients that present these annotations in the document context. Similarly, bookmarks can be thought of as kind of annotations that attach bookmark metadata to documents and help users find documents and informal categories or topics assigned to them.

Our Annotea metadata storage and retrieval platform built on Semantic Web technologies now also supports bookmark metadata server. Clients may share, query and present the bookmarks and topics.

Use of RDF metadata for bookmarks offers flexibility in connecting to other bookmark systems and to annotation functionality. The metadata is machine readable and can be converted from and to other bookmark formats. Metadata from several bookmark sources - either local or from selected bookmark servers - is trivial to merge by virtue of the underlying RDF-based data representation. It is also possible to add additional application- or user-dependent RDF properties such as links to formal ontologies to bookmarks and topics. Applications designed according to the principles of RDF will be able to accommodate this additional data even if the application can not present or edit these novel properties. The Annotea framework would also support objects that have a dual role as both bookmarks and annotations. In addition, other metadata based formats, such as RSS newsfeeds, could be presented as sources of Annotea shared bookmarks.

Keywords

Annotation, bookmark, topic, ontology, metadata, Semantic Web

INTRODUCTION

The Web is a collaborative space that lets users share their thoughts, their work, their images, their ontologies, and other aspects of their life by publishing Web pages and linking them to other information. Collaboration via shared Web pages is constrained as long as readers of Web documents cannot easily also add their contributions to appear in the document context.

Shared Web annotations support common collaborative tasks, such as sharing comments, questions, or brief discussion threads in the Web. They are one important means supporting information sharing and collaboration within groups of co-workers, students preparing a report, or any other group.

In addition to sharing comments, collaborating users often also need to share lists of interesting Web documents and their categories or topics. These topics are usually very informal at first and evolve gradually as more understanding about the studied area is acquired.

Collaboratively-maintained indices of the Yahoo variety have made different design choices including an evaluation and review process that is often cumbersome for highly dynamic groups.

Bookmark functionality and bookmark categories help present interesting documents in a hierarchical category view. However, most current browser implementations do not support sharing or exchange of bookmark data other than in XHTML format. With the exception of XBEL [29], the current bookmark exchange formats also severely restrict the additional machine readable information that can be preserved in the bookmark database.

In addition, the traditional browser bookmark user interfaces do not capture as much information as users might wish. Similar to annotations, bookmarks are normally created in the document context while the user is viewing a document, but later when the user revisits the same context the browser does not give any clue to the user that she has previously bookmarked the page or about the categories that it might belong to. Bookmarks are only presented in informal categories or topics and users browse bookmark lists in hierarchical views.

