REAL-TIME DISCOVERY OF CURRENTLY AND HEAVILY VIEWED WEB PAGES

Kazutaka Maruyama
Information Processing Center, The University of Electro-Communications
Chofugaoka 1-5-1, Chofu, Tokyo, 182-8585, Japan
kazutaka@acm.org

Kiyotaka Takasuka, Yuta Yagihara, Satoshi Machida, Yuichiro Shirai, Minoru Terada
Dept. of Information and Communication Engineering, The University of Electro-Communications
Chofugaoka 1-5-1, Chofu, Tokyo, 182-8585, Japan
{*takasuka,yagihara,machida,gechena,terada*}*@ice.uec.ac.jp*

Keywords: Peer-to-peer, clustering, web browsing, browser extension.

Abstract: The amount of information on web increases explosively, so it is difficult for web users to find web pages they want. There are some approaches to resolve this problem, such as semantic web which make web information systematic, the improvement of search engines' algorithm, and so on. Dealing with web as a huge database, these technologies works well, however, they cannot provide any useful solutions to get hot news which expands quickly to the world because of their time lag.
In this paper, we propose a system whose users can know currently and heavily viewed web pages. The key features of this system are as follows: (1) to find hot news in web, (2) to provide recommendations to users without any content analysis, and (3) to apply the system to other communication tools like IM as their infrastructure to find appropriate contact targets. We describe our policy of the system implementation and show the result of a pilot experiment with a pilot implementation.

1 INTRODUCTION

In the recent web, the amount of information on web increases explosively, for example consumer generated media like blog, and the pace seems to be accelerated much more in future. Users know that web pages they want to find exist somewhere in web world, but they can find only part of the target pages and always worry about the lack of useful ones.

Two major approaches to this problem are semantic web and the improvement of search engines. Web contents with rich meta data and search engine algorithms such as PageRank can produce more appropriate answers to users' queries. However these solutions have three problems. First, it is considerably difficult to add appropriate and sufficient meta data to all existing web contents, and also probably all ones generated in future. Second, the intervals of crawling by search engine robots and of updating the indexes do not become much shorter and are not in real-time. Third, the solution cannot create any clusters of the users, not the contents. We call them *communities*. In real life, communities exist everywhere such as in schools, offices, web bulletin boards, and so on. We can always talk with each other about various hot topics in such communities. For instance, a person who knows music scene thoroughly may tell us about not only recent trends of CD charts, but also a rumor about unpublished next iPod. Search engines provide results only to explicitly given terms by users, but we cannot find other topics which implicitly relate to the terms because they have no user communities.

We propose an infrastructure to find currently and heavily viewed web pages in real-time by exchanging viewed URLs at the client side. Three advantages of this architecture are as follows.

1. We can quickly discover web pages which are attracting considerable attention, such as a new beta service of Google disclosed tonight.
2. We can know web pages which most of *friendly* users view. The term of friendly users means ones who usually view the same web pages as we have viewed.
3. Communication tool developers or researchers can use our system as an infrastructure of new tools they are developing, such as IM, because it creates communities of users who usually view the same web pages, and probably have the same preferences.

Since our system does not need any meta data, all the existing web contents in the world are available and the content providers and consumers can use the system without any special actions (the first problem above is solved). The system provides currently and heavily viewed web pages mostly in real-time, because exchanging URLs need not crawl or index any web pages (the second is solved). In addition, it can very easily introduce the mechanism correspondent to mouth-to-mouth advertising in real life by creating user communities with only usual actions of web users, that is to view web pages in their browsers (the third is solved).

The remainder of this paper is organized as follows. Section 2 shows the features of our proposal with an use case scenario. In section 3, we discuss the losses against gains of the implementation strategies of the client and server side subsystems, and describe the implementation of our currently developing system. Section 4 shows the result of a pilot experiment with a pilot system. Other applications of our system are described in section 5 and the related works are discussed in section 6. Section 7 and 8 describe conclusion and future works respectively.

2 FEATURES OF OUR SYSTEM

In this section, we show the features of our proposal in the view point of users and the current user interface through an use case scenario.

