A System Architecture for Intelligent Browsing on the Web

Hsiangchu Lai, Tzyy-Ching Yang

Department of Information Management, National Sun Yat-sen University, Taiwan, R.O.C.

Compared with traditional business operations, WWW-based commerce has many advantages, such as timeliness, worldwide communication, hyper-links, and multimedia. However, there are also several browsing problems, such as getting lost, consuming a great amount of time browsing, and lack of customized interactive features. To acquire a competitive advantage over the countless number of Web sites, it is critical to solve these browsing problems. The purpose of this paper is to systematically review all browsing problems and then propose a system architecture for intelligent browsing on the Web. In this architecture, we present five kinds of browsing agents: recommendation agent, new-contents agent, search agent, customized agent, and personal-status agent. In order to support these agents, a user analyzer is provided to maintain the user profile by analyzing log files and CGI parameters. A site monitor is provided to maintain the site database by monitoring all changes to the site. We also developed a prototype to demonstrate the feasibility of the proposed system architecture. Finally, due to the time limitations, a laboratory experiment was carried out to verify the only value of the customized agent. The value of the agent was confirmed.

Keywords: Intelligent agent, Web browsing, Electronic commerce, Intelligent browsing.

1. Introduction

The World Wide Web has gained amazing popularity through the availability of “point and shoot” browsing tools [9] like Netscape and Explorer. Due to the surprising growth of the Internet population, WWW-based commerce has become a new competitive business tool [5] whether the goals are to increase revenue, to streamline business processes, or to enhance productivity [8].

Compared with the traditional business operation environment, the World Wide Web has many advantages, such as timeliness, worldwide communication, hyper-links, interactive features and multimedia. However, boundless resources and lots of freedom result in several browsing problems, such as getting lost, browsing inefficiently, and missing the most valuable pages or new content. In addition, lack of one-to-one and customized interactive features of traditional sales representatives are the major weaknesses of current WWW-based commerce. In fact, it is critical to have such customized interactive features in order to acquire competitive advantages over the countless Web sites.

There have been several studies on the design of browsing tools on the Web. The most popular browsing tool might be the search engine, which provides an efficient search on the Internet for users. However, some times, users cannot determine the exact keyword for a search. Excite for Web
Servers is a search engine that offers not only retrieval tools, but also an interactive browsing tool that can help users navigate unfamiliar collections of information [4]. Its browsing features include: (1) **Concept-Based Text Searching** which enables users to easily find the documents that interest them using a concept even if they don’t know the exact keywords. (2) **Query-by-Example** allows users to indicate "find more documents like this one" with the click of a button.

Another example is Siteseer. It is a Web page recommendation system based on the organization of an individual’s bookmarks. It recommends pages that have been bookmarked by a user’s virtual neighbors who present the same interests as the user [17]. Siteseer has intrinsic limitations because of its purely collaborative approach. The most notable is its inability to help the first-time user or creating a new category of bookmarks. In addition, if a user wants to enjoy this service, he must first submit his bookmarks.

Compared with the above browsing tools, Web Browser Intelligence (WBI) and Letizia are more intelligent. WBI is a multi-agent system that keeps track of users’ Internet activities and generally helps make Web browsing simpler and easier. It is an implemented system that organizes agents on a user’s workstation to observe user actions, proactively offer assistance and modify Web documents [2]. It can annotate hyperlinks with network speed information, check favorite Web pages for changes, and provide shortcut links for a common path [1].

Letizia is a user interface agent that works with a Web browser, such as Netscape [11]. This agent traces user behavior, such as following links, initiating searches, and requests for help [13]. It is autonomous in that it browses concurrently with the user, searches and analyzes Web pages while the user is browsing, and displays personalized recommendations continually, without explicit user requests or other intervention [12].

Although the browsing tools described above do improve browsing activities in some aspects, they don’t solve all browsing problems. All of the browsing problems encountered by users must be reviewed comprehensively and then a systematic framework for solving all of these possible browsing problems can be proposed. These works will help a Web site to highly satisfy its users, which will lead users to visit it frequently. However, if a Web site wants to provide all users satisfactory service, it should be able to monitor each user and what users find interesting. Only then can a Web site provide appropriate contents to its users. This implies that a Web site should have some intelligent abilities.

Intelligent agents are a solution because intelligent agents are autonomous softbots that can act intelligently based on the monitored environment. Therefore, if there are some kinds of intelligent agents to improve the browsing performances of users, the above problems may be solved. The purpose of this paper is to systematically review all browsing problems and then propose a system architecture for intelligent browsing on the Web. The motivation is to help a Web site gain a competitive advantage over the countless available Web sites.

The rest of this paper is organized as follows. In Section 2, we provide a classification framework for Web browsing activities and then use it as a foundation to review all of the browsing problems and
to explore the required capabilities for a Web site to solve these problems. In Section 3, we propose
the required intelligent capabilities for a Web site for improving browsing activities. Next we
propose a system architecture for intelligent browsing and discuss all of the details of the major
components. In Section 4, a prototype is used to explain the design in detail and demonstrate its
feasibility. We also carried out a laboratory experiment to verify the value of the proposed agents in
Section 5. Finally, we end this paper with conclusions.

2. Classification framework of Web browsing activities and problems

Before discussing online browsing problems, we tried to categorize browsing activities first. We
propose two dimensions, Web familiarity and purpose of browsing. Web familiarity refers to a user’s
familiarity with a Web site. This will affect the way the user browses. The purpose of browsing
refers to the purpose by which a browsing behavior is driven. If there is no purpose, the user may
browse whatever he pleases or randomly. We can classify purpose into two further categories.
The purpose can be either a regular purpose or an ad hoc purpose. For example, a user may have a
habit of reading online news every day. Therefore, to browse for regular purpose may mean that the
user browses a Web site for the same purpose regularly. On the other hand, if a user browses the
New York Times for job hunting, this is an ad hoc purpose because it will occur only when a user
wants to find a job. Table 1 is the proposed classification framework for browsing activities. In the
following subsections, the browsing problems for two types of users will be discussed separately and
then the required capabilities for a Web site will be induced.

