ヒューマンインタフェース 71－4
（1997. 3. 6）

Internet Scrapbook: Creating Personalized World Wide Web Pages

杉浦 淳　小池 雄一　古関 義幸
NEC C&C研究所
川崎市宮前区宮崎4-1-1
E-mail: {sugiura, koike, koseki}@mmp.cl.nec.co.jp

現在 WWW (World Wide Web)上には多種多様の情報が提供されている。しかしながら、WWW からユーザが必要とする情報のみを取得するためのコストは小さくない。本稿では、WWW 情報のパーソナル化のためのシステム Internet Scrapbook について述べる。本システムでは、WWW ページからユーザが必要とする箇所のみを切り出し、一つの個人用ページにまとめることができ、また、作成した個人用ページの内容を自動的に最新情報に更新することが可能である。これにより、ユーザは自分の必要とする情報のみを効率よく閲覧することができる。

Internet Scrapbook: Creating Personalized World Wide Web Pages

Atsushi Sugiura　*Yuichi Koike*　*Yoshiyuki Koseki*
C&C Research Laboratories, NEC Corporation
4-1-1 Miyazaki, Miyamae-ku, Kawasaki 216, JAPAN
E-mail: {sugiura, koike, koseki}@mmp.cl.nec.co.jp

This paper describes an information personalization system, called Internet Scrapbook, which enables users to create a personal page by clipping and merging their necessary data gathered from multiple Web pages. Even when the source Web pages are modified, the system updates the personal page, replacing with the latest data extracted from the source pages. Therefore, once a user creates a personal page, the user can browse her necessary information only.

1. INTRODUCTION

WWW (World Wide Web) browsers, such as NCSA Mosaic [8], Netscape Navigator [9] and Microsoft Internet Explorer [6], allow end-users to access Internet information resources. While the operations on those browsers are simple, the users are required to spend much time and care on the daily access for the user desired Web information, by the following reasons:

- The users usually need to browse a portion of a Web page. On a countrywide weather forecast page, for example, the user wants the forecast only for his/her resident area. So, the users are often required to search the page for the desired information either by eyes or by using string search function provided by the browser.
- Usually, the necessary information exists in several different pages. For example, the user might need a weather forecast, cyber news, the weekend sports result, etc. from different pages. In such cases, the users have to repeat operations of specifying URLs (Uniform Resource Locators) and searching the downloaded pages for the necessary information.

If the Web pages are frequently modified, repetitive access to those pages is a heavy burden for the users. Our goal is to enable users with little programming skill to automate the daily Web browsing tasks and browse only the necessary information with minimum efforts.

This paper describes an information personalization system, called *Internet Scrapbook*, which allows users to create a personal scrapbook page by clipping and merging only their necessary data from Web pages. Our approach to achieve the goal is based on an example-based or demonstration-based programming technique[1], that is, a user demonstrates the objective task on example data and a system generates a program corresponding to the user demonstration to execute the task on behalf of the user. In Scrapbook, the operations of creating the personal page are the user demonstration. The system records locations where the user clips from the source Web pages and generates matching patterns for determining the locations, indicated by the user, in the source pages even when the pages are modified on the Web sites. Since the system automatically updates the personal page by re-constructing it with the latest information, the user can avoid repetitive operations of specifying the URLs and searching for the information.

In the following sections, we first mention related work and next explain main facilities of Internet Scrapbook. After showing experimental results in terms of updating the personal scrapbook pages, we finally have a section for the discussion.

2. RELATED WORK

In order to reduce the operation cost for the Web access, auto-pilot systems, such as Internet Access Manager [4] and WebClip [14], have been developed. Those systems automatically download the Web pages whose URLs are specified by the user. Although the users do not have to repeatedly specifies the URLs, they are still required to search the downloaded page for their necessary information because the auto-pilot systems do not extract a portion of a Web page.