2.1 An Use Case Scenario

Alice is a very common user of web, every day crawls her favorite news sites and blogs, and looks for a reputation of a restaurant which she found on her way home by using search engines. She changed her jobs recently and was not satisfied because she could not find interesting topics in web well, which a well-informed colleague in the former office told her, through news sites, blogs or search engines. She could not find news sites or blogs just suitable for her interests and was tired of reading so many RSS feeds. Search engines are very powerful tools when she is looking for topics with explicit purpose, but they do not help her to look for recently popular topics or somehow nice events.

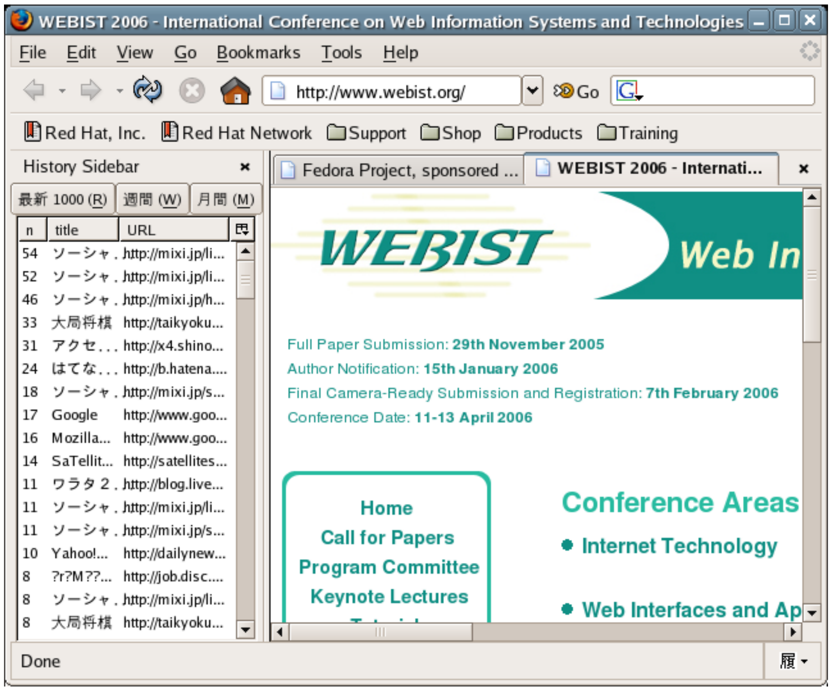
One day, she began to use a system mentioned in a certain blog. It was said that users of the system exchange URLs they are just viewing. The information exchanged is only URL, so she did not have to take care of copyright violation in contrast to file sharing. Following the instruction of the official site of the system, she installed a tiny extension for Mozilla Firefox, an add-on program for the web browser, to her own Firefox. She clicked an icon added by the extension at the lower right of Firefox, then a pop-up menu appeared there. It controls the appearance of the sidebar described below, and enables or disables the mechanism of sending out URLs she is viewing. Turning on the sidebar, she saw the appearance of her Firefox such as figure 1. The sidebar lying on the left side of the window has a few buttons to list the correspondent rankings, and clicking these buttons causes switches of the list displayed in the pane. Entries of blogs or diaries in social network services viewed heavily were listed at the top of the lists. Alice attempted to participate in the project of exchanging URLs and enabled the mechanism of sending out URLs by the menu of the lower right icon. Therefore URLs she is viewing became to be shared in the world.

One week after, URLs listed in the sidebar became suitable for Alice's preference. The project web page said that the system connects users to each other with fewer hops, who tend to view the same web pages, and consequently provides more appropriate rankings. She viewed web pages in the same way as before, but she became to know some cool products which she had never known or some recipes of healthy diets. Using the new way of retrieving information different from news sites, blogs or search engines, she became to use web more efficiently and had fun with various hot topics.

2.2 Summary of Features

We described a basic use case of our proposal through Alice's experience. The summary of the features is as follows.