Table 1: Classification framework of Web browsing activities

2.1. For users unfamiliar with a Web site

A user that is unfamiliar with a Web site may browse the Web site aimlessly or search some
particular information for an ad hoc purpose. If a user happens to browse a Web site without any
purpose, he may just try each link step by step or randomly to see what the Web site has. This may
cost the user a lot of time to browse all of the hyperlinked pages backwards and forwards continually.
This time-consuming process will become worse if the homepages contain multimedia [15]. In
addition, the user usually quits before he has browsed all of the pages because nothing he has seen
excites him. Furthermore, during the browsing process, he may miss some valuable pages or get lost
due to the hyperlinked structure. For example, in order to know the details of a selected topic, users
usually browse it using a depth-first search. It is then very easy to miss other links at higher or lateral
levels. Being a Web site designer, it is important to help users quickly understand the hyperlinked
structure and contents of the site and keep users interested through the entire browsing process.

There are several possibilities for solving the above problems. First, a Web site can provide an
overview map to help users quickly see the entire site picture. Second, providing a trace map for a
user during his browsing process can avoid the well-known lost in hyperspace. For example,
WebMap updates a 2-dimensional graphical map of a user’s journey by dynamically analyzing his
navigation action [3]. Third, dedicating a window to providing all main options wherever the user is
browsing can help prevent users from getting lost or quitting before browsing all pages. Fourth,
recommending the most often visited pages of that particular Web site to users will not only catch their
attentions quickly but also allow them to become aware of the interesting pages immediately. Fifth,
highlighting the most valuable pages can reduce the risk of those pages being neglected by users.
Finally, providing valuable content is the root of attracting users’ attention all the time.

When a user, unfamiliar with a Web site, browses it on purpose to search for information, this is an
ad hoc purpose. Because the user is not familiar with the locations of the desired information or the
provided search engine, he may have an inefficient search or not be satisfied with the search result.
Here, we propose several helpful solutions. First, users need some intelligent search engines
embedded in the Web site to help them make a more efficient and effective search. For example,
Excite for Web Servers [4] offers concept-based text searching which enables users to restart a search
using synonyms of the original keyword. Second, providing an overview map will give the user a
better idea where the information he is looking for might be. Finally, listing popular search keywords
and hot links can be useful too.

2.2. For users familiar with a Web site

Different from users unfamiliar with the Web site, users familiar with the Web site may visit it very
often, know what the contents are, and even have registered as members. However, for purposeless
browsing, users may have issues similar to those encountered by users unfamiliar with the Web site.
Therefore trace maps or hot pages are also applicable here. But there is a unique issue encountered
by users familiar with the Web site. That is, a user may miss something new due to the complicated
hyperlinks. In order to solve this problem, a widespread design is to highlight new contents.
Because the Web site may have collected browsing activities or personal backgrounds of frequent
users, it is possible to provide tailored homepages for users. For example, a Web site can provide a
user with hot pages that have been browsed by other users with backgrounds similar to the user. The
site can also present all new content that has been changed or added since the user’s last visit. This
should be helpful in making that user become a frequent user. Overall, for the user browsing a
familiar Web site aimlessly, how to catch his attention quickly is an important design issue.

When a user browses a Web site purposely, it may be for a regular or ad hoc purpose. Most of the
time, the regular purpose involves browsing routinely updated homepages, such as to read online news
everyday. The design for an online news site may be kept unchanged for a long time but its contents
change every day or even every half-hour. The possible problems for this type of site may include
losing the customers’ interest because nothing is exciting, inefficient browsing, and missing new content.
Because most Web sites not only update their contents frequently but also change their design quite
often, this implies that a Web site should remind the frequent user with a regular purpose that there is something new while presenting him with regularly updated information. A current browsing agent, Web Browser Intelligence (WBI), can check if the user’s favorite pages have new content and then provide suggestions to the user [2].

When a user visits a site regularly for the same purpose, his browsing behavior may form a pattern that displays a particular browsing path through the site. If the Web site can present a tailored page or filter information for each regular user based on his browsing pattern, it could save the user lots of time and keep his visiting interest. For example, Letizia, a user interface agent [12], can trace “user browsing behavior - following links, initiating searches, requests for help - and tries to anticipate items that may be of interest to him. [13]”

A user with an ad hoc purpose for browsing a Web site may have the same problems as users unfamiliar with the Web site. Therefore, the solutions discussed in the previous subsections will also be useful. For example, when a user sends an online order or posts messages on a board, he may be concerned about the progress afterwards. On these occasions, the Web site should sum up the user’s personal status instead of asking him to jump to different links to check all information. The above discussions were summarized in Tables 2 and 3.

Table 2: Problems of Web browsing
Table 3: Requirements of improving Web browsing

Table 2 lists the problems of Web browsing while Table 3 summarizes the explored requirements for a Web site to improve Web browsing performance. Both tables are based on the proposed classification framework for Web browsing activities in Table 1. This kind of systematic discussion can serve as a foundation to design a Web site which can provide comprehensive help to all kinds of users regardless how frequently they visit the Web site, how familiar they are with it or whether they access it by chance or on purpose.

3. System architecture

Based on Table 3, we can learn that the general required characteristics of a Web site are to provide valuable content, an overview map and allow users have greater browsing flexibility. However, it is not enough to provide an excellent browsing environment for its users unless the Web site has intelligent capabilities, as listed in Table 4.

Table 4: Intelligent capabilities and agents for improving browsing activities

In order to have intelligent capabilities, a Web site has to develop several kinds of intelligent agents. An intelligent agent is a software program that has the capability of autonomous goal-oriented behavior in a heterogeneous computing environment. As in any other software, the basic component is the user
interface [14,18]. In order to have intelligence, it should have a control engine to guide the agent’s behavior based on its knowledge base [7,16]. Therefore, the control engine and knowledge base are the other two major components of an intelligent agent. According to the above discussion, we propose a system architecture for intelligent browsing, as shown in Figure 1.

