

ANNOTIZER: USER-FRIENDLY WWW ANNOTATION SYSTEM FOR COLLABORATION IN RESEARCH AND EDUCATION ENVIRONMENTS

Berndt Jung¹, Ilmi Yoon¹, Hendra Lim¹, F. A. Ramirez-Weber², Dragutin Petkovic¹

¹ Computer Science Department, San Francisco State University

² Biology Department, San Francisco State University

ABSTRACT

This paper presents a new version of our WWW-based annotation system, Annotizer. Annotizer improves collaboration on the WWW by allowing users to create, share and search annotations on existing HTML WWW contents. The focus of this paper is on the design and implementation of the second version of the Annotizer specifically targeting educational and research settings. The two general questions we explore are: (1) How useful are WWW annotations and what purpose do they serve? And (2) what are design and implementation issues for delivering a user-friendly WWW annotation system? The second version of the Annotizer is improved in that it allows the annotation of any WWW page (not only those from the Annotizer server domain) and it employs a much improved UI. While we believe that the Annotizer can be useful in variety of WWW collaborative applications, our initial target is users within the scientific and educational communities.. The Annotizer requires no downloads and the annotations are stored in a central database allowing asynchronous adding, viewing and searching by all registered users. In this paper we present the motivation, new functionality, architecture and UI of Annotizer.

KEYWORDS – Annotation, Collaborative Research, Web technology, JavaScript

1. Introduction

Typical collaborative patterns in WWW usage are as follows; the majority of information is posted by web site owners and then users contribute their information or opinions through discussion boards, ripples or blogging styles. Lately, using Wiki technologies, users are allowed to directly edit the main WWW content. The above mechanisms significantly contribute to WWW collaboration, and they can often create their own communities (i.e. discussion or study groups). These methods work well if discussion or feedback is made on the subject of interest as a whole rather than specific part of WWW contents. However, when feedback or questions need to be pointed to a specific part of the documents (e.g. portion of the text or an image), the discussion board, ripples or bloggings may not be effective. In such cases, communication and collaboration through more precise annotations may serve as an efficient collaboration tool for study groups, researchers, educators, and students.

In order to understand the effectiveness of the annotation paradigm, we studied traditional annotations first, which are made by user markings on hardcopy documents. Reading those annotated documents (markings and notes related to specific parts of documents), or “active reading”, substantially improves the comprehension of the material. Marshall [1] studied the effectiveness of annotations, and developed seven different categories of annotations. Some interesting categories include:

- Commentary, where users editorialize information with related facts or opinions.
- Highlighting/Marking in or around text, where the purpose of an annotation is simply to draw attention to a single segment of a document.
- Link Making, where readers seek to clarify connections between information contained in a document with other information, either for themselves or for a community.
- Reading records, where readers use annotations as a memory aid.
- A community memory bank, where annotations serve as a vehicle for expressing the evolution of thought or an archive of interaction for a community of contributors.

Marshall also noted that annotations are a primary vehicle for supporting collaboration around documents [2]. When such annotations are implemented on the WWW, with additional functionality that WWW enables (i.e. sharing, searching), we believe that they can significantly contribute to enhancing both web-navigating experience and collaborative activities through the WWW.

In the context of our application for annotations, namely in the area of collaboration for education and research, the key use cases are based on the fact that main (base) content is generally static (i.e. papers, research results, class material etc.) and that collaboration requires annotations as a vehicle for comments, opinions, clarifications and discussions related to specific portions of that base content (text segments, image areas etc.). For example: key researcher in the field may post annotations on portion of somebody’s paper, or a student might post annotations for their instructors asking for clarification on a particular piece of text or image. Consider also the need for the association of metadata with large data sets such as gene sequences or image sequences as an example. Using an annotation mechanism, the repository’s annotation data can evolve over time to become a richer, more comprehensive