- A quite new channel for information retrieval is established. In real life, for instance, a crowd on a street shows that there is something interesting such as a street performer. While the same method of information retrieval did not exist so far, it is introduced into the world of web.
- The proposed system provides real-time information. Users need not to wait for a blog entry to be submitted by the owner who saw a news source. For example, we can reach an entry of Slashdot, to which comments are increasing.
- Communities are created. The server side system connects friendly users to each other with fewer hops by using URLs they have viewed. In other words, unfriendly users are kept away from each other and S/N ratio of the provided rankings would be improved.
- All users have to do is an installation of a single extension once, in contrast to social bookmarkings which need their users to continue submitting each entry for exchanging URLs. The users of our system can provide useful information of web pages only by viewing web pages in their usual way and can get useful URLs from others in the world.


WEBIST 2006 - International Conference on Web Information Systems and Technologies
File Edit View Go Bookmarks Tools Help
http://www.webist.org/
Go
Red Hat, Inc. Red Hat Network Support Shop Products Training
History Sidebar
最新 1000 (R) 週間 (W) 月間 (M)
Fedora Project, sponsored ...
WEBIST 2006 - Internati...
WEBIST
Web In
Home
Call for Papers
Program Committee
Keynote Lectures
Conference Areas
Internet Technology
Web Interfaces and Ap
Done

n	title	URL
54	ソーシャ .	http://mixi.jp/li...
52	ソーシャ .	http://mixi.jp/li...
46	ソーシャ .	http://mixi.jp/h...
33	大局将棋	http://taikyoku...
31	アクセ . . .	http://x4.shino...
24	はてな . . .	http://b.hatena....
18	ソーシャ .	http://mixi.jp/s...
17	Google	http://www.goo...
16	Mozilla...	http://www.goo...
14	SaTellit...	http://satellites...
11	ワラタ 2 .	http://blog.live...
11	ソーシャ .	http://mixi.jp/li...
11	ソーシャ .	http://mixi.jp/s...
10	Yahoo!...	http://dailynew...
8	?r?M??...	http://job.disc....
8	ソーシャ .	http://mixi.jp/li...
8	大局将棋	http://taikyoku...

Full Paper Submission: **29th November 2005**
Author Notification: **15th January 2006**
Final Camera-Ready Submission and Registration: **7th February 2006**
Conference Date: **11-13 April 2006**

Figure 1: Alice's Firefox with our extension.

The use case described in this section includes the functions which are not implemented yet, but shows the user experience through our proposal. Another aspect of the system, an infrastructure of communication tools, is described in section 5.

3 SYSTEM STRUCTURE

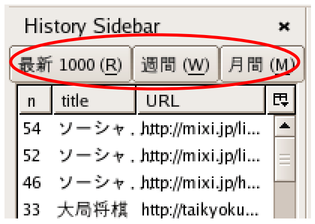
In this section, we describe the implementation strategies of the client side and the server side subsystems, discuss the losses against gains of the strategies, and describe the current system implementation.

3.1 Extension vs. Proxy

One of the important behavior is to capture URLs an user is viewing in real-time without any interferences. There are two implementation choices: (1) add-on program for web browsers such as extension or toolbar, and (2) web proxy server.

The best way for users is web proxy. All users have to do is to set up their browsers so that the proxy is used, then all the web behavior of the users is captured by the proxy in real-time. The proxy would be placed at school or office, or be installed as a quite small Java program without cache facility into each user's PC. The greatest benefit of this choice is that the installation procedure of users is the minimum. When a proxy is used by many users, for example for all students in a college, the single client side subsystem can capture so many users' web behavior. However, this solution has a disadvantage of collecting too many meaningless URLs. When an user clicks a certain link, his browser, and the proxy of course, must retrieve many resources in the clicked document, such as images, style sheets, flash applications including advertisements. The ranking of captured URLs will place the Google logo image at the top of the list. Of course such the result is not expected. Using Referer fields in HTTP or analyzing transferred HTML, clicked images may be distinguished from ones included clicked web pages. But the analysis makes the system complex and the performance down, consequently, the users must become irritated.