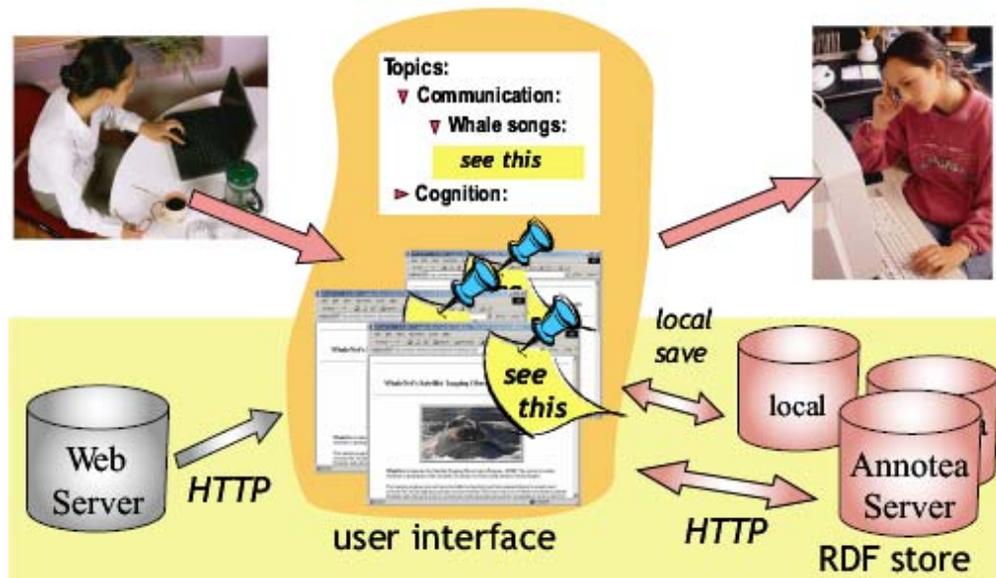


Figure 1: Annotea bookmark architecture

Bookmark data should be useful and available in many other ways. Rendering the bookmark with traditional annotation like icons on the page when the user so prefers would help the user recall context and reduce duplicated work. Shared bookmark data containing classification information would benefit users who are collaborating in similar research areas. They could open a bookmark shown on a page to find other related topics and follow them for bookmarks to other related documents.

Annotea [2, 19] is a W3C Semantic Web Advanced Development [28] project providing a framework for rich communication about Web pages through shared annotations based on the W3C's Resource Description Framework [22] metadata interchange specifications. Annotea uses RDF so that it can be easily extended to support many kinds of annotations and annotation-like collaborative applications such as bookmarks that share metadata about Web documents. This metadata can be stored either locally or in one or several user selected servers and retrieved by other users who have subscribed to and have permission to access those servers. Annotea clients, such as Amaya [4], query the local RDF stores and the selected server RDF stores and present the annotations and bookmarks to the users.

Shared bookmarks use the same Annotea framework to create and maintain cooperative community indices grouping Web documents into informal categories or "topics". Annotea bookmarks are metadata about Web documents or other resources and share a similar structure with annotations. A bookmark can be catalogued under one or more topics and these topics can be presented to the user as topic hierarchies or more complex graphs. Bookmarks and topics are unambiguously identified with a URI, making it easy for users to share both the categories and the documents placed in the categories. An Annotea client

could declare bookmarks to also be annotations and then present bookmarks as icons on a Web document when the document is browsed. This would help a user know immediately if she already has bookmarked a document in a certain category.

Use of metadata and URIs makes it easy to share and merge bookmarks coming from different sources. Bookmark databases with somewhat different RDF properties are still readily mergeable. A variety of presentations of the RDF data is possible. Use of this metadata design makes it easy to share bookmarks with other users, share bookmarks between browsers, and query the bookmark data in new ways not supported by current bookmark systems. Through additional properties in the data instances it is also possible to associate bookmark categories (topics) with more formal ontologies.

In the following sections, we will describe in more detail the Annotea architecture for shared bookmarks, the bookmark schemas and examine some future directions for further developing Annotea.

ANNOTEA ARCHITECTURE FOR BOOKMARKS

The shared bookmark application uses the same Annotea architecture as is used for annotations (see Figure 1). The annotation and bookmark metadata is stored separately from the Web documents themselves. The Hypertext Transfer Protocol is used to access this metadata from one or more RDF metadata servers. Metadata may also be stored in RDF form in local files. A user can create bookmarks and associate them with Web documents by using the functions in the client provided user interface. The same user or another user can then later use the bookmarks user interface to help find the bookmarked Web documents again or find other related information. Client implementations are free to present bookmarks in any way

the developer chooses. Figure 1 also illustrates possible ways in which the user may examine the bookmarks in a topic hierarchy or as icons in the document context in a similar way as annotations are currently presented in our Amaya client.

Each individual bookmark is represented as an RDF resource of type `Bookmark`. The metaphor of folders for arranging bookmarks under informal categories or topics has been generalized to resources of type `Topic`. Properties of Topics as well as of Bookmarks are expressed as RDF statements and stored in local files or in one or more bookmark servers via HTTP [13].

SCHEMA FOR SHARED BOOKMARKS

The Annotea bookmark schema [5] defines three basic resource types: bookmark, topic, and shortcut. The bookmark and the topic resources provide the basic concepts found in common browser bookmark implementations as well as in XBEL [29].

XBEL data can be easily mapped into the Annotea bookmark schema, though the reverse (Annotea RDF bookmark data to XBEL) will frequently involve loss of information. The shortcut resource type is intended to be used in scenarios where a user wishes to import data and add properties without exporting the added properties. Our current implementation does not yet expose an interface to shortcut resources.

