AN E-COMMERCE TECHNOLOGY : WEB PERSONALIZATION ON THE PATRON WEB

Kiyomitsu, Hidenari

(Citation)
Kobe University Economic Review,46:91-105

(Issue Date)
2000

(Resource Type)
departmental bulletin paper

(Version)
Version of Record

(JaLCDOI)
https://doi.org/10.24546/81000928

(URL)
https://hdl.handle.net/20.500.14094/81000928

AN E-COMMERCE TECHNOLOGY: WEB PERSONALIZATION ON THE PATRON WEB

By HIDENARI KIYOMITSU

kiyomitu@kobe-u.ac.jp

Internet-related terms have become socially popular, and they are essential tools for doing business on-line. In this paper, we propose the notion of personalizing the contents of Web pages based on users' spatio-temporal information (time and location) and their access histories. The major objective of the proposed mechanism is to realize the reflection of a Web page author's intention about his Web data. A page personalization rule for web page reconfiguration is described in a declarative manner, which the author can define easily as its metadata. Furthermore, we introduce the related works about Web personalization and adaptation as a background technology for e-commerce and e-business.

Keywords: WWW, Personalization, Adaptation, E-Commerce, E-Business, Database, Active Database, ECA-Rule

1. Introduction

Currently, there are many words beginning with "E": cf. E-Commerce, E-Business, E-Trade. They are Internet applications for performing business transactions between a customer and an enterprise enabled by TCP/IP communication. Actually, the existing infrastructure is significant, and the concepts and strategies of them are much more important.

Suppose that a vast number of Web pages have been stored on Internet sites already, and the authors connect them using the hyperlinks. Hyperlinks are very flexible in the sense that they allow authors to freely connect their pages with each other. How do you realize business relationships such as one-to-one, business-to-business dealings of bidirectional communications over computer networks? Hyperlinks form an implicit navigation path for users to follow, but only by their own choice. For example, when a user visits a Web page for the first time, the author of the page may wish the user to read contents of a copyright notice. While some obedient users may click the link anchors to read them, others will not. Furthermore, even for those obedient users, it is sufficient for them to read the copyright notice only once. Our first goal is to embed it as contents of the page for users visiting it for the first time, and to hide it from others.

It is also important to know the place from which a user accesses a Web page and the date when he accesses it. If a user comes from page A by clicking an advertising banner, the author of the present page may wish to embed a commercial message as a part of its contents. However, if another user comes from page B, then the author may wish to add an additional content to the current page. In our page personalization rules, we consider several components concerned with users' access histories, such as times of users' visits, time duration since previous access, and the links used to navigate to the current page.

In our page personalization rules, we consider the user's physical location, the logical location at which he is assumed to be, etc. Our use of the user's physical location is worth stressing. For example, suppose that the user wishes to know the way to the 3rd research building on Rokkodai campus. If he is currently at the bus stop "The front gate of Kobe university", he needs a local and detailed map. On the other hand, if he is far from the building, he will need a global map. In order to realize this goal, we introduce the notion of a geographical cookie named geo-cookie. A default value of a user's geo-cookie is the place where he actually exists, but he can specify it as the place where he is assumed to be.

In summary, our personalization rules mainly consider the following components concerned with Web page reconfigurations and access rights:

- Logical and physical locations from which a user reaches a Web page.
- User's temporal situation and histories of access to a Web page.
- Aggregate information about all the users' access behaviors for a Web page.

The remainder of this paper is organized as follows. In Section 2, we describe the notion of page personalization rules. Section 3 describes the reconfiguration by location. Section 4 describes the reconfiguration by time. In Section 5, we show our prototype implementation and examples. Section 6 describes related work and the major contributions of our work. In Section 7, we compare our work with the conventional active rules of active database systems. Section 8 gives concluding remarks.

2. Page Personalization Rules

Suppose that an author intends that his users read the contents of a copyright notice that is usually embedded in a page called "Read Me First". He thus establishes a hyperlink from his "Home page" to the page "Read Me First" using a specific link anchor. Not all users, except for some obedient ones, will do exactly as the author wants them to do.

Conventionally, it is not certain that an author's intention is truly reflected by a user's browsing behavior, because each user's link navigation pattern depends upon the user's own intention. The author might intend to force a user to read the contents of "Read Me First" only if the user is accessing the page for the first time. However, he does not intend that all users do so. Therefore, our solution to this problem is that the system adds all the contents of "Read Me First" to the Web page only for those users who are visiting the page for the first time.