**Fig. 1. A system architecture for intelligent browsing**

In this architecture, we present five types of application agents to provide intelligent browsing capabilities: a recommendation agent, new-contents agent, search agent, customized agent and personal-status agent. In order to allow these intelligent browsing agents to work well, it is necessary to have a user analyzer to maintain the user profile by analyzing the log file and CGI parameters and a site monitor to maintain the site database by tracing all content changes in the site. We will discuss all of the details of these major components in the following sections.

### 3.1. User analyzer and user profile

The more a sales representative understands his customers, the better service he can provide. In fact, a Web site plays the role of sales representative and all users of the Web site are its potential customers. The ultimate purpose of all proposed agents is to satisfy users. In order to achieve this ultimate purpose, the starting point is to collect and analyze user behavior. The user analyzer and the user profile are designed for this purpose. The user profile is a database that maintains all information about users. The classification system proposed by Laine-Cruzel, et al. [10] was used as a foundation to characterize each user in terms of stable and variable information. Stable information includes all data about individual inherent characteristics and preferences, such as background and agent-guide. Background information is a general description of the user, such as name, gender, interests, and job. The agent-guide consists of instructions given to agents by the user. For example, a user may ask the number of hot links recommended by the agent, not over 5. All stable information will not change often, while variable information consists of the continually changing information involving the user’s purchasing records or online browsing activities. A user’s transactions with the Web site belong to this type of information. Another major type is the log record of users’ browsing behavior.

Log file and CGI parameters are two main data sources for the user profile. Log file records extensive information, such as client IP, server IP, navigation date and time, document name and file size of all browsing activities that occurred on the Web site. This is all recorded in the Web server and becomes the main information source for analyzing online behavior, such as which homepages are visited the most frequently. However, the log file itself is not sufficient for creating a comprehensive user profile. The CGI parameter is another important data source for the user profile. For example, transaction data comes from CGI parameters rather than the log files. The user analyzer is designed to continually analyze log files and CGI parameters to acquire valuable information in order to update the
user profile.

3.2. Site monitor and site database

To satisfy users, understanding them is only a starting point. The ultimate goal is to provide what concerns or interests the users in the user profile. In addition, a Web site should also provide an efficient and effective browsing function and reduce the risk of getting lost or missing the most valuable information. Therefore, we need to have a full understanding of all of the contents in a Web site. The site database is created to serve this function. It stores important information about each homepage. Site contents and site relations are two major types of information in the site database. Site contents refer to the characteristics of a document, such as URL, the last time it was modified, file size, title, summary and keywords. On the other hand, site relations describe the relationship among homepages. The types can be tree, graph or other structures. This is very helpful in assisting search functions.

The site monitor was created to capture site information in order to update the site database. It will capture what the site documents are in terms of site contents and site relationships and detect any changes. An up-to-date site database is another fundamental element to serve the goal of satisfying users.

3.3. Intelligent browsing agents

From Figure 1, we can see that the purpose of the user profile and the site database is to support five kinds of agents: recommendation agent, new-contents agent, search agent, customized agent and personal-status agent. Each agent’s goal and examples of tasks are listed in Table 5. The recommendation agent was designed to give browsing suggestions based on analyses of the behavior of a reference group and site contents while the responsibility of the new-contents agent is to direct users’ attention to new contents. The search agent helps users to conduct an efficient and effective search. To learn the user’s browsing pattern and therefore to present tailored homepages to each user are the goals of the customized agent. Finally, the personal-status agent will provide an up-to-date personal status upon each user’s request.

Table 5: Goal and examples of tasks of intelligent browsing agents

Each agent consists of a user interface, application layer, control engine and knowledge base. The user interface is responsible for communication between the users and the agent. The application layer defines the goals and execution conditions of the agent. When the user interface receives an external message, the application layer will check if the execution conditions match. If the execution conditions match, the control engine will be enabled. Following that, the control engine controls the problem solving process. It may fire appropriate rules that reside in the knowledge base to decide how to guide the user through his browsing process. It will also collect the required CGI parameters during the process. The system then sends the results to the user interface for presentation to the users.
4. Prototyping

In this section, the experience from running the prototype Web site of the Fu-Wen bookstore is described. The Web bookstore is the university bookstore located at National Sun Yat-sen University, Kaohsiung, Taiwan. Its target customers are all teachers and students from the campus. The reasons we chose it as a pilot Web store for developing intelligent customer services were to have more convenient communications due to its location, to provide better services to all members on campus and to provide an opportunity to do field experiments. We developed some applications based on the proposed system architecture for intelligent Web browsing.

The prototype was run on the Windows NT platform. We chose WebSite Professional 2.0 as the Web server because it provided us some examples and modules for developing CGI programs. All CGI programs were written with Visual Basic (VB) 5.0, which could maintain the databases of Microsoft Access 97.

4.1. User analyzer and user profile

Log file and CGI parameters were the two sources for the user profile. Table 6 is an example of the log file and access.log provided by WebSite Professional 2.0. It recorded client IP, server IP, navigation date, navigation time, document name, HTTP protocol and the file size of any browsing activities occurring at the Web site. These records could be used to analyze online behavior, such as the number of hits on each homepage.

Table 6: An example of the log file

However, the log file itself was not enough to provide all of the required information for the user profile. CGI programs were developed to supplement it. The major function of the CGI programs was to collect information about users, activities and the related environment on the Web site by passing on CGI parameters.

The user analyzer was developed to analyze the log file and CGI parameters and update the user profile. It was executed automatically whenever there was new data from the log file or CGI parameters. It then analyzed the data contents and updated the user profile. It deleted log records that were unnecessary or had been maintained for one month.

