Web Montage: A Dynamic Personalized Start Page

Corin R. Anderson
University of Washington
Seattle, WA, USA
corin@cs.washington.edu

Eric Horvitz
Microsoft Research
Redmond, WA, USA
horvitz@microsoft.com

ABSTRACT

Despite the connotation of the words "browsing" and "surfing," web usage often follows routine patterns of access. However, few mechanisms exist to assist users with these routine tasks; bookmarks or portal sites must be maintained manually and are insensitive to the user's browsing context. To fill this void, we designed and implemented the MONTAGE system. A web montage is an ensemble of links and content fused into a single view. Such a coalesced view can be presented to the user whenever he or she opens the browser or returns to the start page. We pose a number of hypotheses about how users would interact with such a system, and test these hypotheses with a fielded user study. Our findings support some design decisions, such as using browsing context to tailor the montage, raise questions about others, and point the way toward future work.

Keywords

Personalization, user modeling, adaptive user interfaces, adaptive web sites

1. INTRODUCTION

Despite the exploratory ring of the terms "browsing" and "surfing," web usage often follows routine patterns of access. For example, a graduate student might read a web-based newspaper the first thing in the morning, then spend a few hours on software development, with intermittent consultation of online programming documentation. Following a break at noon for lunch and to read comics on the web, the student might return to programming, then take a mid-afternoon break to check news and a few more comics, and finally consult online transit information shortly before leaving at 5:30 P.M. Such stereotypical patterns of web access are common. However, despite the regularity with which users view content, few mechanisms exist to assist with these routine tasks. Lists of bookmarks must be authored and maintained manually by users and are presented in a cumbersome hierarchical menu. Links and content on personalized portals, such as MSN.com [12] or MyYahoo! [17], are more easily navigable, but still must be chosen and managed by users in an explicit manner.

The challenge that we tackled—and in turn share with the web research community—is to develop tools that assist users with *routine web browsing*. Routine web browsing refers to patterns of web content access that users tend to repeat on a relatively regular and predictable basis (for example, pages viewed at about the same time each day, or in the same sequence, or when working on the same task, *etc.*). In developing a response to this challenge, we pose the following three hypotheses:

- **Hypothesis 1**: Users want "one-button access" to their routine web destinations (*i.e.*, users want to minimize their effort in retrieving and viewing content);
- **Hypothesis 2**: Tools for routine web browsing are enhanced by tailoring links and views to a user's current browsing context (as opposed to displaying a static set of content under all circumstances); and
- **Hypothesis 3**: Past web access patterns can be successfully mined to predict future routine web browsing.

To test these hypotheses, we designed and implemented the MONTAGE system. MONTAGE builds personalized web portals for its users, based on models mined from users' web usage patterns. These portals both link to web resources as well as embed content from distal pages (thus producing a *montage* of web content; see Figure 1). MONTAGE melds concepts from research on predictive user modeling (for web content prefetching [6, 7, 14] or adaptive web sites [1, 15]) with the user interface approaches of automatic bookmarking [10, 11] or web portal sites [12, 17]. After completing the MONTAGE prototype, we fielded the system to evaluate its effectiveness as a tool and to explore the validity of our hypotheses.

In evaluating the hypotheses, we explored methods and constructed a prototype that make the following contributions:

- **Multiattribute utility model for web information access.** We leverage a utility model for prioritizing web content that considers both savings in navigation cost and informational value.
- **Enriched informational context for user modeling.** We demonstrate learning and use of predictive models that leverage a notion of context captured by a rich set of observations, including evidence about the time of day, the time since last access, and the *recent pattern of topics* of information at a user's focus of attention.
- **Automated segmentation and structuring by topic.** We harness a real-time topic classifier to segment and

WWW2002, May 7–11, 2002, Honolulu, Hawaii, USA.
ACM 1-58113-449-5/02/0005.

organize content within and across personalized portal pages, providing users with views on personalized information that is structured by a topic ontology.