representation of the underlying data set. Annotations on the Web, created by trusted experts, in essence could become an important and often critical “layer of content” by itself. The fact that in our application’s base content is relatively static alleviates the problem of “orphaned” annotations (i.e. annotations whose “anchor” changed or disappeared). Our first application of the Annotizer will be used in conjunction with the Hedgehog Signaling Pathways (HSP) web site. The Hedgehog protein super family constitutes a group of closely related secreted proteins that control many crucial processes during the embryogenesis of tissues. The HSP Web site intends to provide content relevant to researchers and students involved in HSP work. This content includes access to extensive and manually verified base content (papers, databases, measurements) as well as manually verified links to closely related resources at other web sites. Due to the complexity and massive amount of data, even experienced users often get lost at the ocean of information. To make the data at repository effectively absorbed by many users, HSP users have expressed a strong desire to have some sort of content annotation.

The Annotizer system described in this paper is the second version of our original work reported in [3]. The key innovations in this version of the Annotizer are: a) a completely new architecture allowing the ability to annotate any WWW page (not only WWW pages from the Annotizer server domain as in the original version); b) a completely new UI with several new functions. The Annotizer was designed based on collected feedback from user groups and our extensive use case analysis.

The structure of the paper is as follows. Related work is discussed in section 2. In section 3, we present the design issues, architecture design and new UI of the Annotizer. In section 4 we present the status and planned deployment of the Annotizer. Section 5 presents conclusions and future work.

2. Related work

Hypertext annotation is a rapidly expanding area of inquiry [1, 4, 5, 6]. One reason for this popularity is the utility of annotations in large data sets. Marshall et al. note that annotations are a primary vehicle for supporting collaboration around documents in their studies of the XLibris project [1]. They allow users to conduct on-going discussions over content in documents, support serendipitous sharing of information about documents, and allow experts to guide others through a particularly complex or large set of documents. The overall goal, they note, is primarily a community building exercise through the annotations, where people with similar interests can connect through the exchange of informal data around formal information repositories.

Carter et al. study the effectiveness of multimedia annotation by providing methods for people to post and acquire digital information to and from public digital displays, and to modify and annotate previously posted content to create publicly observable threads [7]. The experiment was focused on interactions on posted contents

of publicly observable threads using public digital displays such as plasma poster and PDAs as interacting devices.

YAWAS is a simple and lightweight web-based annotation application which allows users to highlight part of the text content within web pages, enter annotations into a database, and search for annotations utilizing Google. However, YAWAS is mainly developed for individual users who can save the database file only at local machines for personal lookup as sophisticated bookmarks. In addition, YAWAS requires installation of plug-in.

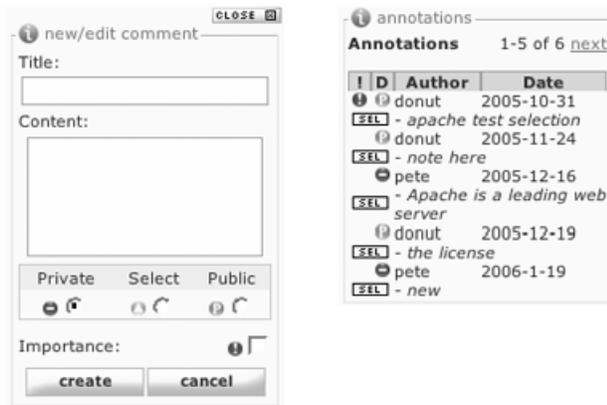
Annotea is a W3C project for annotation and use an RDF based annotation schema for describing annotations as metadata and XPointer for locating the annotations in the annotated document [8]. Annotea is a server application which requires a separate client to interact with. Currently, two clients exist, the Amaya W3C browser, and the Mozilla Firefox browser with the addition of three extension components [9]. Annotea shares similar ideas with the Annotizer, but by requiring specific browser functionality, Annotea will always be limited to supporting browsers, or browsers with additional downloaded components installed. In addition, vehicles for wide-spread use such as, annotation sharing amongst selected user groups, URL subscriptions and image annotations are not supported by the Annotea model.