The use of RDF in Annotea permits shared bookmarks to express additional semantics either from other bookmark, topic or annotation schemas or other not directly related schemas. For example, a bookmark can include other properties from the Dublin Core element set [11]. As a general rule, the client implementations are expected to preserve all properties of a bookmark, even if they cannot present or interpret them.

We briefly describe below the most common properties use to describe an Annotea bookmark. Annotea uses the RDF/XML namespace facility to declare and merge properties from multiple namespaces. The classes and properties described in this document use the following conventions for the namespace references:

- *a*: annotation namespace [1]
- *b*: bookmark namespace [5]
- *dc*: Dublin Core namespace [10]
- *rdf*: RDF namespace [26]

Bookmark Resources

An instance of a bookmark is declared by assigning it the *rdf:type* of *b:Bookmark*. Figure 2 shows an example of a bookmark instance with core bookmark properties, using the conventional RDF node and arc presentation syntax. Each instance of a bookmark is identified with a unique URI reference (URIref), here abbreviated to "F2C75X5" labeling the node that corresponds to the bookmark instance. When a bookmark URIref is an *http:* resource it is expected that an HTTP GET using that URI will return

RDF content containing properties of the corresponding bookmark.

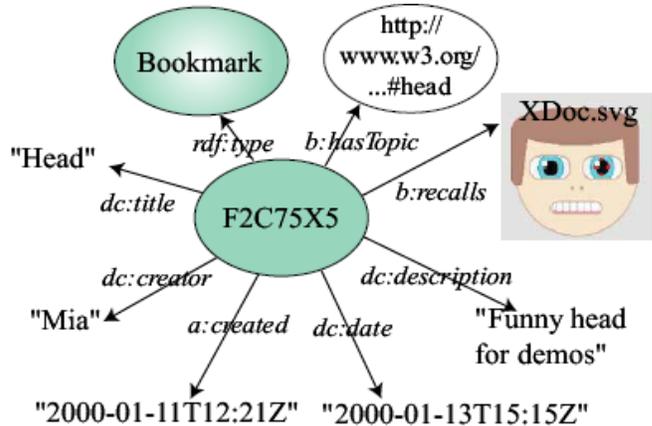


Figure 2: An instance of a bookmark.

The document that the user has bookmarked is identified with the *b:recalls* property. The role of *b:recalls* is to identify the primary document to which the bookmark refers. The *dc:title* property is used to assign a title to the bookmark. The *dc:description* property may be used to give a longer textual description for the bookmark. To facilitate sharing of bookmark data, the *dc:creator* property is used to name the author who created the bookmark. The *a:created* and *dc:date* properties represent the creation and last modification time, respectively. The *b:hasTopic* property associates the bookmark instance with an instance of a topic. Topics are described below. We chose to model the the 'hasTopic' relationship as being a relationship between a bookmark instance and a topic rather than between the document that was bookmarked (identified with *b:recalls*) and a topic. This model is more flexible, as it allows a user to use a bookmark for any purpose without forcing the user to declare a property of the document itself.

Topic Resources

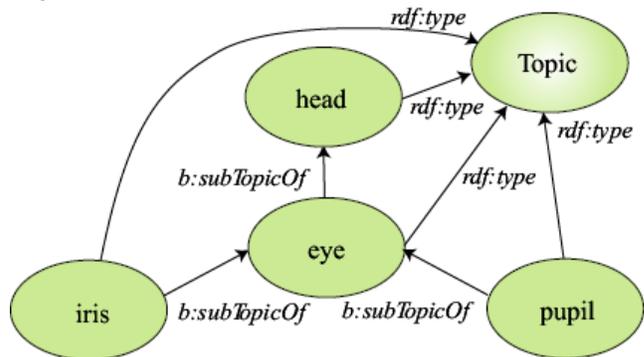


Figure 3: An instance of a topic hierarchy.

A topic defines an informal category for the purpose of classifying bookmarks. Topics may have subtopics and may sometimes also refer to categories in more formal

ontologies. In a similar way as bookmarks, each instance of

a topic is identified with a unique URIref.

