Development of Collaboration Environments for Web-Based Learning

Taro Yabuki and Hiroshi Sakuta
College of Science and Engineering, Aoyama Gakuin University, Japan
Email: {yabuki, sakuta}@it.aoyama.ac.jp

Abstract—In this study, a method to construct a collaboration environment is proposed. In this environment, users can annotate any web pages. Our method has advantages such as (1) the system requirement is adaptable, i.e. it is available on various operating systems and web browsers, (2) annotations can be made up with images, and (3) users can browse annotations of others, etc. Utilizing our system, users can create contents, harness collective intelligence, and get rich experience. Also, we will propose a way to utilize this method in web-based learning and how to enhance its efficiency.

Index Terms—collaboration environments, annotation, Ajax, web-based learning, e-Learning

I. INTRODUCTION

We will propose and implement a method to construct an environment for collaborations. In this environment, users can annotate any web pages. In this paper, “annotation” means information attached to web pages, e.g. notes, question and answers.

The environment has many features as follows:

1) The client software of our system is available on general modern web browsers which means, there are no rigid restrictions on user’s system environment, e.g. variation of operating systems and web browsers (see section V-B).
2) Annotations can be attached to anywhere on a target web page (see section III-A).
3) Our client program does not save a target web page in local disk. As a result, web pages that users view are always up-to-date.
4) Original web pages need not to be modified. (Some web annotation services force web pages to include a script for them.)
5) Annotation can be made up with images (see section V-D).
6) Users can browse annotations of others (see section III-C).
7) Users can reply to annotations of others (see section V-C).
8) Users can subscribe a web feed of annotations of a particular web page or annotations of a particular person (see section V-C).
9) The most interesting web page can be discovered (see section V-D).
10) The most active or valuable person can be discovered (see section V-E).

There are some software or services that provide us part of above features, e.g. WebMemo[1], noteit[2], ScrapBook[3], Internote[4], Diigo[5], MyStickies[6], mooos[7], Nico Nico Bookmark[8], Fleck[9], Stickis[10], and Google Notebook[11]. Their features are summarized in table 1 (Numbers in table 1 indicate features described above.)

<table>
<thead>
<tr>
<th>FEATURES OF WEB-ANNOTATION SOFTWARE OR SERVICES</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebMemo</td>
</tr>
<tr>
<td>ScrapBook</td>
</tr>
<tr>
<td>Internote</td>
</tr>
<tr>
<td>Diigo</td>
</tr>
<tr>
<td>MyStickies</td>
</tr>
<tr>
<td>mooos</td>
</tr>
<tr>
<td>Nico Nico Bookmark</td>
</tr>
<tr>
<td>Fleck</td>
</tr>
<tr>
<td>Stickis</td>
</tr>
<tr>
<td>Google</td>
</tr>
</tbody>
</table>

We can use these software or services in various situations. Especially, e-Learning is one of the most promising targets of our method. The way to apply the method for e-Learning will be discussed in section VI.

Usage of our method will be explained in section II. System configurations will be explained in section III. We will go into detail of our system behavior in section IV and our system implementation in section V.

These are general purpose software or services, i.e. we can use them in various situations. One of the most promising targets of our method is e-Learning. The way to apply the method for e-Learning will be discussed in section VI.

Some discussion will be arisen in section VII. We will present conclusions in section VIII.

II. SYSTEM OVERVIEW

In this section, basic behaviors of our system are explained.

A. Preparation

Before using this system, users have to register themselves with the system. In the middle of registration
process, users are provided two bookmarks by the system. These bookmarks have to be stored in user’s web browsers (figure 1).

Bookmarklet is a bookmark which address field is not a Uniform Resource Identifiers (URI) but a script written in JavaScript. Using bookmarks, user can start and stop a client program.

![Bookmarklets to show and hide the layer.](image)

**Figure 1.** Bookmarklets to start and stop a client program.

### B. Basic user interfaces

Suppose we are browsing a web page as figure 2.

![A sample web page (original).](image)

**Figure 2.** A sample web page (original).

Pressing the “Show” button of the bookmarklet, a menu bar for the system will be drawn on the web page (figure 3). Pressing the “Hide” button, the menu becomes invisible.

The menu is drawn on a newly created layer on the web page (figure 4). This layer is the most essential part of the user interface of our system.

![A schematic view of a newly created layer on the web page.](image)

**Figure 3.** The sample web page with a menu bar to control annotations.

**Figure 4.** A schematic view of a newly created layer on the web page.

### C. Text-based annotations

The most basic function provided by the layer is annotating a web page. Clicking “memo” button of the menu, a new panel for memorandum is created. User can write a plain text in this panel (figure 5). The written text is sent to an annotation server (explained later) and stored in a database. Even if the user quit the web browser, reopening the same web page and clicking “Show” button, previously created annotations are loaded from the server and restored at the same position. The relative positions of annotations are determined by their coordinates on the document. This is the second feature of our system described in section 4.