Our approach to Web reconfiguration is based on author-defined page personalization rules that have three components: a user's behavior, a historical condition, and a server action. The syntax of our page personalization rule is:

user's behavior: historical condition $\Rightarrow$ *server action*

The user's behavior part and the historical condition part of a rule are separated by a colon (:), and the historical condition part and server action part of a rule are separated by an arrow ($\Rightarrow$). The user's behavior part of a rule describes an event to which the rule may be able to respond. The history part of a rule considers the browsing history of a user and all of the server histories as a condition. The server action part of a rule describes the action to be carried out by the rule if the relevant user's behavior has happened and the historical condition evaluates to TRUE.

The *active rules of active database systems* are closely related to our use of Web reconfiguration for changing the view of a page. We regard a user's request for reading a page and his situation (location, date, his schedule, etc.) as an event, and an access history of each user or an aggregation of access histories as a condition. Also, the behavior of the server is regarded as an action.

2.1 Definition

A page personalization rule for adding, hiding, or replacing Web content including hyperlinks is described in a declarative manner, which the author can define easily.

The user's behavior part (event part) of a rule can be regarded as an event of ECA rules in active database systems. The event part of a rule consists of:

- page: A page for which a user has issued a request to read
- location: The place where a user is or where he is assumed to be
- date: The date when a user issues a request to read the page

The event that a user is accessing page A is denoted in our syntax by: `read(A)`.
In Web systems, a request for reading a page is the only event for accessing a page by users. Therefore, it is simply abbreviated as: `A`

For example, an event in which a user from Kobe requests page A in the afternoon is expressed by:

`read(A), location = Kobe, date = afternoon`

The historical condition part of a rule can be regarded as a condition of ECA rules. In our page personalization rule, the historical condition consists of two parts: a user's browsing history and the aggregation of all users' browsing histories. The syntax of a historical condition part is denoted by:

<personal history>,<aggregation>

The personal history part consists of:

- times: The number of times that the page has been accessed by the user.
- valid_time: The time duration elapsed since the last time the user accessed the page.
- old_info: The identifier of added (or hidden or replaced) content.
- referrer: A page from which a user navigated to reach the current page.
- location: The place where a user was or where he is assumed to be.
- date: The date when a user issued a request to read the page.

The aggregation part consists of:

- total_num: The total number of users who have accessed the page.
- distinct_users: The number of different users who have accessed the page.
- old_info: The identifier of added (or hidden or replaced) content.
- referrer: A page from which users navigated to the current page.
- location: The place where users were or where they are assumed to be.
- date: The date when users issued a request to read the page.

Each element is parenthesized. For example, a historical condition where the user has not browsed the page yet and the total number of users who have accessed the page before is less than 2,000 is expressed by:

`<(times = 0)>, <(total_num < 2000)>`

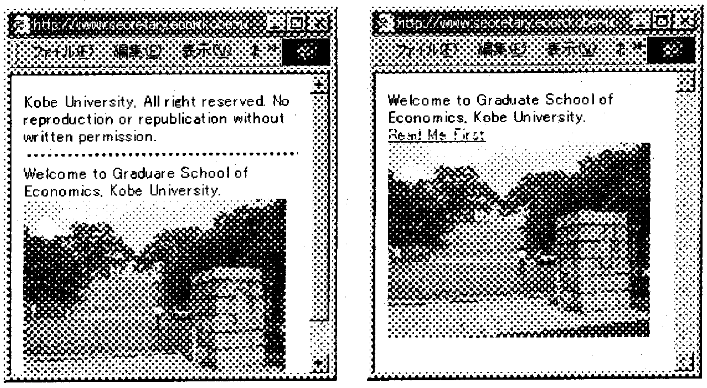
Each of the parameters such as old_info, referrer, location and date is used with a subordinate parameter "times" optionally. For example, the historical condition stating that the user has browsed the page from San Diego more than three times before is expressed by:

`<(location = San Diego, times >3)>, <>`

The server action part of a rule can be regarded as an action of ECA rules. In a page personalization rule, the action consists of two parts: a page to be activated and a function to be carried out. For example, an action in which the server activates the user's access to page A and embeds a content "Read_Me_First" to page A and returns the result is expressed by:

`activate(A), add(A, Read_Me_First)`

Activating page A means that the server issues an access right to the user. Formally, it is denoted by activate(A). Since the only thing our system can do to a page is to activate it, this is expressed merely by A. If there are no rules that have activating page A in their action part, it means that all users have access to page A.