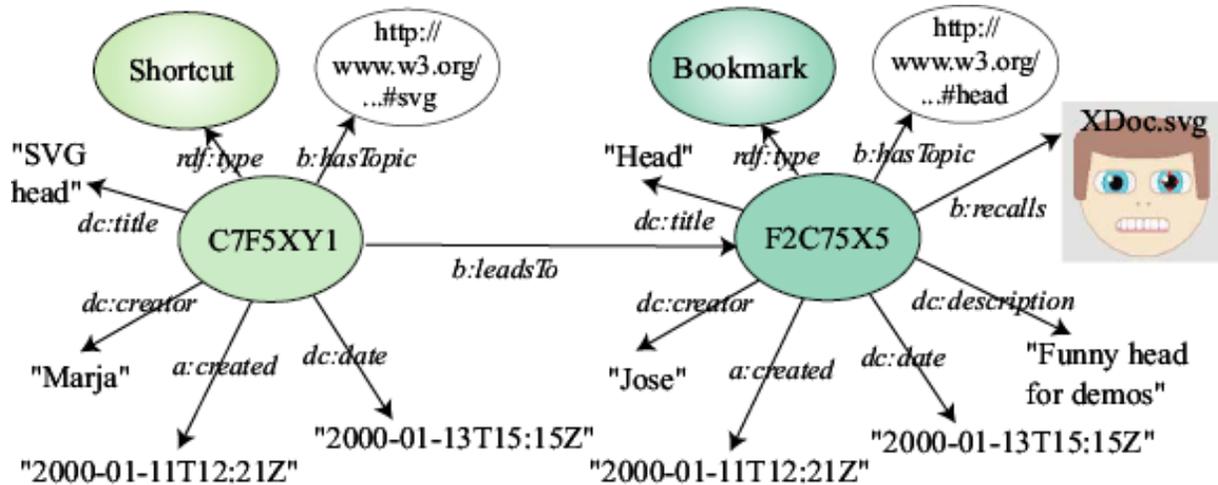


Figure 4: An instance of a shortcut and a corresponding bookmark.

Figure 3 presents an instance of a topic hierarchy associating parts of an anatomy. The *b:subTopicOf* property defines the hierarchy. While the example in Figure 3 shows a strict hierarchy, the use of *b:subTopicOf* is not restricted to trees; full graphs are permitted. To keep the figure simple, most other core topic properties are not shown as these are similar to Figure 2. The *dc:title* property assigns a title to the topic and the *dc:description* property is used to give a longer textual description for the topic. The *a:created* and *dc:date* properties also have the same meaning as for bookmarks.

Shortcut Resources

When sharing bookmarks, a user may wish to associate a bookmark with a personal classification scheme without declaring that the shared bookmark itself is really in this personal topic. The same user may wish to include Topics from a formal classification structure in a personal topic hierarchy where it may be inappropriate to declare that the formal topic has a

b:subTopicOf relationship to the user's personal hierarchy. In such cases the user may define a "shortcut" to a bookmark or a topic. A shortcut is an indirect reference to another object.

Figure 4 shows an instance of a shortcut. A shortcut has its own *b:hasTopic* or *b:subTopicOf* properties depending on whether it is making a shortcut to a bookmark or a topic. The indirect reference is made with a *b:leadsTo* property. The client follows the *b:leadsTo* property to access the corresponding bookmark or topic as necessary.

USING BOOKMARKS IN AMAYA

Client implementations are free to present bookmarks in any way the developer chooses. Currently, Amaya has implemented the basic bookmark and topic related functions. We also have separately implemented and tested the bookmark HTTP protocol with a bookmark server but

need to put the user interface and the shortcut resources together with it.

The user creates a bookmark by selecting the New Bookmark item in the menu. This opens a bookmark window similar to the screen capture shown on the right in Figure 5. The user may choose to use default values for the bookmark or use the selection widgets in the window to choose one or more topics in the Topic Hierarchy field to classify the bookmark and write a title and description for it. The user can create a new topic for the hierarchy by using the New Topic button, at the bottom of the window or select New Topic from the menu.

The View Bookmarks menu item opens a hierarchy window presenting bookmarks in a topic hierarchy. Figure 5 shows a hierarchy window on the left side with the bookmark and topic hierarchy that is fully defined in [27]. The user can select a bookmark in this hierarchy and display its properties as shown in the right-hand window.