Pointcast Network [10] and SIFT [13] are information personalization services which provide information according to the user interest. In those services, the provided information is selected by

keywords and topics that the user had specified in advance. However, it is a heavy burden for the users to specify the most appropriate keywords for explaining the user interest. In Scrapbook, the users are not required to give such keywords, but simply give an example of their desired information by selecting a portion within a Web page.

Krakatoa Chronicle [3] is a personalized online newspaper which infers the user interest from his/her page scrolling operations for reading articles and personalizes the order of articles in the newspaper. This system is similar to Scrapbook, in respect of that the users tell the system their interest using the examples (actual articles). In Krakatoa Chronicle, however, the user interest is so implicitly indicated, and it is hard for users to imagine how the actions of reading articles will be reflected in the future personalization. In Scrapbook, the users can explicitly indicate the desired information.

Scrapbook can be categorized as a programming by demonstration system. The programming by demonstration (PBD) [1] is a technique to allow users to automate a repetitive task by simply demonstrating how the task can be accomplished on example data. Since the PBD offers an advantage that the special programming skill is not required to create programs, many PBD systems, such as SmallStar [3], Peridot [7], Eager [2], DADIE [11] and DemoOffice [12], have been developed. Scrapbook is one of the first PBD systems to automate the task of extracting specific portions from Web pages and to reduce the Web browsing cost.

3. Internet Scrapbook

3.1. Overview

Internet Scrapbook is an information personalization system which allows users to browse their necessary information only. Also, with its facility of programming by demonstration, it enables users with little programming skill to automate the daily Web browsing tasks.

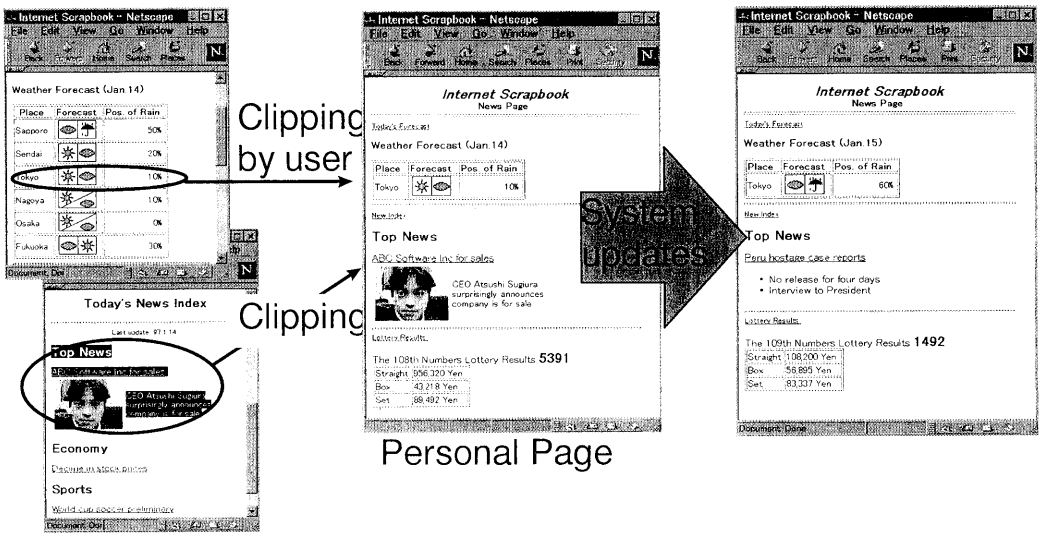
Main user task on Scrapbook is to create a personal page by clipping portions from Web pages. Concretely, the user first selects data on one browser for regular Web browsing (Figure 1a), and next copies the selected data to the other browser for the personal page, invoking a clipping command provided by Scrapbook (Figure 1b). Repeating these operations, the information in multiple Web pages can be gathered on one personal page.

Every time the user copies the Web data between two browsers, the system generates a matching pattern for finding the location where the user selects in the source Web page. Even when the source pages are modified in the Web sites, the system extracts portions, corresponding to the user selections, from the new source pages and re-composes the personal page with the latest information. Therefore, once the user creates the personal page, the user can obtain the latest information by simply invoking a updating command of the personal page (Figure 1c).