Kobe University. All right reserved. No reproduction or republication without written permission.

Welcome to Graduare School of Economics, Kobe University.

Welcome to Graduate School of Economics, Kobe University.

Read Me First

FIGURE 1. An Example of Reconfiguration

2.2 Example

An example of defining page personalization rules is as follows:

The system adds all the contents of read_me_first.html to page A for a user who accesses it for the first time or has not accessed it for more than six months.

A: <(times=0)>, <> ⇒ add(A, read_me_first.html);
A: <(valid_time=6 months)>, <> ⇒ add(A, read_me_first.html);

Figure 1 shows two different visual images - the left one shows the page for the first_time users and the right one shows the page for the repeaters (frequent visitors). Note that the top one contains an additional copyright notice.

3. Reconfiguration by Location

Suppose that a user is at Kobe and he is browsing a page about Kobe University. What happens when he clicks a link anchor to a page that explains how to go to Kobe University? Conventional systems will provide the same page with the same contents, for example the same map and the same timetable of the bus to Kobe University, no matter whether the user is browsing from San Jose, New York or Osaka. We believe it is desirable to change a Web view according to a user's geographical information. For example, our Web reconfiguration mechanism provides the way from Rokkomichi station to Kobe University for a user who is in Japan, and the way from Kansai airport to Kobe University for a user who is not in Japan.

In order to personalize the view of a page based on a user's location, we introduce the concept of geo-cookies. Here, the cookie is a chunk of information concerned with each user's access, which is updated by a Web server and transmitted to the user's Web browser client. Our notion of the geo-cookie is an extension of a conventional cookie, in that it contains additional information about the user's physical location. Using our geo-cookie, an author can define a page activation rule as follows.

map.html, location = Japan: <>, <> ⇒ replace(map.html, abroad.gif, national.gif);

This example means: if a user in Japan browses the page map.html, then the system will replace an image called national.gif with another image called abroad.gif. Both images, abroad.gif and national.gif, are either prepared by the author himself or calculated by a geographical system. This way, we can expect all users, whether living in Japan or coming from abroad, to be able to reach Kobe University.

Suppose that another author intends that users who have previously accessed page A from San Diego should get the contents of the page named 'present' when they access page B. This page personalization rule is expressed as:

B: <(A, location=San Diego)>, <> $\Rightarrow$ add(B, present)

Since some aggregation information of users' histories is available in our page personalization rules, authors can define another kind of useful rule. Suppose that an author wants to embed contents of the page named 'thanks' into page A, if the total number of users who have accessed page A from San Diego exceeds 1000. This is expressed as:

A: <>, <(location=San Diego, times > 1000)> $\Rightarrow$ add(A, thanks)

Also, geographical information can be described in the action part of rules:

A: <>, <(location=San Diego, times > 1000)> $\Rightarrow$ (activate(B), San Jose)

This indicates that a user can browse page B from San Jose once he has accessed page A, on the condition that the total number of users who have accessed page A from San Diego exceeds 1000.

The location of a user can be described in all three parts of a page personalization rule. That is, we use the location as a modifier, which specifies a user's access situation from his spatial viewpoint.

4. Reconfiguration by Time

It is important to consider the date when a user accesses the page. Suppose that some place can be reached by subway in the daytime, and that the place can be reached only by taxi after midnight. If an author defines the following page personalization rule, the image for the way will be changed according to the time of the day:

map.html, date = midnight: <>, <> $\Rightarrow$ replace(map.html, day.gif, night.gif)

The function replace is like **do instead** in active rule [6]. After midnight, the system sends the contents of "night.gif" instead of "day.gif" to any users. The date of access need not be a real date.

Another example:

index.html, date = morning: <(midnight, times > 3)>, <>
⇒ (present.html, Sunday), add(index.html, link-to-present)

This page personalization rule means: If a user has previously accessed the page "index.html" more than three times after midnight and he is now accessing the page in the morning, then the system issues an access right to the page "present.html", which he will be able to access on Sunday. The system also adds a hyperlink for him from the page "index.html" to the page "present.html".