Another implementation choice is extension or toolbar, which have the opposite features. These add-on programs work as a part of a web browser and can capture explicitly clicked URLs only. This feature is an obvious advantage against the proxy. On the other hand, the extension has a few disadvantages. First, users have to install the extension or toolbar, even if it is a very small program and only two or three clicks finish the installation. Second, at least two implementations for Firefox and Internet Explorer are required in order to support over 90% of web browsers in the world. Since a Firefox extension works on all platforms supported by Firefox, such as Windows, Linux,



n	title	URL
54	ソーシャ .	http://mixi.jp/li...
52	ソーシャ .	http://mixi.jp/li...
46	ソーシャ .	http://mixi.jp/h...
33	大局将棋	http://taikyoku...

Figure 2: Three buttons of the sidebar.

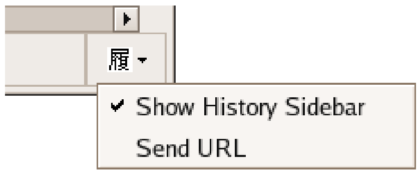


Figure 3: Pop-up menu.

Mac OS X and so on, a toolbar for Windows IE and the extension can cover almost all the client environment.

As described in section 2, we chose the extension/toolbar implementation, but the toolbar for IE is not implemented yet. The appearance of Firefox with the extension is shown in figure 1. In the current version, the sidebar on the left of the window has three buttons (figure 2), and three rankings appear by the correspondent buttons, i.e. recent top 1000, weekly chart and monthly chart. An additional ranking of URLs to which accesses increase suddenly would be added. Figure 3 shows the pop-up menu from the icon at the lower right of the window. Sending URLs is disabled in the default setting.

3.2 Server Centric vs. Peer-to-Peer

The extension is the client side half of our system. The server side one collects URLs sent from the client side. There are also two choices: (1) server centric and (2) peer-to-peer.

Obviously, the server centric approach is simpler. Powerful servers with high speed CPUs and large disks in a data center could deal with URLs sent from all users in the world. For example, Google and Yahoo! start their services which save the search histories of each user. They must use such many servers and disks. In this approach, it is easier to exchange users' web behavior and to analyze relevance among users. On the other hand, the disadvantage is that the large system of many servers is too expensive and requires a large amount cost for maintenance.

Another approach is P2P. The users install a tiny program running on background written in Java into their PC and the program works as a node of the P2P system. The client side programs, extensions or proxy servers, send URLs to and receive rankings from the node. The large system need not be constructed. However, a simple and flat P2P network such as Gnutella cannot deal with so many URLs from so many users in the world and cannot connect friendly users to each other. Clustering P2P nodes and a layered architecture, such as Skype(Skype - The whole world can talk for free, nd) or Winny(Kaneko, 2005), could resolve this problem.

At present, we choose the server centric approach because of the ease of development. The server consists of some CGI programs, the extension sends URLs to it as POST data of HTTP and receives the rankings from it by GET method of HTTP. Note that there is an explicit interface between the client side subsystem and the server side one. The interface is defined as URLs and CGI parameters over HTTP. When we replace the server with P2P based subsystem, if the interface between the extension and the server is preserved, no changes are required for the extension. In addition, the interface of HTTP is extremely suitable for web browser extensions.

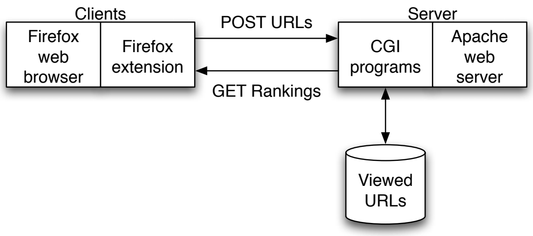


Figure 4: Current system structure.

3.3 Current System Structure

The current system structure is shown in figure 4. The client side subsystem consists of Firefox web browser and the extension and the server side subsystem consists of Apache web server and some CGI programs written in Perl. The extension sends clicked URLs to the server by POST method and the server stores them into disks. When the user clicks a button of the sidebar shown in figure 2, the extension sends a GET requests to the CGI program correspondent to the clicked button, then receives the list of the ranking.