An important design principle in Scrapbook is to make the pattern matching procedure simple enough for users to understand its mechanism and easily examine whether the personal page will be updated as desired. Scrapbook is an inference system that infers a user desired portion in a source page from a single example, and there is no guarantee that the inference is always correct. Therefore, it is essential that the users can anticipate the system behavior in advance.

Figure 2 shows a system architecture of Scrapbook. The system consists of three modules: clipping module, matching pattern generator, and personal page updating module.

Current version of Scrapbook operates on Windows95/NT3.51, and it can copy and paste the data selected on Netscape Navigator 2.0 and upper versions. We are planning to make Microsoft Internet Explorer (MSIE) available.



(a) Browsers for regular browsing (b) Browser for personal page (c) Updated personal page

Figure 1: Overview of Internet Scrapbook

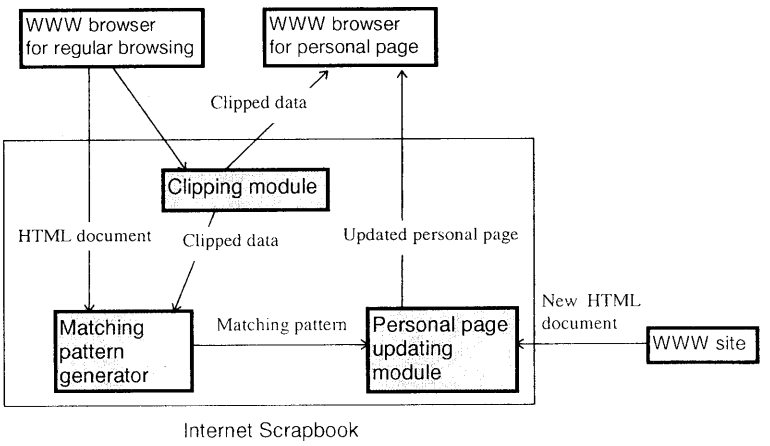


Figure 2: System architecture.

3.2. Data Clipping

Clipping module copies the data selected on the Web browser and pastes it on the other browser for a personal scrapbook page. In the data clipping, it is required to copy not only the selected texts but HTML tags which affects the selected texts. However, Netscape Navigator, our target browser, does not have a facility to offer such specific HTML tags to other applications. In order to copy the related HTML tags, Scrapbook obtains the source document through NCAPI (Netscape Client Application Program Interface), parses it and extracts all the HTML tags related to the selected text from the source document.

3.3. Pattern Matching

Every time the user clips the data to the personal page, the system generates a matching pattern for finding a portion where the user had selected in the source Web page, from a single example of the data selection operation. The generated matching patterns are used in updating the personal page.

In Scrapbook, the data to be extracted is determined by *line patterns*, which are the previous/first/next lines of the user selection on the original source page. For example, if the user selects data as shown in Figure3a, the system generates the following line patterns.

[Previous Line] "Last updated: 97.1.14"
[First Line] "Top News"
[Next Line] "Economy"

To determine a starting point of the data to be extracted, the system first tries to find a point where the previous/first line patterns completely match. If such a point can not be found, the system performs partial matching to find a point with the largest matching degree.

For example, let us consider a case where the Web page of Figure3a is modified as shown in Figure3b. In the page of Figure 3b, there is no adjacent two lines that match both the previous line pattern "Last updated: 97.1.14" and the first line pattern "Top News". So, the system tries to find adjacent two lines which contain the words in line patterns most. At the locations (1) and (2) in Figure 3b, the first line pattern "Top News" completely matches. However, the system chooses the location (2) because a part of the previous line pattern "Last updated:" matches in (2).