Ont-O-Mat-Annotizer is developed for authoring and Annotating web pages [10]. Ont-O-Mat Annotizer not only support users to create meta data for an existing web page, but also provides a tool to author a web page as well as produce meta data using ontology at the same time. Focus of Ont-O-Mat Annotizer is creating semantic web from existing web pages rather than helping users to comment, clarify or discuss using annotations.

Recently, Wiki pages are gaining popularity by



Figure 1. Screen shot of Annotizer; The Annotizer wraps a set of tools and an interface around HTML documents (web pages). The end result is a browser within a browser, complete with an address bar, and history navigation. The area marked by “Annotizer Tools” is where user can add, view and search annotation. The area marked by “To be annotated” is regular navigation area where user can go to any site on the WWW.



a. Annotation Edit panel b. Annotation view panel

Figure 2. Portions of Annotizer interfaces

allowing users to directly edit the whole content of the WWW site [11]. The difference of Wiki approach compared to the Annotizer is that the Annotizer deals with an “annotation layer” over the base content which itself is not modified, thus offering different paradigm (that of “comments” vs. “content submission”).

Web Discussion [12] and iMarkup [13] are commercial and proprietary annotation systems supporting multiple users. iMarkup specifically supports multimedia annotations such as image annotation, voice annotation, and sticky notes. Both products require client application downloading and installation and have limited portability. WWW Discussion, YAWAS, and iMarkup are not available for the Linux and Mac environments yet.

In our original work [3] we presented the first version of our annotation system that had almost full functionality for HTML text and image annotations, but could handle only pages from the same domain as the Annotizer main server which posed sever limitations to it usefulness. Its user interface was also not optimal. In our limited usability study we concluded that users find such system effective, useful and relatively easy to use.

3. The Annotizer Architecture

The development of the Annotizer was motivated in large part by the scientific and education community (in our case biologists using HSP site) who are eager to use annotation for their collaborative research. Based on our research and discussions with potential users, we defined the following high-level requirements and functionality for the Annotizer:

- In order to be able to create and access annotations, users have to register on the Annotizer site.
- Users’ access annotation functionality through a “drawer” located on the left hand side of the browser, (Fig. 1). The “drawer” can be hidden when not in use.
- Users shall be able to post and access annotations at any time, from any modern browser.
- Basic functionality shall include adding annotations to HTML text and images, viewing annotations and searching for annotations.

- Annotation functionality shall not require any downloaded or installed software or plug-ins.

The requirements can be categorized into their impacts on (1) user interface design such as minimization of number of clicks and (2) implementation requirements such as portability across diverse platforms and browsers without requiring a separate program or browser plug-in which forces us to work within the constraints of JavaScript and browser renderable elements. In this section, the first part focuses on user interface design issue while the second focuses on implementation architecture.

The Annotizer needs to provide a set of tools for adding, viewing and searching annotations while allowing user to navigate WWW, and not require any external application or browser plug-in. The end result is an annotation tools enhanced browser, complete with its own address bar, and history navigation, within a browser.. In order to use the Annotizer users must first login with a valid username and password and then use the field titled “Annotate this page” to navigate the WWW just as users usually use the address bar of a standard web browser. It is important not to use the main browser forward and back buttons, but the Annotizer navigation buttons instead, in order to stay within the Annotizer while navigating the WWW.

There are two types of annotations supported by the Annotizer, page and text selection. A page annotation is a comment specific to the entire page. A text selection annotation is a comment specific to selected text on a specific page. To create a page annotation for the current page, user just clicks on the “page” button in the main menu and then enters annotation info in the form that will appear after the click. To create a text selection annotation, user first selects the text to annotate using left mouse button which highlights selected text, then clicks the “selected text” button to display the panel at figure 2a of the Annotation Tool area, and then enters the annotation information in the form. This approach allows one-mouse-click operation and also allows standard usage of cut and paste. It was selected after many design discussions and replaced the model used in our first version of Annotizer, which precluded the use of cut and paste.