The date of a user's access can be described in all three parts of a page personalization rule. That is we use the date as a modifier, which specifies a user's temporal access situation.

5. Expression in XML

The rules for Web page reconfiguration are expressed in XML. First, we specify a DTD for our Active Web (Figure 2). The XML element of `RULE` has three subelements `EVENT`, `CONDITION` and `ACTION`, and two attributes ID and NAME. An attribute ID indicates the identifier of the rule which is used by the system. The other attribute NAME indicates the name of the rule which is used by the author. The event part of a rule is: the subelement of `EVENT` is `READ` only, because Web systems allow users only to issue "http_get" requests. Subelements of `READ`, `LOCATION` and `DATE` indicate the location from and date at which the request is issued respectively. The other subelement `REFERRER` indicates the source page where the user is navigating a link. The historical condition part of a rule `CONDITION` has two

```
<!ELEMENT RULES (RULE*)>
<!ELEMENT RULE (EVENT,CONDITION?,ACTION)>
<!ELEMENT EVENT (READ)>
<!ELEMENT CONDITION (USER?, AGGREGATE?) >
<!ELEMENT ACTION (ADD|HIDE|REPLACE)>
<!ELEMENT READ (URL,REFERRER?,DATE?,LOCATION?)>
<!ELEMENT REFERRER (#PCDATA)>
<!ELEMENT DATE (#PCDATA)>
<!ELEMENT LOCATION (#PCDATA)>
<!ELEMENT USER (TIMES?,DATE?)>
<!ELEMENT AGGREGATE (CONTENT, TIMES?,REFERRER?)>
<!ELEMENT ADD (LINK|IMAGE|...)>
<!ELEMENT HIDE (LINK|IMAGE|...)>
<!ELEMENT REPLACE ((LINK|IMAGE|...), (LINK|IMAGE|...))>
<!ELEMENT LINK (URL,(LABEL|IMG))>
<!ELEMENT IMAGE (URL,ALT?)>
<!ATTLIST RULE ID #IMPLIED NAME (#PCDATA?) >
```

FIGURE 2. XML DTD Specification for Web Personalization

subelements. A subelement `USER` is used to evaluate an access history of a user who gives rise to the event of the rule or triggers the rule. The other subelement of `CONDITION`, `AGGREGATE` indicates aggregation from the access history date of the server (c.f. time of all access to page A). Since the historical condition part of a rule is optional, our DTD allows authors to define event-action rules. With the action part of a rule, an author can configure **ADD**, **HIDE** and **REPLACE** functions which enable a reconfiguration of the combination of contents in the page.

Also, we show an example of an XML statement (Figure 3). This XML statement specifies that, if a user accesses to page default.asp from advl.asp using the hyperlink, then the system add a link anchor to page speciall.asp with label "Special Offer". If a user accesses page default.asp from adv2.asp, then the system adds a link anchor to page special2.asp with label "Special Offer".

```
<RULES>
<RULE ID=rule1>
<EVENT><READ><URL>default.asp</URL><REFERRER>adv1.asp</REFERRER></READ></EVENT>
<CONDITION><USER><TIMES operator=mt>10</TIMES></USER></CONDITION>
<ACTION><ADD><LINK><URL>special1.asp</URL><LABEL>SpecialOffer</LABEL></LINK></ADD></ACTION >
</RULE>
<RULE ID="rule2">
<EVENT><READ><URL>default.asp</URL></READ><REFERRER>adv2.asp</REFERRER></EVENT>
<ACTION><ADD><LINK><URL>special2.asp</URL><LABEL>SpecialOffer</LABEL></LINK></ADD></ACTION>
</RULE>
</RULES>
```

FIGURE 3. An Example of XML Statements

The extensible style sheet language(XSL) is a kind of implementation for presenting a XML document. A query language and its implementation can be embedded in XSL such as XQL and Javascript calling JDBC/ODBC drivers.

A query result for presenting a XML document in XSL is shown as an example (Figure 4). This example of Figure 4 indicates the action add a link with its (source) anchor string or (source) anchor image. The implementation for dynamic reconfiguration rule in XML processes runtime XML and XSL from original XML document and original XSL style.