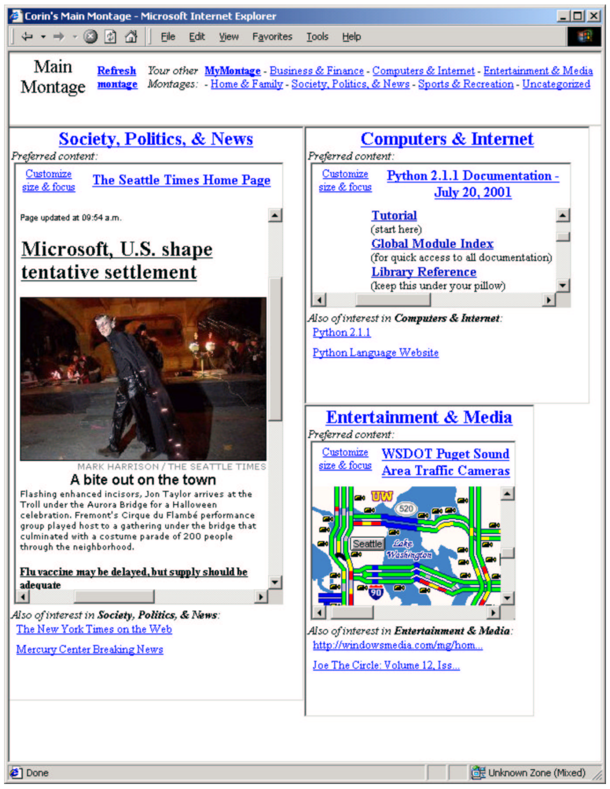
- **Persistent configurable information lenses.** We show how a system can compose personal portal pages by assembling user-configurable clippings from parent pages that provide persistent views or "lenses" onto the parent pages.
- **Layout of content based on expected utility.** We use an expected-utility search procedure to both select the best subsets of content to display and to dynamically position this content on personal portal pages.
- **Construction and validation of MONTAGE personalized portal system.** We weave together multiple components to build a working MONTAGE prototype system that runs on a proxy or on a client, and we perform a user study to explore the validity of several hypotheses.

In the next section, we present the overall challenge in more detail, followed by an outline of MONTAGE in Section 3. We discuss implementation specifics of MONTAGE in Section 4 and present our experimental findings in Section 5. Section 6 compares MONTAGE to related work and section 7 concludes with a brief summary.

2. ROUTINE BROWSING

As we noted earlier, not all web usage is random or novel; web users also tend to revisit sites and pages in a regular, predictable manner. In the example of the graduate student presented at the beginning of the paper, the pages the student viewed depended entirely on his current *context*, where context is taken as the time of day and the general topic of the pages viewed previously. More generally, we define the context of a web browsing session as the set of attributes that influences (either consciously or subconsciously) the selection of pages a user views in the next session. Many factors can be included in a formalization of context. For instance, the context can include the time of day, the period of time elapsed since the last session, the general topic of the last session, the most recent non-browsing computer activity (*e.g.*, the most recently viewed e-mail message), *etc.* We define routine web browsing as the overall pattern of content access that a user performs whenever in the same or similar contexts. For example, if a user reviews a stock portfolio at around 1:15 P.M. every day, then viewing the stock portfolio is a routine behavior—it happens at about the same time each day. On the other hand, if one day the user spends an hour searching for information about fishing in the Pacific Northwest, then this behavior is not routine because the user does not repeat it in a similar context.

Our intuition strongly suggests that routine browsing is a common mode for interacting with the web; our informal surveys early in our study suggested that many people tend to view the same page or constellation of pages when in similar contexts. The more important questions that must be answered are whether we can formulate good notions of context, identify the routine browsing associated with each context, and leverage this information to assist the user. These questions are embodied in the hypotheses we posed in the introduction and are answered with our experiments with the MONTAGE system.


Corin's Main Montage - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Done
Unknown Zone (Mixed)
UW
520
Seattle
Lake Washington
90

Main Montage Refresh montage Your other Montages: MyMontage - Business & Finance - Computers & Internet - Entertainment & Media - Home & Family - Society, Politics, & News - Sports & Recreation - Uncategorized

Society, Politics, & News
Preferred content:
Customize size & focus The Seattle Times Home Page
Page updated at 09:54 a.m.
Microsoft, U.S. shape tentative settlement
MARK HARRISON / THE SEATTLE TIMES
A bite out on the town
Flashing enhanced incisors, Jon Taylor arrives at the Troll under the Aurora Bridge for a Halloween celebration. Fremont's Cirque du Flambé performance group played host to a gathering under the bridge that culminated with a costume parade of 200 people through the neighborhood.
Flu vaccine may be delayed, but supply should be adequate
Also of interest in ***Society, Politics, & News****:*
The New York Times on the Web
Mercury Center Breaking News