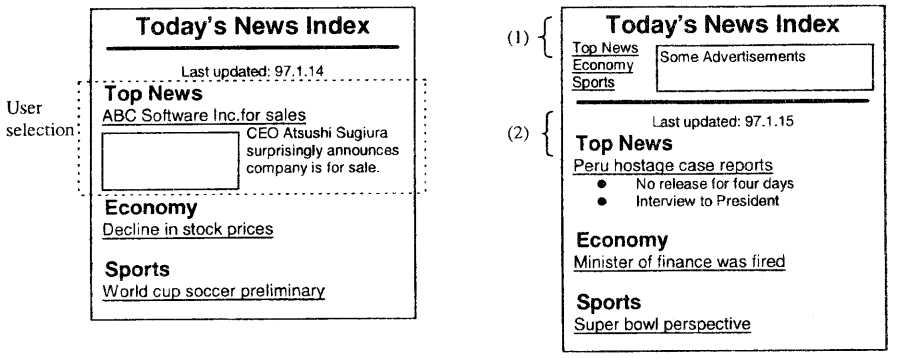
Likewise, the end point of the data to be extracted is determined by using the next line pattern.

This simple matching procedure allows users to anticipate the system's behavior in extracting data from the source page, simply checking the previous/first/next line of data the user had selected. In the case of Figure 3a, it is expected that most of the text contained in the line patterns will remain in the future source page. The first line "Top News" and the next line "Economy" will be unchanged, because they represent categories of articles, not daily articles. In the previous line, although the date "97.1.14" will be changed, the text "Last updated:" will remain. Therefore, it is anticipated that the system will be able to extract the proper portion.

Although the pattern matching procedure is simple, it can operate well on many Web pages. Most of the Web pages are composed so that users can easily browse the pages. In order to help users find their target information, the frequently modified parts of a Web page are usually proceeded by the permanent titles. Since users usually need the frequently modified portions surrounded by those permanent parts, the line

patterns, which are the previous/first/next lines of the user selection, are useful for the data extraction in many cases.

In addition to the line patterns, Scrapbook uses HTML tag patterns, which are the order of the tags in a source page, such as extracting from the first <H2> tag to the second one. However, the system uses the line patterns prior to the tag patterns. The tag patterns are only used when no candidate of the data extraction can be found with the line patterns or when multiple candidates. In cases where neither line patterns nor tag patterns match any portion within a Web page, the system fails the data extraction.



(a) Before modification (b) After modification

Figure 3: Source Web pages.

4. INFORMAL EXPERIMENTS

In order to ascertain the effectiveness of the pattern matching procedure of Scrapbook, we did informal experiments with 90 pages on 25 Web sites for daily news, weather forecast, etc. The pattern matching using both line patterns and tag patterns can find the appropriate portions on 79 pages. Using line patterns only, the pattern matching can operate well on 62 pages.

5. Discussion

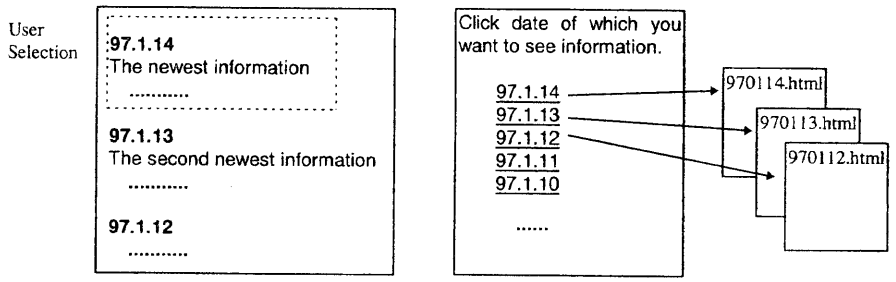
5.1. Updating Personal Pages

As shown in the experimental results, Scrapbook can appropriately extract the latest information from many Web pages. The current pattern matching algorithm of Scrapbook is useful for the pages where the old information is replaced with the latest one. That is, the Web page contains only the latest information with a fixed format. The page shown in Figure 3 belongs to this category.