We collected a member’s background information through a registration-form. In order to become a member, the user had to give his personal background, such as ID number, name, password, gender, address, telephone, e-mail, job, school, department and reading interest.

The user’s background information was only a part of the stable information maintained about him. Other information involved how the user gave instructions to his agents, namely, agent-guide. In the prototype, we allowed users to have the privilege of giving instructions to his agents. Normally, for each type of instruction, there were several options available. For example, the agent-guide of the
recommendation agent included the period of time that the user referred to the agent and the number of recommended links he desired. When a user gave his instructions, the user analyzer would capture these CGI parameters, identify the user and then update the user profile with this data. Regarding variable information, the user analyzer recorded and analyzed all transaction data to study user behavior. The transaction data included orders, book recommendations, book searches and messages, etc.

In addition to transaction data, the log record was another type of variable information. It recorded all browsing activities using the sequence of URLs visited. Since “access.log” did not have information about who had visited the Web, user analyzer utilized the IP address passed by the CGI parameter as a key to identify the user by mapping the log record with the user profile. The person who initiated the log can then be identified. Based on the identification, relevant personal background information can be retrieved and appended to the log record. Meanwhile, meaningless information in the log file is deleted.

4.2 Site monitor and site database

The site monitor was designed to collect site information and update the site database. Due to time and manpower limitation, we chose a full-text search engine applicable to the Chinese environment, Tornado, as a tool instead of writing programs to analyze the homepages of the Web site in order to build the site database. “Tornado” is a product of the Chang-Yang Information Company in Taiwan. The beauty of this is that it could run in a Chinese environment. It could automatically analyze all files and produce keywords, title, summary, and a file name for each file following the path specified by the Web site owner. The results were stored in the site database because some homepages in our Web site were not large enough to form relevant keywords automatically. We manually provided one to three default keywords for these homepages. We used the “Page Info” function of Netscape to collect each homepage’s last time and file size modification. Table 7 shows an example of the site database.

Table 7: An example of the site database

For the purpose of providing intelligent new book recommendations, the site database should include information about books. Whenever we added a new book, we updated the database on book’s title; file name and how many pictures it contained. In addition, we categorized these new books. The book category was used to sort the books.

4.3. Intelligent browsing agents

Up to now, we have only developed partial functions for the search agent, customized agent and personal-status agent. The explanation of how these intelligent browsing agents were designed and managed by describing their agent-guides, control engines and knowledge bases follows:

4.3.1. Search agent
Because Tornado provided most of search functions we required, it was set up as an embedded basic search engine of the Fu-Wen Web site. However, Tornado did not have the learning ability that was necessary for providing an efficient as well as effective search. Therefore, we added an additional function that would learn search keywords from all of the queries requested by users. That is, the agent ranked the frequency of all submitted search keywords whenever there was a search request from users. For each query, the agent would provide the top ten most frequently used keywords for users to choose or users could type other keywords by themselves. Users could also set their search instructions in the agent-guide. The user could also choose the search instructions, which included simple search, synonymous search and tolerable search. The simple search was designed to improve search efficiency when the speed of network transmission was slow. It only displayed the title of each search result. The synonymous search meant that the search would be done with synonymous keywords too. On the other hand, the tolerable search allowed spelling errors. Figure 2 shows the user search interface. Both the agent-guide and the top ten keywords are shown in Figure 2. Figure 3 shows a search result with the keyword, agent. Figure 4 and Table 8 illustrate the algorithm for the search agent.

Fig. 2. Prototype of the search agent  
Fig. 3. A search result of the search agent  
Fig. 4. Algorithm of the search agent  
Table 8: Actions of the search agent

Whenever a user desired to search, the control engine would provide him with up-to-the-minute top ten keywords. After the query was submitted, it would capture the search condition from the user’s agent-guide and then begin the search following the keywords and agent-guide. That is, Tornado would be initiated. The result is presented to the user.

4.3.2. Customized agent

On the Fu-Wen Web site, we presented ten to twelve new books to all users every week. There would be two to five homepages of detailed information for each new book. We kept one hundred new books in the database, based on the rule first in, first out. In order to provide better service, the customized agent was designed to recommend new books to a user based on the analysis of the user’s browsing experiences stored in the user profile. Furthermore, in the prototype, the user could determine the number of books from the most recent browsing experiences used as the analysis base. The prototype also allowed users to set the number of recommended books. For example, if the user set the number of books from the most recent browsing experiences at 50 and the number recommendation books was set at 12, the agent would analyze his most recent fifty browsed books to determine the current reading preferences. The system would then retrieve twelve books that most satisfied the user’s interest. Figure 5 displays a customized book recommendation homepage that would change whenever the user browsed a book.
We briefly describe the algorithm for the customized agent in Figure 6 and Table 9. This agent first identifies the user and retrieves the types of books being browsed. It then analyzes the user’s browsing pattern based on the agent-guide to user preferences. Following that, the books which most satisfy the agent-guide are retrieved and presented to the user. When the user browses a new book from the recommended books, this agent records the browsing action and presents details of the book. This process would repeat until the user quits the intelligent new book recommendation homepage.

4.3.3. Personal-status agent

In the prototype, the personal status included four kinds of information: new book, order, message and accumulated consumption. “New book” referred to books suggested by the recommendation agent. “Order” was related to the current status of the user’s order. “Message” regarded all messages posted or replied to by the user on the bulletin board. “Accumulated consumption” indicated the amount of time that the user spent in the Fu-Wen bookstore. The Personal-status agent helped users to quickly browse all up-to-date personal status information. Without the Personal status agent, the user would have to go to different places to find this information. Viewing numerous sites would make it difficult for the user to get a whole picture of his personal status. The user could make his choice by setting the agent-guide. “Agent task” allows the user to choose the kinds of personal information in which he is interested. The other option is “time period” in which the user can set the time period for the trace back. Figure 7 shows an example. In this example, the user, tcyang, requested three agent tasks: “order”, “message” and “accumulated consumption”. The time period was set as two weeks.
5. Experiment

5.1. Research framework and hypotheses

It is necessary to verify the value of the proposed agents. Hence, we carried out a laboratory experiment to test the customized agent against a time limitation. The research framework is illustrated in Figure 10. There was only one independent variable for this experimental research, namely, the customized agent provision. The dependent variable was “performance”, which was described by the browsing effectiveness and efficiency and user satisfaction. There were two other variables that may interfere with the dependent variable. The so-called interfering variables included familiarity with the WWW and presentation style. Familiarity with the WWW was controlled by randomization. The presentation style was the same for all subjects. Because the customized agent provides tailored service to the user’s preferences, it should reveal a better performance. Therefore the following hypotheses are proposed.