The system produces any explicit recommendations of web pages, because the ranking of the recent top 1000 shows the recommendations from users of the system. The weekly and monthly charts show the recommendations from them in the week and the month respectively. The ranking of URLs to which accesses increase suddenly, which is still developing,

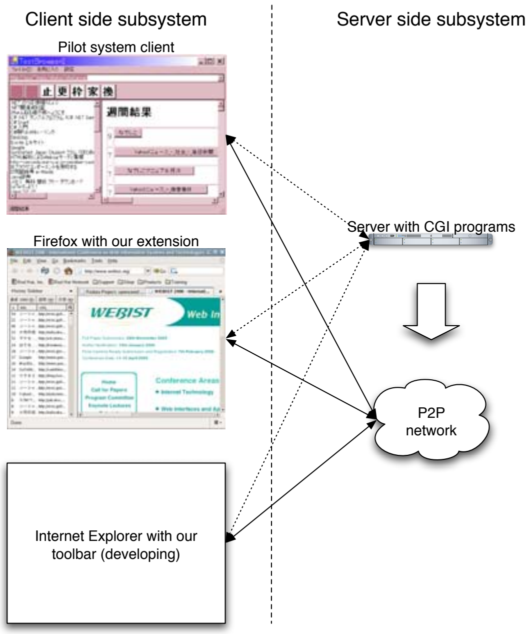


Figure 5: Three kinds of clients and two of servers.

shows the recommendation at the moment.

Prior to developing the extension, we developed a pilot implementation of the client side subsystem, the top in figure 5 (Takasuka et al., 2005). In fact, the server side subsystem was developed for the pilot client. The current client, the middle in figure 5, preserves the interface between the pilot client and the server, and replaces the pilot one without any changes of the CGI programs. The coming toolbar of IE shown in figure 5 would work together these existing clients.

Clustering users are not introduced yet into our system. First, users of our system are still around ten. Because the extension has some problems, our system is not released yet to the outside of our laboratory. Second, the clustering is relevant to the identification of URLs. This point is described in section 8.

4 PILOT EXPERIMENT

In order to examine the usefulness of our proposal, we implemented a pilot system to do a pilot experiment. In the pilot system, a client side subsystem is a simple and tiny web browser written in C# with IE component on Windows (shown in figure 5), and a server side subsystem consists of the CGI programs described above. The simple browser has forward/backward buttons, a simple bookmarklet and a facility of sending URLs to the server side subsystem.

In this experiment, nine student users used the pilot system for four days. The total accesses from the browser to web pages were 448 times. 70 accesses in them were from the rankings of the simple browser. Table 1 shows the result of the questionnaire. Each item is rated in one to five, one means the best and five the worst. The usefulness of the system we propose is evaluated comparatively highly. In addition, note that the rate of the psychological barrier in sharing web history is neutral. While the experiment is really small, the proposed system was accepted favorably, and the increase of the number of users would make the benefit more and the barrier lower.

5 OTHER APPLICATIONS OF OUR SYSTEM

While we do not implement the facility to connect friendly users yet, once the facility is introduced, it could be used as an infrastructure of communication tools. Two application examples are described below.

The first is a chat among the users who are viewing the same web page. In contrast to Instant Messenger, of which the users designate their partners to talk to in advance, the chat we propose connects users who are viewing or have been viewing the same web page and helps them to talk to each other about the web page or some topics about it. In web bulletin boards such as Slashdot, the users who have never met before can talk to each other, but the place for the discussion must exist already and they cannot know whether the user who wrote a certain comment is viewing the page now. Our proposal provides the interchange between the web users through ordinary web pages. The users need not look for a thread, in which they are interested, and can talk to friendly users only by viewing their favorite web pages usually. Improving the chat system, as IM users can know that their friends have logged in, the users could know the others are viewing a certain web page. The improvement makes web pages rendezvouz.

The second is a P2P based bulletin board system. It is difficult for P2P based BBSs to gather friendly users in the same thread because of the structure of P2P. However, using our system as its infrastructure, the P2P based BBS could gather many users, who tend to view the same web pages and may have the similar preferences.

The core application is the rankings extension described in section 3, which sends data to web history exchange layer in order to connect friendly users to each other and receives the rankings from the layer (figure 6). The other two applications described in this section only receive location information of friendly users from the layer.