Figure 2b depicts annotations available to view at a particular page; if annotations have been created on a page a user is visiting, the annotations window will be populated with annotations which the user is allowed to see. These include all annotations created by the user, as well as all annotations labeled public, and any select annotations which the user has been invited to see. By clicking on an item, the full annotation content will be shown within the Annotizer Tools frame and the associated text will be highlighted at the web page (Figure 3). Users may sort the table by clicking on the column title as well as search any annotations using keywords.

Users have several options to decide how to share users’ annotations: “Public” will allow anybody who is a registered Annotizer user to see the annotations. There is an option to send an e-mail alert to selected users regarding a specific annotation. “Private” will allow the annotation to be

visible only to the owner of the annotation. “Select” allows the annotation to be visible only to selected people (via e-mail entry), and by the annotation owner. Annotations may also be set to the different level of importance. Users may also choose to *subscribe* to particular pages. In this case, the user is alerted via email when annotations visible to the user are posted to the subscribed page.

To search annotations, users either type the text in the search field or use Search tab for more advanced options. Search results will appear as the list of page URLs that contain relevant annotations. Upon clicking on a selected URL the requested URL will be loaded, and the selected annotation will be displayed.

Data is stored in central relational Data Base System. In case of text annotation, this data includes annotation comment, a selection, an author (user), and a URL (page). The comment itself is made up of fields which store creation and modification dates, priority (importance), a subject or headline, the text content of the annotation, a public field which refers to visibility of the annotation, and ids which refer to selection, user and url entities (Figure 4).

To meet portability and the no plug-ins requirements, JavaScript was chosen as a primary tool. JavaScript natively allows scripts to interact with only pages from the same domain due to security issues. Therefore, we cannot interact with a page of one domain in one frame (The Annotizer home page, tools area) with a page from another

domain (any site outside of Annotizer home site) in another frame using JavaScript. Therefore, a request for a new page while navigating through the Annotizer is sent to the Annotizer server, not to the intended WWW site. The Annotizer server handles the request in the following ways. The Annotizer engine forwards the request and fetches a page file (helloworld.html at figure 5 example) from the originally intended WWW site. Then this file (helloworld.html) is processed in Annotizer server in order to allow it to be annotated. To do this the file is parsed i.e. “indexed”. Once the page has been parsed, we output (render) the resulting encoded HTML back to the client, in main Annotizer viewing area. The encoded page, from the browser’s perspective is from the same domain as the rest of the Annotizer content and scripts, and hence annotation tools executed in JavaScript can deal with user selections (i.e. annotation targets) from that page. The parsing or indexing process adds unique id attributes to all of the HTML tags within the document, and rewrites all links to keep proper reverence to other HTML resources. These ids are used to store in the central database and later find (locate) the placement of the annotations. When the pages are revisited, the annotations can consistently be placed correctly. Originally, the ids were just consecutively ordered integers. The problem arose that even slight modifications to an HTML page would prevent the Annotizer from re-locating the annotation. For instance, adding an image would affect all of the ids of tags below the image. Rather than just