In some pages, however, the pattern matching algorithm does not operate well. A typical Web page where the pattern matching fails is that the latest information is added to the head of the source page and it appears in the page with the older information, as shown in Figure 4a. Let us consider the case where a user selects data in the dash line. In this case, the texts "97.1.14" and "97.1.13" are used as line patterns. Since the line pattern texts remain as it is even when new information (e.g. one for 97.1.15) is added, the

system always extracts the same data. We need to improve the matching procedure so that it can extract the new information by using tag patterns prior to line patters in such situations.

Another typical case is that the latest information is provided in a newly created Web page (this means a URL for the latest information is changed) and a hyperlink to jump to the new page is added to an index page, as shown in Figure 4b. Scrapbook only knows the URL for the source page where the user first used for the data clipping, and it is unable to access different Web pages from the original source page. However, if URLs are named using the date information when the pages are created as shown in Figure 4b, the newest page might be able to be found in the Web site. We are planning to develop the capability of retrieving new URLs.



(a) (b)

Figure 4: Typical Web pages that the system fails to extract proper portion.

5.2. Information Discovery

Scrapbook extracts a portion from a Web page and shows the users only the extracted portion. While this facility provides efficient browsing of the Web information, this might cause overlooking of useful information. For example, if prompt announcements and temporary special reports are inserted to the source Web page, the users do not have an opportunity to see such information through browsing the personal page only.

5.3. Copyrights of Web pages

When creating personal pages in Scrapbook, the users should pay attention so that they do not infringe on the copyrights of source Web pages. Since the data clipping from the Web pages means their modification, the user should never copy data from pages where its modification is prohibited and paste it on a personal page. Also, the created personal page should never be re-distributed.

6. CONCLUSION

This paper describes a technique to obtain user desired information from the Web pages with minimum efforts. The proposed demonstration-based technique frees users from the repetitive tasks on the Web without requiring to write any script or program.

Scrapbook is an inference system which anticipates the user desired portion in the latest Web page. There

is no guarantee that the system infers appropriately. However, we consider that the users do not suffer much loss even in the case where wrong locations are extracted from a source Web page. This is because the users can easily browse the whole source page and obtain their necessary information. In addition, the simple pattern matching procedure allows users to anticipate whether the future pattern matching can operate as desired in advance and to understand reasons why the matching did not operate well. We consider that Scrapbook can be used without irritation.

ACKNOWLEDGMENTS

The authors express their appreciation to Satoshi Goto and Shiro Sakata of NEC Corporation for giving them the opportunity to pursue this research.

REFERENCES

[1] Cypher, A. ed. "Watch What I Do: Programming by Demonstration," MIT Press, 1993.

[2] Cypher, A., "Eager: Programming Repetitive Tasks by Demonstration," in [1], pp.205-218, 1993.

[3] Halbert D. C., "SmallStar: Programming by Demonstration in the Desktop Metaphor," in [1], pp.103-123, 1993.

[4] Internet Access Manager: http://www.psinfo.nec.co.jp/iam/

[5] Kamba, T., Bharat K. and Albers M.C., "The Krakatoa Chronicle: An Interactive Personalized Newspaper on the Web," Fourth International World Wide Web Conference Proceedings, pp.159-170, 1995

[6] Microsoft Internet Explorer: http://www.microsoft.com/ie/

[7] Myers, B.A., "Peridot: Creating User Interfaces by Demonstration," in [1], pp.125-153, 1993.

[8] NCSA Mosaic: http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/

[9] Netscape Communication: http://www.netscape.com/

[10] PointCast Network: http://www.pointcast.com/

[11] Sugiura, A. and Koseki, Y., "Creating Database Queries by Demonstration," Proceedings of the Eleventh IEEE Symposium on Visual Languages, pp.164-171, 1995.

[12] Sugiura, A. and Koseki, Y., "Simplifying Macro Definition in Programming by Demonstration," Proceedings of UIST'96, pp.173-182, 1996.

[13] Yan, T.W. and Garcia-Molina, H., "SIFT: A Tool for Wide-Area Information Dissemination," USENIX Technical Conference, pp.177-186, 1995

[14] WebClip: http://www.paperclip.com/products/webclip.htm