Table 1: Evaluation of the pilot system.

Questions (1:good, 5:bad)	Av.
Serviceability of the browser	3.1
Usefulness of sharing web history	2.0
To watch URLs others view is helpful	1.7
There is no psychological barrier in sharing web history	2.8

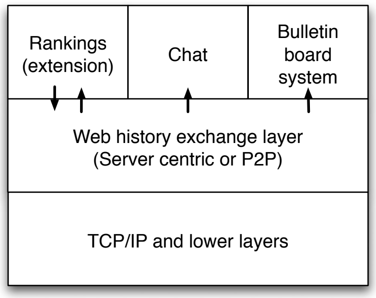


Figure 6: Web history exchange network as a basis of communication tools.

Other projects of communication tools would use the layer as their infrastructure. Whenever a quite new network service starts, there are two problems, the scalability of the server and the gathering of the sufficient initial users. Our system is useful for new network services or communication tools. Singh et al.(Singh et al., 2001) said that P2P networks can provide a substrate for community-based service location. The system we propose just can be applicable to it.

6 RELATED WORKS

Webmemex(Alaniz et al., 2003) provides the web page recommendations based on sharing web behaviors among the well-known users. They register each other in their contact list of Yahoo! Messenger, probably also well-known in real life. The proxy server gathers the viewed URLs and the back end system computes the relevance among the web pages by HTML analysis. When an user clicks a certain link, the system recommends some web pages which are related to the clicked web page and are viewed by other users of the group. It is similar to our proposal, but there are some essential differences.

- Our system deals with only URL strings, not the contents of the web page. The vocabulary in the web world is changing very quickly, so it is too difficult for such the analysis based system to continue following the changes, especially in some languages which do not place a space between the words such as Japanese. Since the data format used in the web world is also changing quickly, it seems not to be realistic to analyze the various contents of the web pages in the various areas.
- We aim for the discovery of the web pages which are attracting considerable attention with little time lag. The computation of the relevance of the contents among the web pages causes the opposite result to the purpose and it is suitable for post-processing. It should be dealt with as a different challenge.
- Our system does not limit the range of exchanging URLs. Limiting the range to the well-known users leads to the consequence of losing chance of coming across information in different areas. In addition, since the well-known users strongly connect to each other both in real and virtual life, their psychological barriers avoid exchanging URLs.

Social bookmarking services such as del.icio.us(del.icio.us, nd) aim for the similar goal. When an user finds others who bookmarks some web pages in which the person are also interested, the person implicitly gets their recommendations of hot topics or good web pages through their bookmarks. The first disadvantage is that the social bookmarking users have to explicitly go into action to add the URLs they prefer. The second is that the users looking for hot topics or good web pages have to look for friendly users too.

Fast el al.(Fast et al., 2005) introduce a social networking in order to cluster the nodes of P2P file sharing based on the downloaded music file categories of the P2P users. In comparison between P2P file sharing and P2P URL sharing, search queries in file sharing correspond to advertising of URLs in URL sharing, and downloaded files correspond to really viewed URLs by the users who receive the others' advertisements. The number and frequency of the transmission between the queries and the advertisements seem to be considerably different, and URL sharing mostly deals with statistical information, for instance many users are viewing a certain URL, in contrast to file sharing which finds the real entities, music files. The different knowledge from the paper about P2P clustering would be found.

Winny(Kaneko, 2005), a P2P file sharing software with over 200,000 users in Japan, uses three keywords for a background file downloading process. The keywords are also used for P2P clustering, but the users have to set the keywords in advance. In the Winny network, the files opened to be shared by a certain node is cached at the other nodes, such as Freenet(Clarke et al., 2002), and the network load is heavy. Therefore, in order to realize more efficient file sharing, Winny constructs the layered P2P network based on each node's connection speed, for example, up to 64kbps, xDSL or fiber. The P2P URL sharing distributes much smaller data such as URLs, so it have to make much of whether the reboot intervals of the nodes are long or whether the nodes have a fixed global IP address, rather than the connection speed.