Clicking “others” button on the menu bar, user can read annotations written by others. This is the sixth feature of our system described in section 4.

The client program is implemented as Asynchronous JavaScript + XML, Ajax[12]. Therefore, during the operation described above, data transformations between client and server are performed asynchronously in background. In addition, there are no page transitions invoked by the operations.

### D. Graphic-based annotations

In addition to plain text, we can use handwriting images in annotations. This is the fifth feature of our system.

---

1 Although figures are screenshots of Safari on Mac OS X, the client program can be executed on almost all general modern browsers, e.g. Firefox, Opera, Internet Explorer, and Konqueror.

2 Although ‘X’ in Ajax means XML, data format used in our system is not XML but JavaScript Object Notation, JSON. This difference, however, is not essential.
described in section 1. In an example of figure 7, the user can choose color and thickness of brush strokes.

III. SYSTEM CONFIGURATION

In this section, we explain the system architecture.

Figure 8 shows the way to browse simple web pages written in HTML, and figure 9 shows it using our system.

The script located in center of figure 9 takes the major role in our system. This script sends the user ID and the URI of the web page browsing to the annotation server. If there are annotations about the web page stored in a database, then annotation server returns them to the script. Returned annotations are displayed on the web page—to be exact, the layer on the web page—by the script. As a result, users are able to read those annotations.

The annotations displayed on the web page and the annotations stored in a database are always synchronized by the script running in background. Therefore, if someone who browses the same page adds an annotation, then the added annotation appears on the layer immediately.

IV. SYSTEM BEHAVIOR

Figure 9 is a sequence diagram of the system. The action sequence of the system is as follows:

1) User loads a web page from a web server.
2) Using a bookmarklet (“Show” button), the user creates a layer on the web page browsing.
3) The script embedded in the layer sends the user ID and the URI of the web page to the annotation server. If there are annotations about the web page, then they will be returned to the script and displayed on the layer. This process does not interrupt the user, because it is executed asynchronously in the background.
4) The script continually checks whether annotations
stored in the database are altered or not. If they are updated, then the script downloads the change from
the server and updates the contents of the layer. 5) User can add, alter and delete the annotations.
6) Results of user’s operations are sent to the server and stored. Data transmission does not interrupt the
user.
7) Pressing the "Hide" button, the layer becomes invisible.

Figure 9. Sequence diagram of the system.

V. IMPLEMENTATION DETAILS

The system consists of two components.

1) Client-side component: Consists of bookmarklets and HTML pages. Both of them involve a JavaScript code. There may be no other easy choices to implement.

2) Server-side component: Consists of a web server and web services running on it. We chose Apache HTTP Server for the web server and PHP for the language to implement the web services. However, they are not the only choices.

A. Bookmarklet

Bookmarklet is a kind of Bookmarks or Favorites of web browsers. Normal Bookmarks hold a URI of a web page. On the other hand, bookmarklets hold a script written in JavaScript. They are part of the client-side component. We can operate web browsers by this script. Bookmarklet of the client-side component sends the user ID and the URI of the web page to the annotation server and allocates a space for the layer returned from the server. To allocate a space, the script adds an iframe element into the web page.

B. Inline frame element

Inline frame element is an element to insert another page into a document. The bookmarklet makes absolute position of inserted page to be 0px from top left, the page size to be as same as the one of the documents, and z-index—the stack level—to be greater than one of the documents. As a result, the inserted page becomes an upper layer of the original document (see figure 11).

Both bookmarklets and inline frame elements are supported by all modern web browsers. Accordingly, our system is available on them. This is the first feature of our system described in section II.

C. Data format for transmission

As described above, client-side script starts as a part of the web page being browsed. The script must be able to transmit an annotation between the client and the annotation server. Data transmission cannot be realized by means of a simple JavaScript command like XMLHttpRequest, because it cannot request to any server whose domain is not the same as the domain where the original web page comes from.

We can solve this problem by the method called JSON with Pudding, JSONP[13]. JSONP is a data transmission method to send and receive data by utilizing script tags. Data format of the received data is JavaScript Object Notation, JSON[14]. JSON can be used easily in not only JavaScript but also other programming languages, because it is a plain text data.

To send data from client to server, the client-side script creates a script elements whose type attribute is "text/javascript" and src attribute is "http://(server-side application name)/data=(percent encoded JSON string)".

To send back data from server to client, the server returns a string like "func(JSON string)". Becoming the contents of the script element, the string is executed. The callback function (func) must be implemented in advance.