Computers & Internet
Preferred content:
Customize size & focus Python 2.1.1 Documentation - July 20, 2001
Tutorial
(start here)
Global Module Index
(for quick access to all documentation)
Library Reference
(keep this under your pillow)
Also of interest in ***Computers & Internet****:*
Python 2.1.1
Python Language Website

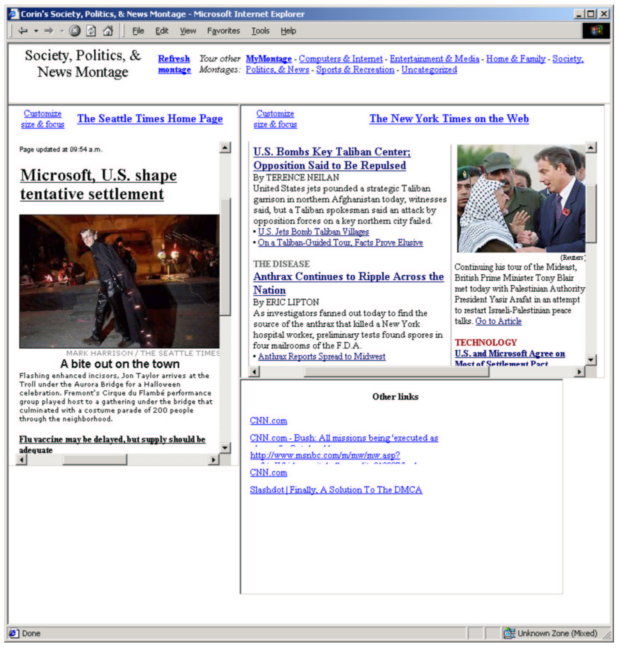
Entertainment & Media
Preferred content:
Customize size & focus WSDOT Puget Sound Area Traffic Cameras
Also of interest in ***Entertainment & Media****:*
http://windowsmedia.com/mg/hom...
Joe The Circle: Volume 12, Iss...

Figure 1: Main montage. Content is grouped by topic, and each topic heading links to a topic-specific montage.

3. THE MONTAGE SYSTEM

A web montage is a page that offers "one-stop shopping" for users to find the information they want. A montage combines content from many different pages, linking to pages or embedding distal content, saving the user the need to follow even a single link to view content. Three typical montage views are shown in Figures 1, 2, and 3. In Figure 1 is the "Main Montage" which displays links and embedded content grouped by topic. This particular montage has three topic-specific panes: "Society, Politics, & News," "Computers & Internet," and "Entertainment & Media," each containing a cropped view of a distal web page and links to other pages within the respective topic. Thus, the user can immediately view the afternoon's current news, the user's most frequently-viewed programming documents, and the current traffic conditions around the area. In Figure 2 is a topic-specific montage, which shows several embedded pages and related links. Both the topic-specific montage and the main montage are assembled automatically to fit just within the user's current browser window, to eliminate the need to scroll the page. The user can navigate between these montages by following links at the top of each view. Figure 3 shows a simplified montage that contains only links. Still other visualizations are possible: a browser toolbar showing iconic representations of pages; a persistent display of bookmarks auxiliary to the browser window; *etc.* We plan to explore these alternatives in future work.

Building a web montage is a two-step process, similar to that used in PROTEUS [1]. In the first step, the MONTAGE


Corin's Society, Politics, & News Montage - Microsoft Internet Explorer
File Edit View Favorites Tools Help

Society, Politics, & News Montage

Refresh montage

Your other Montages: **MyMontage** - Computers & Internet - Entertainment & Media - Home & Family - Society, Politics, & News - Sports & Recreation - Uncategorized

Customize size & focus

The Seattle Times Home Page

Page updated at 09:54 a.m.

Microsoft, U.S. shape tentative settlement

MARK HARRISON / THE SEATTLE TIMES

A bite out on the town

Flashing enhanced incisors, Jon Taylor arrives at the Troll under the Aurora Bridge for a Halloween celebration. Fremont's Cirque du Flambé performance group played host to a gathering under the bridge that culminated with a costume parade of 200 people through the neighborhood.