7 CONCLUSION

Exchanging URLs which users are viewing in real-time brings hot topics, which are attracting considerable attention, without any keywords explicitly designated. The effective way to realize it is that the client side subsystem should be implemented as an add-on program for web browsers, and the server side as P2P. In the current version of our system, the server side is implemented as the server centric CGI programs, but the P2P could take the place of it by preserving the interface, which consists of HTTP and CGI, between the server side and the client side. The pilot experiment showed the usefulness of our proposal. In addition, the system for exchanging URLs could be applied to other communication tools as their infrastructure.

8 FUTURE WORKS

The top pages of portal sites tend to be placed at the higher rank. This problem could be resolved through the user interface of the extension. The users disable the appearance of such entries in the rankings, for example by a right click on a target entry, and exchange the *disabled* portal sites via the server side subsystem.

The strategy of clustering users is quite important and difficult. Creating clusters is an effective way to reduce the number of connections, the traffic on P2P network, and the computation of the similarity of users. The separation of clusters, however, may make users lose opportunity to be offered interesting web pages(Linden et al., 2003). In ordinal recommendation systems such as e-commerce marketing systems, each user's purchases in a day may be usually less than ten items, even if they are heavy users of the system. But web pages each user views in a day are at least ten pages, and heavy web users view more than 100 pages. In addition, there are web users obviously more than e-commerce site users. Reduction by clusters must be introduced for effective exchange of web histories. Parameters for establishing clusters, such as the number of connections of each node, the limit of hops of web history transfer, and so on, could be found through simulation by using huge amount of proxy logs of our university.

The identity of URLs is also important. While CMSs generate different URLs to each blog entry, known as *permalink*, visitors can read the entries not only in their permalinks but also in the summary pages of recent entries, of the day, of the month and so on. Consequently, a certain entry has many URLs where it can be read. It is desirable that these URLs are dealt with as the same. This problem could be resolved by the introduction of scores between similar URLs. The URLs which indicate the same entry are similar, because most CMSs add the date of the entry to its permalink. In addition, the similarities between URLs would help us to cluster users. Users who often see different entries of the same blog should be placed in the same cluster.

In the view of the privacy, it is important not to send URLs which should not be shared. Our extension does not send any URLs in the default setting and explicitly shows its setting, whether sending URLs is turned on or off, by the correspondent icons. On the other hands, the web pages without appropriate access controls, such as password authentications or `Limit` directives of web servers, must be seen by outsiders. It is equivalent to being opened even if the URLs are not opened. This problem is known as "Google hacking" among security experts. Appropriate settings of access controls would make the problem trivial.

Spammers may attack this system by sending URLs which they want to advertise. This problem is known as "shilling attacks". However, since the information for clustering is URLs themselves, the spammers may be classified to a spammers' cluster and cannot influence the ordinal users. The clustering of spammers would be inspected through simulation.

REFERENCES

Alaniz, A., Truong, K. N., and Antonio, J. (2003). Automatically Sharing Web Experiences through a Hyperdocument Recommender System. In *Proceedings of the 14th ACM Conference on Hypertext and Hypermedia*, pages 48–56.

Clarke, I., Miller, S. G., Hong, T. W., Sandberg, O., and Wiley, B. (2002). Protecting free expression online with Freenet. *IEEE Internet Computing*, 6(1):40–49.

del.icio.us (n.d.).
http://del.icio.us/.

Fast, A., Jensen, D., and Levine, B. N. (2005). Creating Social Networks to Improve Peer-to-Peer Networking. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pages 568–573.

Kaneko, I. (2005). *Technology of Winny*. Ascii Publishing. (In Japanese).

Linden, G., Smith, B., and York, J. (2003). Amazon.com Recommendations: Item-to-Item Collaborative Filtering. *IEEE Internet Computing*, 7(1):76–80.

Singh, M. P., Yu, B., and Venkatraman, M. (2001). Community-based Service Location. *Communications of the ACM*, 44(4):49–54.

Skype - The whole world can talk for free (n.d.).
http://www.skype.com/.

Takasuka, K., Shirai, Y., Maruyama, K., and Terada, M. (2005). A web browser that shares browsing histories. In *FIT2005*, pages 355–356. (In Japanese).