**Fig. 10. Research framework**

H1: The customized agent will allow users a more effective as well as efficient browsing.
H2: The customized agent will increase user satisfaction.

5.2. Experimental design

In the experiment, the system recommended twelve books, two books from each category, to the user in each round. There were six categories consisting of travel, humanities, biography, computers, management, and science. The task of the customized agent was to recommend new books according to a user’s up-to-date reading preferences, which were modified continually based on the user’s previous recommended book browsing behavior. Furthermore, the sequence of displayed recommended books began from the most preferred to the least preferred. The main test of this experiment was to verify the value of the customized agent. However, there is conflict between customized service and individual privacy. The better the customized service, the less the degree of privacy. Therefore, to expose users to customized service is a controversial issue. This resulted in three different treatments. All treatments recommended twelve books. For the subjects in the Group 1, the books were presented to the user without customized agent support. That is, they were displayed randomly. For subjects in both Groups 2 and 3, books were displayed with customized agent support. However, subjects in Group 3 were informed about the customized agent support and subjects in Group 2 were not. There were 110 subjects. All subjects were undergraduate students.
of the Department of Information Management at National Sun Yat-sen University in Taiwan. Each student was randomly assigned to a group in order to reduce the possible impacts of interfering variables to the minimum.

The experiment was conducted in three phases. During Phase 1, subjects filled out a personal background questionnaire. In the questionnaire, the subjects were required to list their reading preferences (1: the most preferred, 6: the least preferred) on the six book categories.

During Phase 2, there were five rounds. In each round, the subjects were required to browse four books among the twelve recommended books. For each book, there were four detailed pages available. The user could browse as many pages as desired. All browsed books were moved to the end of the twelve-book recommendation list in the next presentation.

Different groups got a different new book display. For Group 1, the books were displayed randomly. For Groups 2 and 3, the agent presented the books based on the user’s up-to-date reading preferences that the agent learned from the user’s previous browsing behavior. The more preferred books were placed at the front positions. That is, the book positions also reflected the user’s reading preferences. As stated in the above discussion, the subjects in Group 3 were aware of the customized agent.

In each round, in addition to browsing four books, the user had to order a book. At the end of the experiment, every user drew a lot to see if he could be rewarded with a book from among the four books he ordered. The purpose of this design was to stimulate a real shopping attitude. During Phase 3, subjects answered a post-browsing questionnaire about how they felt about this system.

5.3. Statistical analysis and experimental results

5.3.1. Statistical analysis

For the data analysis, SPSS for Windows 7.0 was used. ANOVA and Bonferroni multiple comparisons were adopted to test the differences in mean among multiple groups.

Among 110 participants, there were 101 useful subjects. The sizes of each group were 34 for Group 1, 33 for Group 2, and 34 for Group 3. The demographics associated with the subjects are detailed in Table 11. The subjects were typically males who had one to three years of Web experience. Over 70 percent of the subjects spent ten hours or less on the Web every week. Most of them had not bought anything before on the Web before this experiment. This is similar to the demographics of the Internet population [6] and therefore the experimental results can be generalized.

Table 11: Demographics associated with the subjects

5.3.2. Experimental results

(1) Browsing effectiveness and efficiency

To test hypothesis H1, the positions of all books clicked by users were recorded. Since the system
displayed twelve books in each round, the positions were represented by 1, 2..., 12 separately. The greater the number of times this book was displayed, the smaller its position number. The ANOVA table in Table 12 shows there existed significant differences among all groups and therefore we further used Bonferroni multiple comparisons to compare the mean between every two groups among the three groups. Table 13 reveals that mean of browsing positions in Group 1 was significantly higher as compared with either Group 2 or Group 3. However, the difference between Group 2 and Group 3 was not significant.

Table 12: Browsing position – single factor analysis of variance
Table 13: Browsing position – multiple comparisons

The books were displayed randomly for Group 1. The books were displayed according to the user’s up-to-date reading preferences for both Group 2 and Group 3. In other words, for Group 2 and Group 3, the position of a book also reflected the user’s reading preference. Therefore, compared with Group 2 and Group 3, the browsing position mean was significantly greater for Group 1, implying that subjects tended to browse books based on their reading preferences rather than book positions. Because the customized agent was to recommend books based on the user’s up-to-date preferences, it induced the effectiveness of the agent. Furthermore, because the customized agent displayed the more preferred books at the front positions, browsing was more efficient for users with the support of the customized agent. In sum, H1 is confirmed.

We also tested the hypothesis H1 for each round using ANOVA. Except for the fourth round, groups in all of the other rounds had significant differences and therefore Bonferroni multiple comparisons were done for these four rounds. The browsing position mean was significantly greater for Group 1 than the other two groups in the first, second and fifth rounds. In the third round, Group 1 was also significantly greater than Group 1 than Group 3. Figure 11 shows that users in Group 2 and Group 3 tended to click the books at the more front positions. This implies not only that users clicked books matching their reading preferences but also that the customized agent provided a more efficient browsing. There were no significant differences between Group 2 and Group 3 in each round.