VI. APPLICATIONS TO E-LEARNING

The applications of our method are expected to be used especially in e-Learning circumstances. In this section, we introduce the way to utilize our system in e-Learning.

A. Digital note

In using web pages as teaching materials, one of the most serious problems is that we cannot take notes in web pages. Due to this problem, learners had to print out the web pages or use notebooks. These solutions reduce the advantage of online materials that they are easy to update and decrease the efficiency of learning (see figure 15).

This problem is solved by utilizing our method, because the most basic function of our system is to take notes on web pages.

This may increase the server's workload and could be problematic when the number of clients is large.

© 2007 ACADEMY PUBLISHER
B. Questions and answers

In ordinary e-Learning systems, it is common to learn by web materials and utilize discussion boards (see figure 10). If learners want to take a look at discussions on the discussion board, then they must first find out the corresponding point in teaching materials. The cause of this problem is that teaching materials and the discussion board are separated.

Utilizing our system, it becomes possible to ask questions and get answers on web pages as follows:
1) Attach a question to an annotation.
2) As described in section 11 user can read annotations of others. Therefore, teachers and other learners can receive the question and reply to it. Using the RSS feed (explained later), they can receive the question without opening the corresponding web page.
3) The answer is an annotation too so, if an answer is submitted and stored in the database, then it is downloaded by the client-side script and displayed on the questioner’s web browser.

This is the seventh feature of our system described in section 11. It may be possible to answer questions automatically by introducing full text search engine and searching answers from existing annotations. Collecting feedback from user, the value of the answer can be determined.

To sum up, our system makes a paper notebook and discussion boards needless (figure 10).

(a) Ordinary e-learning systems need paper notebooks.

(b) Ordinary e-learning systems need a BBS.

(c) Utilizing our system, we can make notes and ask questions just on the materials.

Figure 10. Paper notebooks and discussion boards become needless.

C. RSS feed

It is convenient to receive annotations by RSS feed, because users do not have to open web pages to read annotations (figure 11). There are two useful RSS feeds for e-Learning (This is the eighth feature of our system described in section 11).

1) User’s annotations: If user can get a grasp of their annotations, then they need not to record the web page separately. By subscribing the RSS feed of student’s annotations, teachers can check student’s degree of progress.

2) Annotations about a particular web page: Teachers can utilize the annotations about their teaching materials as feedback. It is possible to read the annotations by opening the web page however, RSS feed is more convenient, because it is pushed to user automatically. If teachers subscribe the RSS feed of questions from students, then they can answer to it rapidly.

D. Interesting web pages

Web pages attached many annotations can be regarded as interesting. This is the ninth feature of our system described in section 11. This feature is realized by many social bookmark services. Social annotation systems like ours are social bookmark services in the sense that URI of a web page is stored in a center server.

E. Active or valuable people

People who make many annotations can be regarded as active. If their annotations are highly appreciated, they can be regarded as valuable. This is the 10th feature of our system described in section 11. This may be a powerful incentive for students to use our system and make good annotations or answers to questions.

VII. DISCUSSION

Constructing the environment for collaborations, we can embed some functions in web pages as follows:
• Making user able to create contents (see section [6.C].
• Harnessing collective intelligence (see section [6.D] and [6.E].
• Providing rich user experience (see section [6.C].

These functions are a portion of features of Web 2.0 presented by Tim O’Reilly in 2005[13]. In recent years, World Wide Web has developed its ability vastly by introducing these features. To our regret, we have not been able to enjoy these features unless web page creators have implemented them in advance. Using our method, however, we can embed these features in web pages without modifying it, even if web page creators have not implemented them.

VIII. CONCLUSIONS
We proposed and implemented a method to construct collaboration environments and provide Web 2.0 features to any web pages. Provided features are as follows:
• Annotating web pages.
• Browsing annotations of other people.
• Creating discussion board wherever users like.
• Getting feeds of annotations.
• Discovering interesting web pages and active or valuable people.

Applying our proposed system to web-based learning:
• Text editor or paper notebook and discussion boards become needless.
• Students can pose questions and get answers whenever they like.
• Teachers can check student’s degree of progress.

Proposing applications to promote collaboration on our system is the challenges for the future.

REFERENCES


Taro Yabuki received his PhD, MS and BS degrees in science from the University of Tokyo, Japan, in 2004, 2001, and 1998, respectively.
He is currently an assistant professor at Aoyama Gakuin University, Japan. His research interests include interactions between human with complex systems.

Hiroshi Sakuta received his PhD degree in engineering from the University of Tokyo, Japan, in 1979.
He is currently a professor at Aoyama Gakuin University, Japan.