Flu vaccine may be delayed, but supply should be adequate

Customize size & focus

The New York Times on the Web

U.S. Bombs Key Taliban Center; Opposition Said to Be Repulsed

By TERENCE NEILAN

United States jets pounded a strategic Taliban garrison in northern Afghanistan today, witnesses said, but a Taliban spokesman said an attack by opposition forces on a key northern city failed.

• U.S. Jets Bomb Taliban Villages

• On a Taliban-Guided Tour, Facts Prove Elusive

THE DISEASE

Anthrax Continues to Ripple Across the Nation

By ERIC LIPTON

As investigators fanned out today to find the source of the anthrax that killed a New York hospital worker, preliminary tests found spores in four mailrooms of the F.D.A.

• Anthrax Reports Spread to Midwest

(Reuters)

Continuing his tour of the Mideast, British Prime Minister Tony Blair met today with Palestinian Authority President Yasir Arafat in an attempt to restart Israeli-Palestinian peace talks. Go to Article

TECHNOLOGY

U.S. and Microsoft Agree on Most of Settlement Pact

Other links

CNN.com

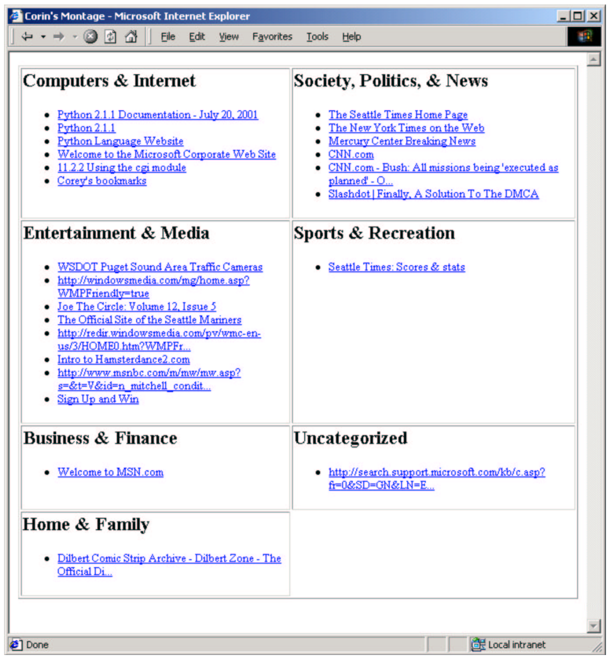
CNN.com - Bush: All missions being 'executed as

http://www.msnbc.com/m/mw/mw.asp?

CNN.com

Slashdot | Finally, A Solution To The DMCA

Figure 2: A topic-specific montage. This montage embeds content from several pages.


Corin's Montage - Microsoft Internet Explorer
File Edit View Favorites Tools Help

Computers & Internet

- Python 2.1.1 Documentation - July 20, 2001
- Python 2.1.1
- Python Language Website
- Welcome to the Microsoft Corporate Web Site
- 11.2.2 Using the cgi module
- Corey's bookmarks

Society, Politics, & News

- The Seattle Times Home Page
- The New York Times on the Web
- Mercury Center Breaking News
- CNN.com
- CNN.com - Bush: All missions being 'executed as planned' - O...
- Slashdot | Finally, A Solution To The DMCA

Entertainment & Media

- WSDOT Puget Sound Area Traffic Cameras
- http://windowsmedia.com/mg/home.asp?WMPFriendly=true
- Joe The Circle: Volume 12, Issue 5
- The Official Site of the Seattle Mariners
- http://redir.windowsmedia.com/pv/wmc-en-us/3/HOME0.htm?WMPFr...
- Intro to Hamsterdance2.com
- http://www.msnbc.com/m/mw/mw.asp?s=&t=V&id=n_mitchell_condit...
- Sign Up and Win

Sports & Recreation

- Seattle Times: Scores & stats

Business & Finance

- Welcome to MSN.com

Uncategorized

- http://search.support.microsoft.com/kb/c.asp?fr=0&SD=GN&LN=E...

Home & Family

- Dilbert Comic Strip Archive - Dilbert Zone - The Official Di...

Figure 3: A links-only montage.

system collects and mines web access logs for each user. From these logs, MONTAGE (a) selects candidate pages to link or embed on the montage; and (b) builds a model of the user's navigation patterns and browsing interests. In the second step, MONTAGE uses this model to calculate the *expected utility* [8] of displaying each candidate page to the user, and then assembles a montage of the highest-scoring ones. We detail these two steps next.