Figure 5 shows an architecture for using ASP on Microsoft IIS and ISAPI Filter. In order to return the content of the page in HTML, "default.asp" includes a script to parse runtime XML and XSL as follows.

```
<xsl:for-each select="RULES/RULE/ACTION/ADD">
<xsl:if test="LINK/LABEL">
<A><xsl:attribute="HREF">
<xsl:value-of select="LINK/URL">
</xsl:attribute>
<xsl:value-of select="LINK/LABEL">
</A>
</xsl:if>
```

```
<xsl:if test="LINK/IMAGE">
<A><xsl:attribute="HREF">
<xsl:value-of "select=LINK/URL">
</xsl:attribute>
<img><xsl:attribute="SRC">
<xsl:value-of select="LINK/IMAGE/URL">
</xsl:attribute></img></xsl:if>
</xsl:for-each>
```

FIGURE 4. XSL Example

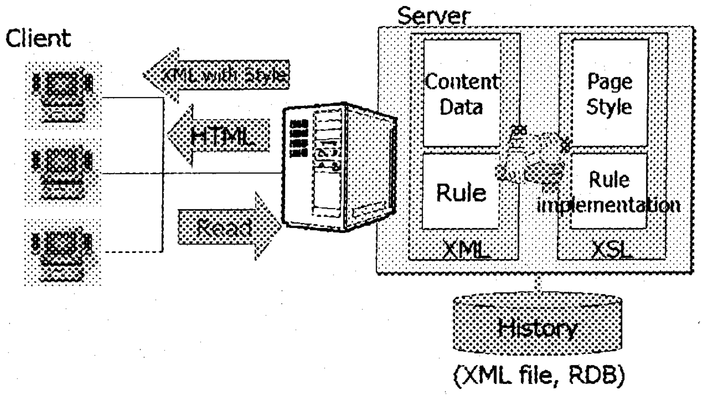


FIGURE 5. An Implementation Example on Microsoft IIS

```
Set XMLDoc=Server.CreateObject("Microsoft.XMLDOM")
Set XSLDoc=Server.CreateObject("Microsoft.XMLDOM")
XMLDoc.async = false
XMLDoc.load(Server.MapPath("default.xml"))
XSLDoc.async = false
XSLDoc.load(Server.MapPath("default.xsl"))
response.write(XMLDoc.documentElement.\
transformNode(XSLDoc.documentElement))
```

FIGURE 6. Visual Basic Script

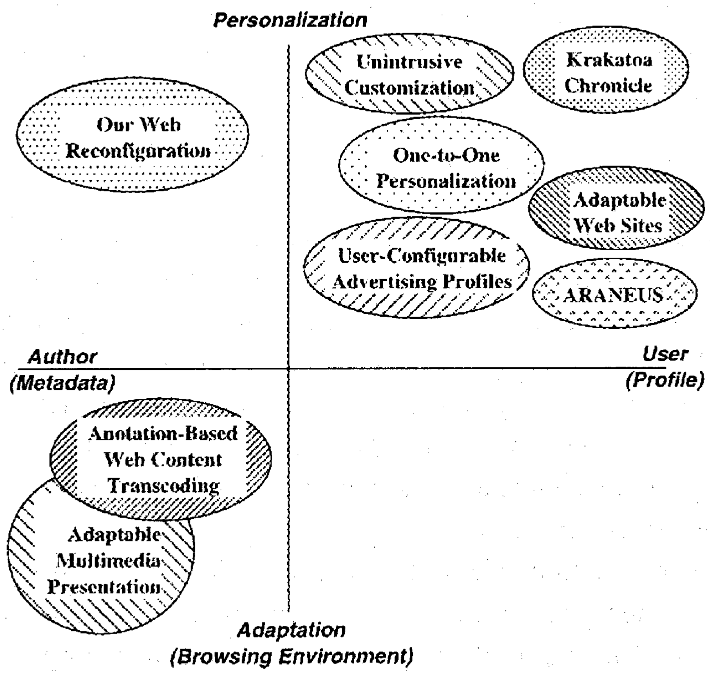

Personalization
Unintrusive Customization
Krakatoa Chronicle
Our Web Reconfiguration
One-to-One Personalization
Adaptable Web Sites
User-Configurable Advertising Profiles
ARANEUS
Author (Metadata)
User (Profile)
Anotation-Based Web Content Transcoding
Adaptable Multimedia Presentation
Adaptation (Browsing Environment)

FIGURE 7. The Matrix of Research Areas

6. Related Work

With the growing importance of Web personalization and adaptation, there has been a lot of research in this area. Figure 7 classifies current research directions concerning Web personalization and/or adaptation. As shown in this figure, the research projects are mainly classified under the following categories:

- **Personalization vs. Adaptation**
  Here, Web *personalization* means that Web contents or Web sites are customized for each user or each user community. Web *adaptation* means that Web contents or Web sites are customized for each user's browsing environment, such as PC, mobile phones, and PDAs.