Fig. 11. Comparisons of average browsing position

(2) User satisfaction

The subject satisfaction level was measured with a 5-point scale. “1” was the most preferred and “5” was the least preferred. The result is depicted in Figure 12. Four questions were concerned with recommendation priority. Group 2 got the highest satisfaction (mean=9.24), Group 3 was the second (mean=9.26), and Group 1 was the third (mean=9.35). Another question about the overall recommendation method revealed the same result. Group 2 was better (mean=2.15) than Group 3 (mean=2.32) and Group 1 (mean=2.53). Hence H2 was supported although the support was not
significant. One of the possible reasons might be that subjects in Group3 had higher expectations than Group 2 due to their awareness of the customized agent support. Therefore they might have comparatively lower satisfaction. Another reason for the lack of significant difference might be that there were not enough rounds.

Fig. 12. User satisfaction level

From the above analysis, the value of the customized agent was verified. In fact, another related research on customized advertising [19] also demonstrated that customized Web advertising achieves higher advertising effectiveness than non-customized advertising.

6. Conclusions

In order to acquire competitive advantages over the countless number Web sites, a challenge is to allow the users of a Web site to enjoy the browsing process and subsequently become frequent visitors. This paper not only comprehensively summarized all of the browsing problems, but also proposed a system architecture for intelligent browsing. This architecture has four principle aspects.

First, it provides a classification framework for Web browsing activities using two dimensions: Web familiarity and browsing purpose. This can serve as a foundation to understand Web browsing activities and problems.

Second, it proposes a system architecture for intelligent browsing on the Web. In this architecture, the user analyzer and the site monitor are designed to maintain the user profile and the site database separately. The user profile and site database support five kinds of agents: recommendation agent, new-contents agent, search agent, customized agent and personal-status agent. Each agent has a control engine and a knowledge base.

Third, three browsing agent prototypes were developed to demonstrate the feasibility of the proposed architecture. We described the details of the control engine and knowledge base of each agent. Meanwhile, it also shows the way to maintain and utilize the user profile and the site database.

Finally, a laboratory experiment for verifying the value of the customized agent was carried out. The value of the customized agent was confirmed. It provides effective as well as efficient browsing for users.

There are some promising issues for future research. We hope to develop a more intelligent agent. How to apply Artificial Intelligence technology would also be a worthwhile research issue. How the different agents communicate with and support one another is another promising research area. For this research, a common communication language and coordination mechanism must be developed.

Third, all of the agents proposed are passive agents because none will operate unless users visit the Web site. To be more proactive, agents that can push messages to users are worthwhile. Finally, more experiments should be done in order to verify the value of the agents and how the agents should be designed.
7. Acknowledgements

This material is based on work supported by National Science Council, R. O. C. under Grant No. NSC-86-2416-H-110-011.

8. References


### Table 1
**Classification framework of Web browsing activities**

<table>
<thead>
<tr>
<th>Purpose of browsing</th>
<th>Web familiarity</th>
<th>unfamiliar</th>
<th>familiar</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purposeless browsing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Purposeful Browsing</td>
<td>regular</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ad hoc</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 2
**Problems of Web browsing**

<table>
<thead>
<tr>
<th>Purpose of browsing</th>
<th>Web familiarity</th>
<th>unfamiliar</th>
<th>familiar</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purposeless browsing</td>
<td></td>
<td>• waste time</td>
<td>• waste time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• get lost</td>
<td>• get lost</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• miss the most valuable pages</td>
<td>• miss the most valuable pages</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• quit before browsing all pages</td>
<td>• lose interest due to nothing new or exciting</td>
</tr>
<tr>
<td></td>
<td>regular</td>
<td>• browse inefficiently</td>
<td>• lose interest due to less value of content</td>
</tr>
<tr>
<td></td>
<td>ad hoc</td>
<td>• have inefficient and ineffective search</td>
<td>• have unsatisfied search result</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• have unsatisfied search result</td>
<td>• have unsatisfied search result</td>
</tr>
</tbody>
</table>

### Table 3
**Requirements of improving Web browsing**

<table>
<thead>
<tr>
<th>Purpose of browsing</th>
<th>Web familiarity</th>
<th>unfamiliar</th>
<th>familiar</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purposeless browsing</td>
<td></td>
<td>• overview map</td>
<td>• overview map</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• trace map</td>
<td>• trace map</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• a dedicated window to main options</td>
<td>• a dedicated window to main options</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• hot link recommendation</td>
<td>• hot link recommendation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• valuable contents</td>
<td>• valuable contents</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• flexibility</td>
<td>• flexibility</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• highlight of the most valuable links</td>
<td>• highlight of the most valuable links</td>
</tr>
<tr>
<td></td>
<td>regular</td>
<td>• tailored homepages</td>
<td>• highlight of new contents</td>
</tr>
<tr>
<td></td>
<td>ad hoc</td>
<td>• highlight of new contents</td>
<td>• information filtering</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• intelligent search engine</td>
<td>• intelligent search engine</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• default keyword list</td>
<td>• default keyword list</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• hot link recommendation</td>
<td>• hot link recommendation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• personal current status</td>
<td>• highlight of new contents</td>
</tr>
</tbody>
</table>
Table 4
Intelligent capabilities and agents for improving browsing activities

<table>
<thead>
<tr>
<th>Required Intelligent capabilities</th>
<th>Required intelligent agents</th>
</tr>
</thead>
<tbody>
<tr>
<td>For users unfamiliar With the Web site</td>
<td></td>
</tr>
<tr>
<td>• Recommend hot links</td>
<td>→ recommendation agent</td>
</tr>
<tr>
<td>• Highlight the most valuable contents</td>
<td></td>
</tr>
<tr>
<td>• Provide default search keywords</td>
<td>→ search agent</td>
</tr>
<tr>
<td>• Provide intelligent search</td>
<td></td>
</tr>
<tr>
<td>For users familiar With the Web site</td>
<td></td>
</tr>
<tr>
<td>• Recommend hot links</td>
<td>→ recommendation agent</td>
</tr>
<tr>
<td>• Highlight the most valuable contents</td>
<td></td>
</tr>
<tr>
<td>• Provide default search keywords</td>
<td>→ search agent</td>
</tr>
<tr>
<td>• Provide intelligent search</td>
<td></td>
</tr>
<tr>
<td>• Highlight new contents</td>
<td>→ new-contents agent</td>
</tr>
<tr>
<td>• Provide customized contents</td>
<td>→ customized agent</td>
</tr>
<tr>
<td>• Monitor personal status</td>
<td>→ personal-status agent</td>
</tr>
</tbody>
</table>