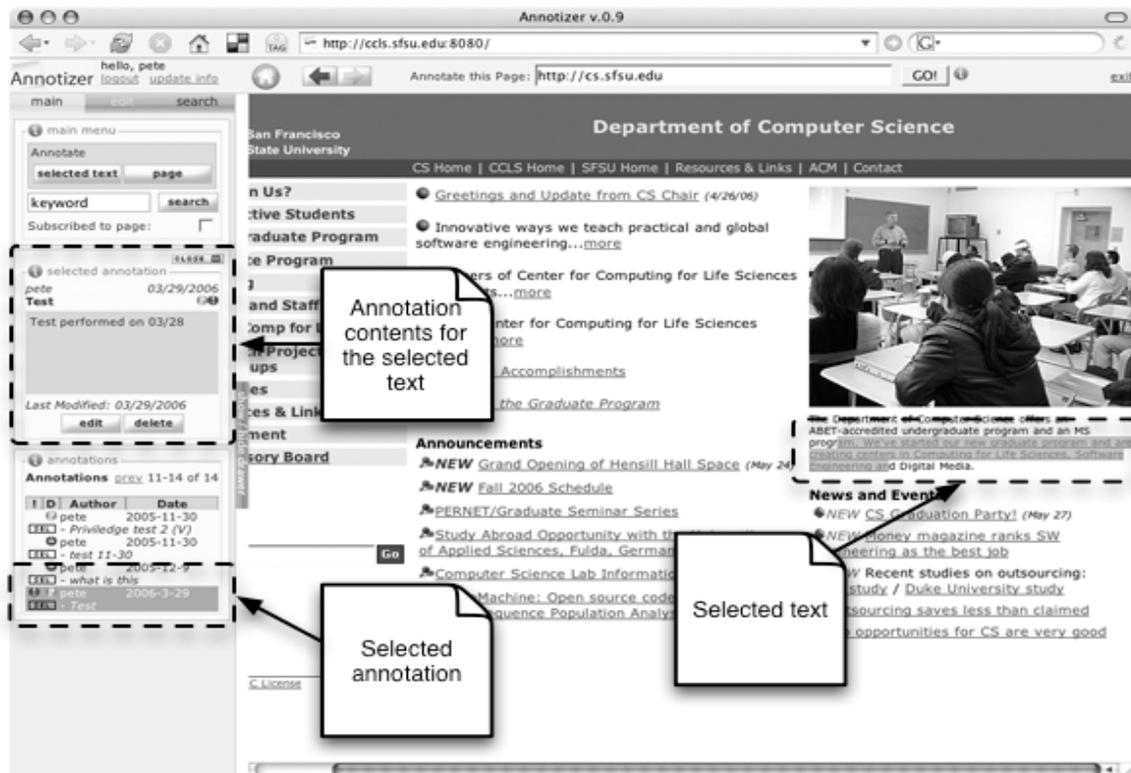


Figure 3. If annotations have been created on a page a user is visiting, the annotations window will be populated with annotations which the user is allowed to see. These include all annotations created by the user, as well as all annotations labeled public, and any select annotations which the user has been invited to see. By clicking on an item, the full annotation content will be shown within the Annotizer Tools frame and the associated text will be highlighted at the web page.

number consecutively, we calculate a hash code of the content of the annotation target, and not just its position on the page. The hash code is a calculated number, which corresponds to the text content of the tag. Therefore, if annotated target text was moved, its id remains the same. When HTML pages are slightly modified, the annotation data remain useful.

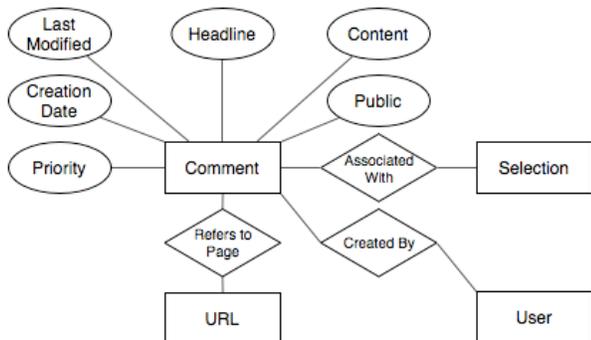


Figure 4. Each text annotation is made up of a comment, a selection, an author (user), and a URL (page). The comment itself is made up of fields which store creation and modification dates, priority (importance), a subject or headline, the text content of the annotation, a public field which refers to visibility of the annotation, and ids which refer to selection, user and url entities.

To parse and index the HTML we used an open source NekoHTML parser based on the Xerces2 xml parser [14]. Both are distributed under the Apache Software Foundation. The parser is open source and produces a w3c DOMDocument object as a result of parsing. An issue related to parsing procedure is that parsing needs to handle malformed HTML robustly. An example is non-valid XHTML pages. Unlike HTML, XHTML is required to be well-formed. However many non-valid XHTML files exist on the WWW. We require a parser which invisibly handles or fixes malformed HTML. We evaluated many freely available parsers and the NekoHTML parser was found to handle a diverse range of malformed HTML. In our experience, it can handle most HTML pages.