- **Author-specified metadata-based vs. User-profile-based**
  This category means that the Web personalization/adaptation is done either mainly by metadata specified by content authors, or by user-profile data captured explicitly or implicitly from users.

- **Page-based vs. Site-based**
  Here, *page-based* Web personalization / adaptation means that each Web page is a unit for personalization or adaptation. *Site-based* Web personalization / adaptation means that a collection of Web pages are dynamically (and virtually) organized as a Web site.

The above three categories are orthogonal and therefore, any combination of the three categorical aspects is possible. As described in the previous section, our approach can be regarded as metadata-based web personalization for each Web page.

6.1 User-profile-Based Personalization

Kamba et al. have developed an interactive personalized newspaper on the Web, called the *Krakatoa Chronicle*[8]. At the server side, articles are automatically gathered from news sites, and are selected for each user by analyzing the characteristics of gathered articles and each user's profile. Selected articles are organized into a personalized newpaper with flexible layout control. The user profile is derived and updated by a user's explicit relevance feedback and the user's implicitly-captured browsing behavior.Also, a community score for the articles is dynamically computed by aggregating user profiles. This work can be regarded as a Web personalization based on user profiles.

Baudish and Leopold[3] introduced *user-configurable advertising profiles*, which adopts the usage of relevance feedback in selecting advertising banners. They addressed several problems of advertising banners. In their approach, a banner clicked by a user and its target page are automatically recorded and used to update the user profile. Also, negative feedback

TABLE 1. Semantics of the User Profile

	Type	Definition	Source
Our Approach	Individual, Aggregation	Implicit	History
Krakatoa Chronicle	Individual, Aggregation	Explicit	History, Non-History
Unintrusive Customization	Aggregation	Implicit	
User-Configurable Advertising Profile	Individual	Explicit	Non-History
One-to-One Personalization	Individual	Implicit, Explicit	History, Non-History
Annotation-Based Transcoding	-	-	-
User-Adaptable Multimedia Presentation	Individual	Explicit	Non-History
Adaptable Web Sites	Individual	Explicit	Non-History
ARANEUS	Individual	Explicit	Non-History

information is recorded and used to update the user profile. This method is a kind of relevance feedback, and the system explicitly asks the user what advertising banners he likes and does not like. It should be noted that the approach uses only user information, and that it cannot treat the page author's intention regarding users browsing and navigation.

Langheirich et al. have developed *unintrusive* customization techniques for Web advertising[10]. Although their approach does not use individual user profiles, it uses statistical information about users' behaviors (clicking advertising banners). The relevancy between keywords and advertising banners is computed and updated by the users' behaviors. Based on the relevancy, most relevant advertising banners are automatically inserted into Web pages retrieved by search engines.

In the field of database community, Web personalization has been becoming an important research issue, especially from the viewpoint of e-commerce. Ceri et al. recently introduced a solution to the *Web one-to-one personalization* by using business rules (a kind of metadata) and user purchase histories (a kind of user profile)[5, 4]. For a user profile, they collect users' identifiers, email addresses, login histories, group memberships, and trace information including purchase histories. The business rules mainly describe the priority of commercial products and the recommendation policies. The notion of the *active database rules* is used for implementing their business rules and for updating user profiles. The major differences between our work and their work are:

(1) Our work uses each user's browsing history as a user profile. In Ceri's work, a user profile mainly consists of user identifiers, email addresses, login histories, group memberships, and trace information including purchase histories.
(2) Page personalization rules are tools for content authors to represent how their content should be viewed by each user. Ceri's business rules are tools for e-commerce service providers to dynamically recommend commercial products.
(3) Our personalization rules are not driven by purchasing events, but by Web user's location, date, referrer, etc.
(4) Our personalization rules are not evaluated by a user's profile but by his browsing history and the aggregate information about all the users' accesses to the Web page.