3.1 Step 1: Mine the user model

The primary source of information for the user model is the sequence of pages the user requested. MONTAGE records the time and date of each page visited, its URL, and the topic of the page's content. The topics can be determined using text classification procedures [3, 4, 13]. We applied a content classifier trained to assign topics to web content, developed by Chen and Dumais [5]. The topic classifier employs the linear support vector machine (SVM) method to assign each page a probability of being in each category of a static topic ontology[1]. We took as the topic of a page the most likely category.

The result of evidence collection is a sequence of requests tagged by topic that MONTAGE further refines into *sessions*. For MONTAGE's purposes, a session is a sequence of page requests that begins with a visit to the user's *start page*—the first page the browser displays, or the page visited when the user clicks the "home" button. In practice, MONTAGE does not know which page is the start page and, thus, uses heuristics to identify when one session ends and when the next session begins. Section 4 provides the exact details on how we clean and segment the data into sessions.

MONTAGE uses the page sequences and sessions to compute five aspects about the user for the model:

- **Candidate pages.** MONTAGE selects a subset of the user's previously visited pages as candidates for inclusion in the montage. MONTAGE places no upper limit on the number of pages selected, but does set minimum requirements for inclusion in this set (such as a minimum number of times the user has ever viewed the page).

- **Interest in page.** MONTAGE estimates the user's apparent interest in each page, primarily by how much time the user spent looking at the page, how many links the user followed from the page, *etc.*

- **Interest in topic.** MONTAGE also estimates the user's interest in the higher-level topic of pages viewed. Because, for instance, although the user may show relatively little of interest in several different pages, he or she may be strongly interested in the single topic encompassing them all.

- **Probability of revisit.** MONTAGE estimates the probability that a user will revisit a page in the next browsing session, given the user's current context.

- **Savings possible.** Placing a link or embedding a page on the montage saves the user navigational effort. MONTAGE measures this savings as the effort the user expended navigating to the page originally. All things being equal, MONTAGE will favor including pages on the montage that would be difficult to revisit manually.

[1]An extension to MONTAGE work would be to adaptively adjust the level of detail in the topic ontology, specializing into subtopics any topics that are overpopulated, while leaving large, sparsely-populated topics general. Although we used high-level categories for our initial user study, our content-tagging subsystem tags content more finely, employing a hierarchical ontology of concepts; we intend to explore this area further.

3.2 Step 2: Assemble the montage

Equipped with the user model, MONTAGE is ready to assemble the content montage. Because the montage depends on the user's current browsing context, MONTAGE builds a new page each time the user revisits his or her montage. MONTAGE begins the assembly by calculating the overall expected utility of viewing each content topic or candidate page. We approximate the value of the page p to a user as a function of the interest, $I(p)$, and the navigation savings, $S(p)$. In the general case, the value is some combination of these two factors: $f(I(p), S(p))$. We treat these factors as independent and have explored both a weighted additive model and a weighted multiplicative model. In our experiment, we chose the multiplicative model, and took as the value of a page, $I(p)^{k_1} \times S(p)^{k_2}$. If we assume the cost of displaying uninteresting content to be zero, then the expected utility of a page is the product of the probability the user will visit the page given the current context, $Pr(p|\mathcal{C})$, and the value of the page. Thus, we take as the expected utility of a page:

$$E[U(p)] = Pr(p|\mathcal{C})(I(p)^{k_1} \times S(p)^{k_2})$$

Likewise, we compute the expected utility of a topic $\mathcal{T}$ as the product of the probability that the user will view any page with topic $\mathcal{T}$ in the current context, $Pr(\mathcal{T}|\mathcal{C})$, and the user's interest in the topic:

$$E[U(\mathcal{T})] = Pr(\mathcal{T}|\mathcal{C})I(\mathcal{T})$$

MONTAGE uses these values to place content on the montage to maximize the total expected utility, subject to the sizes of the browser window and each of the embedded pages. Effectively, MONTAGE solves a knapsack problem where the knapsack is the browser window area and each item is a candidate page or topic with associated size and utility. In practice, we could provide users with tools to inspect and tune measures of interest and navigation savings, and allow users to provide feedback on the function for combining these factors. For example, for the multiplicative model, we could assess from users the relative weighting ascribed to interest versus navigation savings. For our studies, we considered these factors to be equal and provided fixed functions for interest and savings. In Section 4 we describe how MONTAGE mines the interest, savings, and probabilities from the web usage logs.