FUTURE DIRECTIONS FOR ANNOTEA

There are many possible avenues for enhancing the Annotea framework in the future. Annotea and many Annotea clients, such as Amaya and Annozilla, are open source and welcome contributors. Our goal is to have easy to use shared annotation and bookmark tools but also to make Annotea framework easy to extend so that it can support many collaborative annotation- like applications and combine functionalities from them when it makes sense to the user.

An RDF model of bookmark and topics makes it easy to present other information in the bookmark format. This allows the user to select bookmarks when the bookmark user interface helps her be more effective. For instance, it is easy to define transformations from other bookmark formats, such as the XBEL format. In addition, it is also

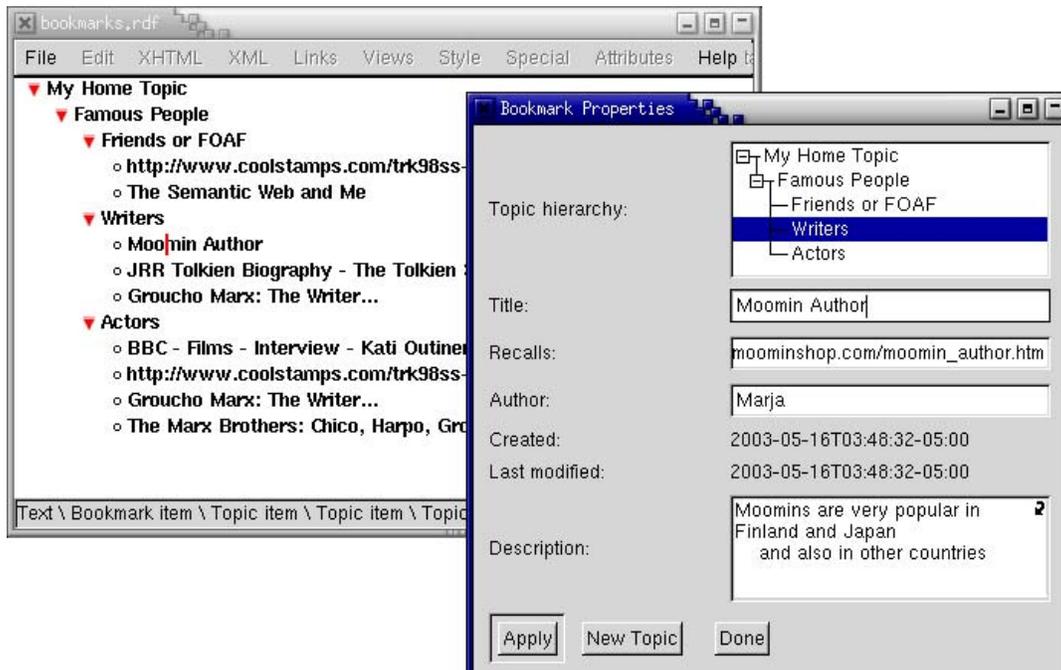


Figure 5: View of topics and bookmarks [27] with one open bookmark window in Amaya.

possible to declare semantic relationships for formats not originally designed for bookmark use that then permit tools to interpret other data, such as RSS [6] newsfeeds, to be presented as Annotea shared bookmarks. We are hoping to get some help in creating a group of tools for making these conversions.

Many groups of users working closely together want to define their own annotation or bookmark subtypes to convey more specialized semantics. After creating annotations and bookmarks we expect that users will find it useful to be able to merge them and create hybrids with functionality from both. With new RDF metadata definitions it is straightforward to create new annotation or bookmark types and also define new RDF properties for these types. Currently in Amaya these modifications are done by updating a list of schemas that Amaya reads at startup. In the future, we would like to enhance this and make the definition of new annotation and bookmark types more dynamic, and to dynamically build parts of the user interface from the schemas without need for user interface programming every time a new property is added. Some preliminary investigations have been made into a stylesheet language in cooperation with IsaViz GSS [18].

There are also needs for easily changing the styles of annotations and bookmark presentations. Arakne [7] offers interesting examples of dynamic presentation styles. Currently in Amaya, annotations are presented with special link icons embedded in document views. The user clicks on an annotation icon to show the content of the annotation in a new window. The bookmarks are presented in a topic hierarchy but could also benefit from being presented with bookmark icon links. Annotea has the some basic

capabilities for this, for instance, we can add the annotation type to a bookmark and use the presentations designed for annotations. However, more work is needed for this to be useful. In the future, we would like to have a CSS type presentation language for defining and selecting the presentation style. We also expect to use Semantic Web logic rules to filter annotations and bookmarks according to users' needs.