Anupam et al. proposed a tool to record each user's navigation behavior, called *WebVCR*[1]. This tool is a client-side mechanism to record each user's hyperlink navigation history and to replay the navigation history. By replaying a navigation history, the user can automatically access a certain Web page that could be reached after a certain link navigation. WebVCR can not only *record* the URLs of pages, but can also remember many user's actions; click, check boxes, fill forms and so on. Furthermore, a user of WebVCR can *play/back* the recorded navigation automatically. Indeed, this is a kind of tool for recording / replaying user's behaviors, but is not a tool to reconfigure Web pages.

The characteristics of user profiles in related work may vary in many ways. Table 1 shows the type, definition and source of user profiles. The type 'individual' means that the profile mainly consists of a user's own personal data. The type 'aggregate' means that it utilizes aggregated information derived from profiles of several users who belong to a community, or of all users. The 'definition' indicates who decides the value of data in the profile. "Explicit" means that a user specifies the value of the element in the profile. The 'source' describes whether or not the user profile uses "History" or "Non-History". If a user profile is extracted from server log, access histories or the user's browsing action, the source of the profile is categorized as "History".

Summarizing this section, our approach achieves not only Web page personalization by active-rule-like metadata, but also access control over pages. However, it should be noted that our approach does not necessarily mean that users should always work in a restrictive manner. We still regard the Web as the environment where a user can freely access any pages. The page author can only suggest how the pages are to be browsed and navigated. Therefore, our research is not meant to restrict users' browsing behavior, but to give some incentives to obedient users.

6.2 Author-specified-metadata-based Web Adaptation

Hori et al. introduced *annotation-based Web content transcoding*[7]. The view of a Web page is dynamically adapted to each user's browsing environment by author-specified metadata. The metadata for an exciting Web page is defined as an external annotation file (XML/RDF) using Xpaths and Xpointers. This metadata is used to replace original contents with alternative contents according to the selection criteria specified in the annotation file. This alternation is similar to the 'replace' function in our personalization rules, but they don't discuss 'hiding' and 'adding' content. They mainly focus on adapting Web page to mobile environments such as PDAs, with small displays and low bandwidth. Their approach does not consider Web personalization for each user.

Rousseau et al. proposed a way for *user adaptable multimedia presentation*, that mainly focuses on synchronized multimedia object playback[13]. Metadata is defined by authors especially for the purpose of synchronized presentation of multimedia objects. Users can reconfigure the playback synchronization according to their interests and their browsing environments: browser, bandwidth, monitor size, etc. This work can be regarded as a kind of Web adaptation for synchronized multimedia content.

6.3 Site-Based Personalization

Perkowitz et al. proposed *adaptive Web sites* that automatically suggest alternative organization of their favorite contents based on user's access pattern[12]. They implemented the PageGather Algorithm for discovering candidate index page contents. It is a user profile approach, because they use visitor access patterns extracted from the server log.

Atzeni et al. proposed *ARANEUS*, which aims at generation *virtual Web sites* (site views) using database technologies[2]. In their approach, a site view is derived from multiple Web sites of interest, and some links to pages in the original sites are established. In the ARANEUS view generation, data are extracted from actual sites of interest and are organized under a database structure, which represents a structured view for the original semi-structured data. Then, conventional *database views* are defined and derived by users from the structured views by means of view definitions. Finally, the resulting table-data are again restructured into a form of hypertext. The major differences between ARANEUS and our approach are (1) our unit of personalization is a Web page, (2) our Web personalized views are automatically derived from author-specified metadata and user behavior histories, while user views are explicitly defined by users in ARANEUS.

7. Comparison with Active Databases

The active rules of active database systems are closely related to our use of Web reconfiguration for changing the view of a page[11]. We regard (1) a user's request for reading a page and his situation (location, date, his schedule, etc.) as an event, (2) an access history of each user or an aggregation of access as a condition, and (3) the behavior of the server as an action. Table 2 shows a comparison between conventional active rules in active database systems and our page-personalization rules.

The notion of active rules is a powerful tool for managing database state. Our page personalization rules can be fired (triggered) only by issuing a request for reading a page. The event of an active rule can also describe a temporal aspect, using **Clock** as a kind of timer. Our personalization rules can describe not only the moment of an event but also a calculated date from a user's schedule. Furthermore, page personalization rules can describe a user's real and virtual location, but active rules cannot describe locations.

As for the condition part, active rules describe the state of the database system and relate with other rules. The historical condition part of one of our page personalization rules is independent from other rules. If different conditions are defined for the same event, each condition is evaluated independently.