Table 5
Goal and examples of tasks of intelligent browsing agents

1. Recommendation agent

<table>
<thead>
<tr>
<th>Goal</th>
<th>Give browsing suggestions by analyzing behavior of reference group and site content.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examples of task</td>
<td>Retrieve log file and count the hits of each homepage; Recommend hot links based on the rank of hits of homepages from all users; Recommend different hot links to different types of users based on the rank of hits of homepages from users of same type; Highlight the most valuable content for all users based on the judgment of the Web site owner; Analyze the interests of different types of users and then present a tailored highlight of the most valuable content to each user according to his type.</td>
</tr>
</tbody>
</table>

2. New-contents agent

<table>
<thead>
<tr>
<th>Goal</th>
<th>Catch users’ attentions to new contents.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examples of tasks</td>
<td>Retrieve new contents from the site database and highlight them for general users; Based on the user profile, retrieve new contents from the site database which is new for that user and present him a tailored highlight of new contents.</td>
</tr>
</tbody>
</table>

3. Search agent

<table>
<thead>
<tr>
<th>Goal</th>
<th>Provide efficient and effective search.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examples of tasks</td>
<td>Provide keywords list; Give suggestion of precise keywords to the user based on his query; Learn the user’s search behavior as basis of providing suggestions for his next search; Provide default keywords for search.</td>
</tr>
</tbody>
</table>

4. Customized agent

<table>
<thead>
<tr>
<th>Goal</th>
<th>Present customized homepages based on the analysis of the user’s browsing pattern.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examples of tasks</td>
<td>Analyze the user profile such as hits of each homepage, personal status, and background to find his browsing pattern and preference; Analyze the site database such as each homepage’s hits, change frequency, size, keywords; Apply AI algorithm to the analysis of the user profile and the site database to provide a tailored homepage for each user.</td>
</tr>
</tbody>
</table>

5. Personal-status agent

| Goal | Provide up-to-date personal status for each user. |
Examples of tasks
Monitor all transactions of Web site and update the related user profile;
Summary up-to-date personal status in a single homepage upon the user’s request.

Table 6
An example of the log file
140.117.95.176 140.117.75.61-[07/Jul/1997:08:15:59 -0800] "GET /bookstore/login.htm HTTP/1.0" 200 3056
140.117.95.176 140.117.75.61-[07/Jul/1997:08:16:00 -0800] "GET /Picture/Backgnd2.jpg HTTP/1.0" 200 9753
140.117.74.176 140.117.75.61-[07/Jul/1997:10:33:11 -0800] "GET /cgi/agent1b.exe HTTP/1.0" 200 830
203.66.35.34 140.117.75.61-[07/Jul/1997:10:52:55 -0800] "GET /hotnew4a.htm HTTP/1.0" 200 2856
203.66.35.34 140.117.75.61-[07/Jul/1997:10:52:59 -0800] "GET /Picture/Hotnew6.jpg HTTP/1.0" 200 4478
203.66.109.55 140.117.75.61-[08/Jul/1997:00:31:55 -0800] "GET /bookstor/picture/first1.gif HTTP/1.0" 200 53 782
140.117.110.207 140.117.75.61-[18/Jul/1997:05:49:58 -0800] "GET /cgi/checkord.exe HTTP/1.0" 200 290

Table 7
An example of the site database
<table>
<thead>
<tr>
<th>File-Name</th>
<th>Words</th>
<th>Graphs</th>
<th>Links</th>
<th>Level</th>
<th>Modified Date</th>
<th>Click</th>
<th>Title</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>award.htm</td>
<td>1989</td>
<td>7</td>
<td>17</td>
<td>3</td>
<td>4/23/97</td>
<td>8</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>award3.htm</td>
<td>1651</td>
<td>6</td>
<td>17</td>
<td>1</td>
<td>4/28/97</td>
<td>10</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>board.htm</td>
<td>1993</td>
<td>9</td>
<td>22</td>
<td>3</td>
<td>3/15/97</td>
<td>5</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Table 8
Actions of the search agent
<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
</table>
| ①     | provide top ten keywords  
• Rank the frequencies of all submitted search keywords.  
• Present the top ten keywords for user’s reference. |
| ②     | capture search condition  
• Retrieve the keyword and agent-guide.  
• Add the keyword to the user profile. |
| ③     | initiate Tornado to start search  
• Create the query syntax following the related syntax rules.  
• Send the query syntax to “Tornado” for search. |
| ④     | display search results to user interface  
• Retrieve the site database to get the title, URL link, and number of the keyword occurrence of searched pages.  
• Summary the search result for users. |

Table 9
Actions of the customized agent
<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
</table>
| ①     | When the user asks system to recommend books  
• Catch IP address from CGI parameters when there is a client request.  
• Decode who is the user by matching the IP and the user’s ID in the user profile. |
| ②     | retrieve agent-guide  
• Retrieve the user’s agent-guide. |
| ③     | compute and sort the book category  
• Category the books browsed by the user recently.  
• Rank the book category based on the number of browsed books of each category. |
| ④     | show the results and send them to the user interface  
• Choose the top one category.  
• Retrieve a new book of the category that the user hasn’t browsed from the site database. |
(3) Put the new book to the recommendation list.
(4) Repeat step 2 to 3 until there is not any book belonging to the category, and then choose the next preferred category.
(5) Repeat step 2 to 4 until the number of books in the recommendation list is equal to the number limit assigned in agent-guide or there is no new book in the database.
(6) Present the recommended books to the user.