RELATED WORK

URIs are designed to be created according to whatever structure the administrator of the Web server finds most appropriate for the particular site. In general, users are not expected to have to recite or remember URIs. As a consequence, bookmark systems in common browsers such as Internet Explorer, Netscape, and Opera provide functionality to store the URI with some information that helps users identify it, such as descriptive title and information about its container or folder. Often browsers or separate converter tools [23] also support exporting and importing functionality to some bookmark formats. This is important for users who frequently change browsers and/or have several devices that store their bookmark content. Unfortunately, the formats are seldom interoperable between different browsers. Conversion tools are done case-by-case and often require the use of several tools in a row. Export formats such as XHTML do not preserve some machine understandable information making other conversions less precise. XML Bookmark Exchange Language [29] is a proposed bookmark language designed for solving the export and import problem. It is currently available in some Linux-based browsers [14, 15, 21]. XBEL provides a good format for the basic bookmark

functionality. However, it is hard to add new bookmark properties or merge bookmark data from several sources as plain XML does not have a general mechanism for doing this operation. Mozilla stores bookmarks internally in an RDF datasource [24]. This bookmark RDF is presented in selected areas of the user interface, such as the personal toolbar, through XUL Templates. Unfortunately RDF seems to be used only internally and not as an exchange format; when a users exports her bookmarks from Mozilla, the user only has a choice of creating an HTML file. The current surface implementation of shared bookmarks in Amaya looks similar to that offered by many browsers. However, the use of the Semantic Web technologies makes merging bookmark data from several sources easy. Also, it is possible to use many other RDF based tools, such as IsaViz, for presenting bookmark data in novel ways. Furthermore, automatic conversions to other annotation or bookmark formats can be made more precise because it is possible to refer to objects and properties in the schema unambiguously.

We have some experience with transformations of annotations. For instance, the IBM SemTag [9] project uses Annotea annotations to present their automatically created corpora of 434 million automatically disambiguated semantic tags in the context of the tagged documents. Accessibility evaluations, such as EARL [12], can be presented as annotations in the document context. With bookmarks, we expect to be able to create RSS newsfeeds from bookmark data or vice versa to present the RSS newsfeeds as bookmarks in a topic hierarchy. This integrates the news feeds with the user's own information organization and makes their structure easy to navigate by following the individual news links with a browser. Also Weblog or sitemap information could be presented in the shared bookmark format.

There is a growing body of annotation and Semantic Web related work [17]. Annotea is focusing more on supporting collaborative work with informal bookmarks and annotations than building formal ontologies. However, the RDF metadata approach lets users add additional metadata for the bookmarks and tie them into more formal ontologies, such as the ones defined by CREAM [16] or Protege [25].

Often users don't only want to share bookmarks but also good topic hierarchies. For instance, students may sometimes benefit from copying a topic hierarchy commonly used in the research field or a group of researchers may want to start gathering research information in a commonly agreed hierarchy and then later adapt to a custom hierarchy to classify their Web documents. With Annotea it is easy to copy only skeletons of topic hierarchies when needed.

Annotea-like metadata based annotations and bookmarks can be used not only in browsers but also with other metadata as part of other clients or environments. For

instance, Haystack adopted RDF metadata based annotations for searching different personal information objects by using the natural language content of the annotations [20]. Haystack uses a different schema than Annotea, but using Owl [8] we can describe correspondences between these schemas.

CONCLUSIONS

Shared bookmarks is another annotation-like application using Annotea architecture. Shared bookmarks is a tool that permits users to informally classify Web resources and share these classifications. Adding bookmark data to Annotea metadata store ensures that the basic Annotea architecture is flexible and extensible and has possibilities to support many similar kinds of applications either directly or by using some transformation rules. The biggest work ahead is to find ways to support the user interface of new annotation or bookmark types so that new annotation and bookmark types can be easily created and presented in various ways without a need to write client code to present them.