Our Web reconfiguration approach is, in spirit, similar to ECA (Event-Condition-Action) rules in active databases. The major differences between our work and ECA rules are summarized as follows:

(1) the targets are not relations in RDB but Web pages,

(2) our rules are formulated not by database events, but by web users' access histories, their locations and date of access.

TABLE 2. Comparison between Active DB and Web Reconfiguration

	Active DB	Web Reconfiguration
Event	Structure Operation, Transaction, Clock, External, Exception	Read Request, Date, Location, Schedule
Historical Condition	Transaction start, abort, commit, etc.	User's History, Aggregation
Action	Behavior Invocation, Structure Operation, Update Rules, Abort, External, Do Instead, etc.	Activate a Page add, hide, replace, date, location

8. Conclusions

In this paper, we proposed the idea of *Web reconfiguration* based on page personalization rules defined by authors. A page personalization rule for providing reconfigurable Web content including hyperlinks is described in a declarative language, which the author can define easily. We introduced a concept of a geo-cookie for reconfiguring Web pages based on a user's physical and logical location. Our geo-cookie can express not only the current location of user, but also a virtual (logical) location. The location and the date of a user's access can be described in all three parts of a page personalization rule. That is, our Web reconfiguration uses the location and the date as modifiers, which specify a user's access as a spatio-temporal aspect. Yahoo's "Yellow Pages" system uses a user's location to retrieve the result of queries. But Yahoo's location is an additional condition. The idea of using geo-cookies and a user's schedule is applicable not only to the desktop environment, but also to the mobile computing environment.

We briefly described our prototype implementation. In our implementation, authors can define page personalization rules using a conventional Web browser in a fill-in-the-blank manner. So, they need not worry about becoming slaves to programming and CGI.

Acknowledgement

Thanks are due to Dr. Sujeet Pradhan and other members of Tanaka Lab. of Kobe University for their useful discussions. This research is partly supported by the Research for the Future Program of Japan Society for the Promotion of Science under the Project: Research on Advanced Multimedia Contents Processing.

REFERENCES

[1] V. Anupam, J. Freire, B. Kumar, and D. Lieuwen (1999), "Automating web navigation with the webvcr," *Proceedings of WWW9* Conference.
[2] P. Atzeni, G. Mocca, and P. Merialdo (1997), "To weave the web,"*Proc of VLDB'97*, pp.206-215.
[3] P. Baudisch and D. Leopold (1997), "User-configurable advertising profile applied to web page banners," *Proc. of the first Berlin Economics Workshop.*
[4] S. Ceri, P. Fraternali, and S. Paraboschi (1999), "Data-driven one-to-one web site generation for data-intensive applications," *Proc. of VLDB'99*, pp.615-626.
[5] S. Ceri, A. M. P. Fraternali, and S. Paraboschi (1999), "One-to-one personalization of data-intensive web sites," *WebDB'99*, pp.1-6.
[6] K. R. Dittrich, S. Gatziu, and A. Geppert (1995), "The active database management system manifesto: A rulebase of albms feature," *Rules in Database System* LNCS-985, pp.1-17.
[7] M. Hori, G. Kondoh, K. Ono, S. ichi Hirose, and S. Sighal (1999), "Annotation-based web content transcoding," *Proceeding of WWW9 Conference.*
[8] T. Kamba, K. Bharat, and M. C. Albers (Winter 1996), "The krakatoa chronicle: An interactive personalized newspaper on the web," *WWW Jornal*, 1.
[9] H. Kiyomitsu and K. Tanaka (1999), "Navigation-dependent web-views: Defining and controlling semantic unit of web pages," *IEEE Proc. of ICSC '99*, LNCS-1749, pp.421-426.
[10] M. Langheinrich, A. Nakamura, N. Abe, T. Kamba, and Y. Koseki (1998), "Unintrusive customization techniques for web advertising," *Proceedings of WWW8.*
[11] N. W. Paton and O. Diaz (1999), "Active database system," *ACM Computing Serveys*, 31(1):63-103.
[12] M. Perkowitz and O. Etzioni (1999), "Towards adaptive web sites: Conceptual framework and case study," *Proceedings of WWW8.*
[13] F. Rousseau, J. Garcia-Macias, J. V. de Lima, and A. Duda (1998), "User adaptable multimedia presentations for the www," *Proceedings of WWW8.*