When the user chooses one book to browse

1. Identify who is the user
   - Catch IP address from CGI parameters when there is a client request.
   - Decode who is the user by matching the IP and user’s ID in the user profile.

2. Capture browsing action
   - Identify which book is browsed.

3. Check the category of the browsed book
   - Retrieve the book category from the site database
   - Add one to the number of browsed book in that book category and update the browsing experiences of the user profile
   - Mark the book been read.

4. Show the book page chosen to the user interface
   - Present the cover of book and other detailed links to the user.

Table 10
Actions of the Personal-status agent

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1. Identify who is the user | • Catch IP address from CGI parameters when there is a client request.  
   • Decode who is the user by matching the IP and user’s ID in the user profile. |
| 2. Retrieve agent-guide | • Retrieve agent task and time period from the user’s agent-guide. |
| 3. Detect whether the status has changed | For each task assigned in agent-guide  
   • Fire appropriate status-updated rules.  
   [new book rules] Check the user’s the most recent book browsing records.  
   [message rules] Check messages that the user posted or replied.  
   [accumulated consumption rules] Check the updated accumulated consumption.  
   • Detect whether the status of the required tasks has changed within the time period. |
| 4. Summarize results and send to the user interface | • Summarize the results on the homepage. |

Table 11
Demographics associated with the subjects

<table>
<thead>
<tr>
<th>Sample size: 101</th>
<th>Hours of using Web every week:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5 hours</td>
</tr>
<tr>
<td></td>
<td>43 (42.6%)</td>
</tr>
<tr>
<td>Gender:</td>
<td></td>
</tr>
<tr>
<td>Male</td>
<td>74 (73.3%)</td>
</tr>
<tr>
<td>Female</td>
<td>27 (26.7%)</td>
</tr>
<tr>
<td>Age*:</td>
<td></td>
</tr>
<tr>
<td>18-23 years old</td>
<td>32 (31.7%)</td>
</tr>
<tr>
<td>24-29 years old</td>
<td>26 (25.7%)</td>
</tr>
<tr>
<td>≥ 30 years old</td>
<td>43 (42.6%)</td>
</tr>
<tr>
<td>Major place of using Web:</td>
<td></td>
</tr>
<tr>
<td>School</td>
<td>30 (29.7%)</td>
</tr>
<tr>
<td>Home</td>
<td>40 (39.6%)</td>
</tr>
<tr>
<td>Company</td>
<td>31 (30.7%)</td>
</tr>
<tr>
<td>Years of Web experience:</td>
<td></td>
</tr>
<tr>
<td>&lt; 1 year</td>
<td></td>
</tr>
<tr>
<td>1-2 years</td>
<td>34 (33.7%)</td>
</tr>
<tr>
<td>2-3 years</td>
<td>25 (24.8%)</td>
</tr>
<tr>
<td>Online shopping experience:</td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>12 (11.9%)</td>
</tr>
<tr>
<td>No</td>
<td>89 (88.1%)</td>
</tr>
</tbody>
</table>
Table 12
Browsing position – single factor analysis of variance

<table>
<thead>
<tr>
<th>Source of variation</th>
<th>Sum of Squares</th>
<th>df</th>
<th>Mean Square</th>
<th>F</th>
<th>Sig.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Between Groups</td>
<td>28.384</td>
<td>2</td>
<td>14.192</td>
<td>17.687</td>
<td>.000*</td>
</tr>
<tr>
<td>Within Groups</td>
<td>78.634</td>
<td>98</td>
<td>.802</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>107.018</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* The mean difference is significant at the .05 level.

Table 13
Browsing position – multiple comparisons

<table>
<thead>
<tr>
<th>(I) GROUP</th>
<th>(J) GROUP</th>
<th>Mean Difference (I-J)</th>
<th>Std. Error</th>
<th>Sig.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Browsing Location</td>
<td>1</td>
<td>2</td>
<td>.9560</td>
<td>.219</td>
</tr>
<tr>
<td>Browsing Location</td>
<td>1</td>
<td>3</td>
<td>1.2324</td>
<td>.217</td>
</tr>
<tr>
<td>Browsing Location</td>
<td>2</td>
<td>3</td>
<td>.2763</td>
<td>.219</td>
</tr>
</tbody>
</table>

* The mean difference is significant at the .05 level.

---

* Some of the subjects were part-time students who are older than normal full-time students.

Fig. 1. A system architecture for intelligent browsing
Fig. 2. Prototype of the search agent

Fig. 3. A search result of the search agent

Fig. 4. Algorithm of the search agent
Fig. 5. Prototype of the customized agent - book recommendation

Fig. 6. Algorithm of the customized agent

Fig. 7. Prototype of setting up the Agent-guide
Fig. 8. Algorithm of the Personal-status agent

Fig. 9. Prototype of the Personal-status agent

Fig. 10. Research framework
Fig. 11. Comparisons of average browsing position

*1: The mean of Group 1 is significantly greater than both Group 2 and Group 3 at the .05 level.
*2: The mean of Group 1 is significantly greater than Group 3 at the .05 level.

Fig. 12. User satisfaction level

Hsiangchu Lai is an Associate Professor of Information Management at National Sun Yat-sen University in the Republic of China. Lai received her Ph.D. in management information systems from Purdue University. Her research interests include electronic commerce, negotiation support systems, and strategic information technology. She has published in IEEE Computer, Group Decision and Negotiation, Journal of Information Systems, Journal of Computer Information Systems, Sun Yat-sen Management Review, and MIS Review. Lai is a member of the IEEE computer Society, ACM, ISDSS, and DSI.
Tzyy-Ching Yang is a doctoral candidate of Information Management at National Sun Yat-sen University in the Republic of China. He received his master degree in management from Yuan-Ze Institute of Technology, Taiwan. His research interests include Web advertisement, intelligent agents and customized marketing in electronic commerce. He has published in MIS Review, Journal of Information Management.