ACKNOWLEDGEMENTS:

Annotea is result of team work. Jose Kahan has enormously influenced the current Annotea design and also implemented the current Amaya interface for annotations and bookmarks. Annotea also has many valuable contributors and collaborators both within W3C and outside and we highly appreciate the work of this community. Some of them are mentioned in Annotea contributors page [3].

Development of Annotea is supported in part by funding from US Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-00-2-0593, "Semantic Web Development", by the European Commission Information Society Technologies (IST-7) programme under SWAD-Europe, and by Elisa Communications Oyj (Finland).

REFERENCES:

1. Annotea Annotation namespace
<http://www.w3.org/2000/10/annotation-ns#>
2. Annotea home page <http://www.w3.org/2001/Annotea/>
3. Annotea contributors page
<http://www.w3.org/2001/Annotea/Contributors.html>
4. Amaya home page <http://www.w3.org/Amaya/>
5. Annotea Bookmark namespace
<http://www.w3.org/2002/01/bookmark#>
6. Beged-Dov, G. et.al., RDF Site Summary (RSS) 1.0
<http://web.resource.org/rss/1.0/spec>
7. Bouvin, N. et.al.. Fluid Annotations Through Open Hypermedia: Using And Extending Emerging Web Standards, in *Proc. of the WWW2002 International Conference, Honolulu, Hawaii, May7-11, 2002.*
<http://www2002.org/CDROM/refereed/656/index.html>

8. Dean, M., and Schreiber, G. (eds.), Web Ontology Language (OWL) Reference Version 1.0. W3C Working Draft, 31 March 2003 <http://www.w3.org/TR/2003/WD-owl-ref-20030331/>
9. Dill, S. et al., Semtag and Seeker: Bootstrapping the Semantic Web via Automated Semantic Annotation, In *Proc. of the WWW2003 International Conference*, Budapest, May 2003. Pages 178-186. <http://www2003.org/cdrom/papers/refereed/p831/p831-dill.html>
10. Dublin Core namespace <http://purl.org/dc/elements/1.1/#>
11. Dublin Core home page <http://dublincore.org/>
12. EARL home page <http://www.w3.org/2001/03/earl/>
13. Fielding, R. et al. (eds.). Hypertext Transfer Protocol - HTTP/1.1. RFC RFC2616, IETF, Jun. 1999. <http://www.ietf.org/rfc/rfc2616.txt>
14. Galeon home page <http://galeon.sourceforge.net/>
15. Grail home page <http://grail.sourceforge.net/>
16. Handschuh, S., and Staab, S. Authoring and Annotation of Web Pages in CREAM. In *Proc. of the WWW2002 International Conference, Honolulu, Hawaii, May 7-11, 2002*. Pages 462-473.
17. Handschuh, S., and Staab, S. (eds.), *Annotation for the Semantic Web*. Volume 96 Frontiers in Artificial Intelligence and Applications, 2003, 240 pp.
18. IsaViz homepage <http://www.w3.org/2001/11/IsaViz/>
19. Kahan, J., Koivunen, M., Prud'Hommeaux, E., and Swick, R. Annotea: An Open RDF Infrastructure for Shared Web Annotations, in *Proc. of the WWW10 International Conference, Hong Kong, May 2001* <http://www10.org/cdrom/papers/488/index.html>
20. Karger, D. et al., Sticky Notes for the Semantic Web. In *Proc. of the IUI2003 International Conference on Intelligent User Interfaces, January 2003*.
21. Konquerer home page <http://www.konqueror.org/>
22. Lassila, O., and Swick, R. (eds.), Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation, 22 February 1999 <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222>.
23. LinkaGoGo home page <http://www.linkagogo.com/>
24. Mozilla rdf:bookmarks home page http://www.xulplanet.com/references/elemref/ref_rdfbookmarks.html
25. Noy, N. et al.. Creating Semantic Web Contents with Protege-2000. *IEEE Intelligent Systems* 16(2):60-71, 2001.
26. RDF namespace <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
27. Sample bookmark file <http://www.w3.org/2003/10/bmsample.html>
28. SWAD home page <http://www.w3.org/2000/01/sw/>
29. XML Bookmark Exchange Language <http://pyxml.sourceforge.net/topics/xbel/>