3.3 User control of montage clippings

In the previous section, we saw that the content embedded on the montage was cropped to a web page *clipping*, a smaller window than the original page (see Figure 4). MONTAGE allows the user to have complete control over the size and position on the distal page of this clipping. By specifying the length, width, and focal point of the clipping, users create persistent lenses onto particular portions of the content of pages. Users can also dictate the frequency with which the content refreshes itself during the day (*i.e.*, if the user simply leaves his or her browser at the montage, MONTAGE will automatically refresh the embedded content with the given frequency).

4. IMPLEMENTATION

The implementation of MONTAGE follows the framework we presented in the previous section; in this section we describe the details of the actual implementation. MONTAGE was coded in Python and runs on both Windows and Linux platforms using Internet Information Services (IIS) or Apache web servers.

4.1 Collecting data and mining models

Users of the MONTAGE system direct all of their web browsing through a proxy that logs each request. In our experiments, we used a single proxy running on a central server, although the MONTAGE framework supports running the proxy on individual users' computers. An important advantage of the individual proxy installation is user privacy—if the proxy and the rest of the MONTAGE system all operate on the user's computer, then the user minimizes the risk of unintentionally sharing private information with third parties. We chose the centralized proxy approach for convenience of the experiment.

Before we mine the usage information, we must clean the logged data. First, MONTAGE removes all requests for embedded web content (such as requests for images embedded on pages, or frames in framesets) by parsing the HTML of every page requested and identifying which URLs are embedded. Second, MONTAGE removes repeated requests for pages that automatically refresh (*e.g.*, `cnn.com` automatically refreshes every 30 minutes). MONTAGE computes the statistical mode of the revisit interval for each URL and, if at least 10% of the intervals belong to the mode, MONTAGE removes any requests that are made within a small tolerance of the mode. Thus, MONTAGE effectively removes the second, third, fourth, *etc.* request for a page, but leaves the first request (the actual visit the user made) intact. Finally, MONTAGE segments the usage data into sessions, placing in the same session all requests made by following links from other requests in the session within a fixed time window (10 minutes) of the previous request.

With the proxy logs cleaned and sessionized, MONTAGE proceeds to select the candidate pages and topics. Any page or topic that has been visited more than once is a candidate. For each candidate, MONTAGE builds a naïve Bayes classifier to estimate the probability the user will view the page in a future context. The model classifies a session as to whether the user will view the page or topic in that session. The particular evidential features employed by this model are: the overall rate with which the user views the page; the rate of viewing the page for each 3-hour block of time in the day (*i.e.*, midnight – 3:00 A.M., 3:00 A.M. – 6:00 A.M., *etc.*); and the predominant topic of the pages viewed during the last 4-hour block of time. We evaluated many other features, such as the time since the previous session and the last URL visited, but these other features either offered less lift in predictability or required more training data than we had available.

The final aspects of the user model are the savings possible when embedding a page on the montage and the user's interest in a page or topic. We estimated the savings as the average number of links followed to reach the candidate page from the first page in each session the candidate appears. The user's interest in a page is estimated heuristically as a weighted sum of the average number of links followed from the page, $L(p)$, and the average number of seconds spent in sessions starting with the page, $D(p)$:

$$I(p) = L(p) * 0.50 + D(p) * 0.03$$

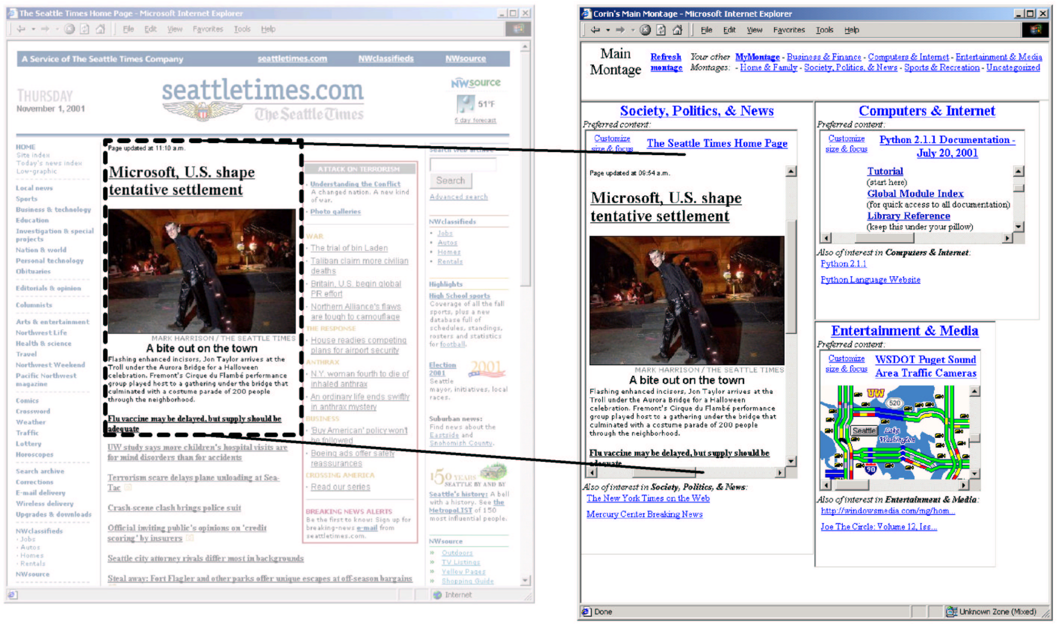

The Seattle Times Home Page - Microsoft Internet Explorer
seattletimes.com
Microsoft, U.S. shape tentative settlement
A bite out on the town
Corin's Main Montage - Microsoft Internet Explorer
Main Montage
Refresh montage
Society, Politics, & News
Computers & Internet
Entertainment & Media
Customize size & focus
The Seattle Times Home Page
Python 2.1.1 Documentation - July 20, 2001
WSDOT Puget Sound Area Traffic Cameras

Figure 4: Cropping content. Distal pages are cropped to a small window when embedded on the montage.

The constants were chosen to equate an average of two links followed from p with an average session time length of 30 seconds. The user's interest in a topic $\mathcal{T}$ is the sum of interest over all pages whose topic is $\mathcal{T}$:

$$I(\mathcal{T}) = \sum_{p \in \mathcal{T}} I(p)$$

4.2 Displaying montages

As the browsing context is potentially different each time the user requests his or her montage, the montage may be rebuilt frequently. Our implementation requires only a few seconds to rebuild a montage with the time dominated by solving the knapsack problem to place embedded content and topic panes in the browser window. We are confident we could improve this time by an order of magnitude with judicious optimization. In our experiments, we rebuilt and cached test subjects' montages only once per hour, both for convenience, and because the set of features we chose for browsing context do not change any faster than about once per hour.

As we described previously, we developed two different visualizations for a montage: the embedded-content montage (Figure 1) and the links-only montage (see Figure 3). The links-only montage is quite simple to display: it is a two-dimensional table and contains only links to web sites. The link anchors are chosen as either the target page's `<title>` or, lacking a title, the URL itself.

The embedded-content montage is a bit more complex. It is formed as a set of nested `<frame>`s: the navigation bar, each topic pane, and the content panes within each topic pane, are all `<frame>`s. The hosting `<frameset>`s specify the size of each pane; it is this mechanism that also sets the size of the cropping window for distal content. To scroll the content to the appropriate position on the distal page, MONTAGE sets the `src` of the frame to be the corresponding URL and additionally adorns the URL with a tag the MONTAGE proxy intercepts (recall, the user directs all browsing through the proxy, including requests made for content embedded on the montage). The MONTAGE proxy passes the request along to the appropriate server (removing the adornment) and inserts a small amount of JavaScript into the resulting HTML stream sent back to the user. The proxy makes no other changes to the returned HTML, but the grafted JavaScript will scroll the page to its appropriate position as it is loaded by the browser. We note that an alternative approach would be to pass the URL directly to MONTAGE and have it fetch the page and modify the content itself—no need for adorning URLs or intercepting requests with the proxy. However, because the URL the browser would see is a MONTAGE page, rather than the actual target site, the browser will not communicate any cookies to the remote server. Thus, to ensure the browser *believes* it really is communicating directly with the remote site, we chose the URL adornment approach.