Rewriting relative links is required to keep the pages within the Annotizer domain. Were the links not rewritten, following a link would cause the linked URL to be loaded from outside of the Annotizer domain. All of the Annotizer tools would no longer work, as javascript could not communicate with a page of a foreign domain.

To improve the user experience, AJAX (Asynchronous JavaScript and XML) is employed throughout the user interface of the Annotizer. Using AJAX, we can maintain two way communications between the client and the server between page requests. For example, to run a search query, rather than having to refresh and return an entire page's content with each query, the server can return just the query results. The sending and receiving on the client side (browser) is managed through JavaScript. These requests are directed to small specific servlets which are part of the Annotizer. For instance, to update state

information, the client would make a request to the SessionServlet. The SessionServlet would query the database for all relevant information and send back an XML response. It is important that the type of response is specified in the root node of the XML response. Just like we have a default servlet on the server side, we have a default JavaScript function which determines what type of information is being sent, and calls the appropriate handler function to manage the response.

Using AJAX has both advantages and disadvantages. The advantages are that our application is extremely dynamic and responsive. Page information can be modified with nearly the same responsiveness as a desktop application. Additionally, we almost naturally separate the application view and control. In doing so, the page's visual design can be plain HTML without server-side script embedded throughout the page resulting in pages that can be visually modified easily without in depth programming knowledge. The disadvantages of using AJAX, or any approach based on JavaScript is portability. While a certain degree of compatibility exists between different browsers implementation of JavaScript, many features are either W3C specific or Internet Explorer specific. The end result is not quite as drastic as writing two versions of the JavaScript, but is not far off either. Additionally, many older browsers do not support AJAX, and will never be supported.

AJAX has not been the only source of difficulties in achieving full portability. In fact writing a cross-browser compatible wrapper for the JavaScript XMLHttpRequest object was reasonably easy. One of the biggest problems was writing a wrapper around Internet Explorer's JavaScript TextRange object. We chose to follow W3C standards as closely as possible, in hopes of minimizing issues from browser to browser. All browsers other than Internet Explorer follow this standard quite closely. While Internet Explorer does follow some of the standards, it does not implement all of them. The Range object is used extensively as a means of inserting and extracting text to/from annotated pages. The range object as a part of the W3C standards is not implemented in Internet Explorer. In order to support Internet Explorer, we had to write a wrapper around the TextRange object and implement all of the properties and methods of the W3C Range object. This task is being completed.

4. Status and Deployment of the Annotizer

Our current Annotizer supports text annotations on HTML documents, with image annotations being finalized (<http://ccls.sfsu.edu:8080>). We are about to launch the Annotizer as part of collaborative tools for the users of Hedgehog Signaling Pathway (HSP) WWW site (researchers, instructors and students) and we will follow with extensive usability study. Our annotations would be used by professors, researchers and students as a collaborative tool allowing them to annotate base HSP content either as a way of posing questions, or clarifications, or simply marking it for their own further study.

After the Annotizer is proven to be effective by the HSP community and several additional features explained in

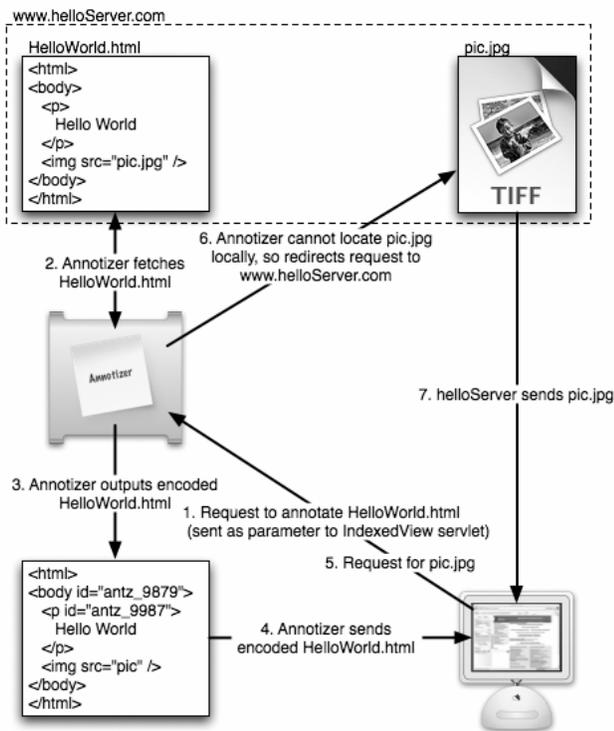


Figure 5. Annotizer Architecture, a page request sequence.

the future work are completely supported, then the Annotizer will be offered to a wider spectrum users who can build their own community around the Annotizer by sharing annotations on related web sites.

5. Conclusions and Future Work

We developed the second version of the Annotizer allowing asynchronous text annotations with central database architecture. We believe that such annotations offer a powerful way for Web based collaboration and content management in research and teaching. Innovations in this version of the Annotizer include ability to access any URL page and completely new UI with some new functionality. We are about to launch the Annotizer with HTML text annotation functionality as a collaborative tool with the currently popular HSP WWW site, to be followed by extensive usability study with its first users.

There are a number of future projects that will extend this initial work. We will automatically archive annotated pages to address the issues of orphaned or modified pages for which the old annotations would not work any more. Threaded discussion on annotation will be added soon. The content of private annotations will in the future be encrypted before being added to the database for full user privacy. New annotation functionality will allow users to establish and label links between content segments. Ultimately, annotation information and links can be mined to gain valuable information both for the users as well as content authors. When the number of user who will use the Annotizer like a portal to navigate WWW and develop

communities of common interest, then support for community needs to be considered. Finally, more comprehensive usability studies must be done to assess the ultimate value of this technology.

References

- [1] C. Marshall, "Toward an ecology of hypertext annotation," in Proceedings of ACM Hypertext '98, Pittsburgh, PA (June 20-24, 1998) pp. 40-49.
- [2] C. Marshall, M.N. Price, G. Golovchinsky and B.N. Schilit, "Introducing a Digital Library Reading Appliance into a Reading Group." In Proceedings of ACM Digital Libraries 99, ACM Press, pp. 77-84.
- [3] SPIE paper
- [4] J. J. Cadiz, Anop Gupta and Jonathan Grudin, "Using Web annotations for asynchronous collaboration around documents," in Proceedings of the 2000 ACM conference on Computer Supported Cooperative Work, CSCW 2000, Philadelphia, pp. 309 – 318.
- [5] Laurent Denoue, "Personal Information Organization using Web Annotations," WebNet 2001 World Conference on the WWW and Internet, Orlando, FL, January 17, 2001.
- [6] Polle T. Zellweger, Niels Olof Bouvin, Henning Jehøj, Jock D. Mackinlay, "Fluid annotations in an open world," Proceedings of the twelfth ACM conference on Hypertext and Hypermedia, August 14-18, 2001, pp. 9 – 18.
- [7] Scott A. Carter, Elizabeth F. Churchill, Laurent Denoue, Jonathan I. Helfman and Les Nelson, "Digital Graffiti: Public Annotation of Multimedia Content," CHI 2004, Vienna, Austria, April 24-29, 2004, pp. 1207 – 1210.
- [8] Web reference of Annotea, <http://www.w3.org/2001/Annotea>
- [9] Web reference of Annotilla, <http://annotation.semanticweb.org/Members/lago/AnnotationTool.2003-08-25.2532>
- [10] Handschuh, S, and Staab, S, "Authoring and Annotation of Web Pages in CREAM," Proceedings of the 11th International World Wide Web Conference, 2002, <http://annotation.semanticweb.org/Publications>.
- [11] Web reference of Wiki, <http://wiki.org/>
- [12] Web reference of Web Discussion, http://research.microsoft.com/~jicadiz/cscw2000_of_ficeAnnotations.pdf
- [13] Web reference of iMarkup, <http://www.imarkup.com>
- [14] Web reference of NekoHTML Parser, <http://people.apache.org/~andyc/neko/doc/html/index.html>