The user may change the size and position of the cropping window on distal content by clicking the "Customize size & focus" link in the upper-left corner of any content pane. In response, MONTAGE opens a new browser window as shown in Figure 5. The user can control three aspects of how the content is displayed in the montage. First, the user can directly change the size and position of the clipping window simply be changing the size and scroll position of the browser window. Drag the window larger, and the clipping window becomes larger. Second, the user can control how text flows on the page by specifying the width and height (in the text fields in the figure) of the *virtual browser window* the page is rendered in. For example, if the user wants to crop the content very narrowly, he or she could specify the virtual browser be only width 400 (pixels) for that page; the page would then be formatted very narrowly.[2] Finally,

[2]Technically, each embedded clipping is placed within an `<iframe>` on a page sourced by a `<frame>`. The width

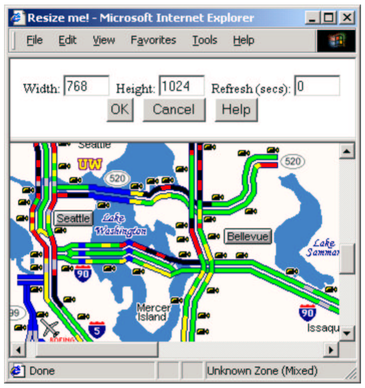


Figure 5: Customizing embedded content. The user can change the cropping window on the distal page by simply resizing and scrolling the browser window.

the user can control how often each clipping reloads itself in the browser window by setting the period, in seconds; zero seconds disables auto-refreshing. By default, MONTAGE sets this value to zero (refreshing disabled), although MONTAGE could suggest a refresh interval based on the user's past revisit frequency to each page.

5. EXPERIMENTAL EVALUATION

In the introduction we presented three hypotheses about users' routine browsing behavior. A key step in answering these questions is actually fielding MONTAGE among users. We conducted a two-week user study of 26 web-savvy users during early October 2001. During the study, users directed all their web browsing through a central proxy running on our "MONTAGE server" (which ran the proxy, a web server, and the MONTAGE implementation). In the first week, we only collected usage information—users browsed the web as they normally would browse. At the end of the first week, we began building models of each user, once per day, to use with MONTAGE. Recall that, although the user's browsing context changes relatively frequently, the predictive model for the user does not—a single additional hour or half-day of browsing rarely changes the model dramatically.

During the second week, we presented users their montages. We instructed users to make their montage their browser's start page, and to revisit the page at least a few times each day. We also added an additional pane to the montage to elicit feedback whenever the user viewed the page (a simple rating reflecting how "pleased" the user was with that montage, ranging from 1 meaning "Not pleased at all" up to 7, "Very pleased"). During this period, 21 of our 26 users actually viewed their montages, on average 25.1 times in the week, and provided feedback 28% of the time. We concluded the study with a questionnaire.

To explore the validity of our hypotheses from section 1, we tested two variables influencing the montage. First, we varied the montage visualization approach: embedded-content or links-only. Second, we varied the model complexity: the expected-utility, context-sensitive model ("complex model") as described in section 4, or a simple model using only the overall frequency of revisits to determine the probability of returning to the page (*i.e.*, ignoring interest in page and savings). We presented every user in the study with two different styles of montage, drawn from the four total configurations; one style in the first half of the second week and the second style in the second half. We particularly sought feedback on the embedded-content / complex model style, so each user viewed that montage, and one other style. Table 1 shows our resulting study groups.

and height fields the user enters simply control the size of the `<iframe>`; the actual browser window size controls the size of the `<frame>`.

Group	Visualization	Model	Number users
1	Links-only	Simple	6
2	Links-only	Complex	6
3	Embedded-content	Simple	6

Table 1: Study groups. Each group also viewed the embedded-content / complex model montage. We count a participant as active if he or she viewed and rated at least one montage during the study.

Visualization	Model	Average score
Links-only	Simple	3.32
Links-only	Complex	4.97
Embedded-content	Simple	1.88
Embedded-content	Complex	3.22

Table 2: Feedback scores. Users rated their montages between 1 ("Not pleased at all") and 7 ("Very pleased").

5.1 General results

Table 2 shows the average feedback score for each of the four montage styles. Recall that a higher score means the user was more pleased with his or her montage. Table 3 shows the scores comparing only one variable at a time, and Table 4 displays each study group's score for their respective styles. Overall, it's clear that the links-only montage using the model that incorporates context is the favorite. We shall next analyze how these findings apply to our hypotheses.

5.2 Hypothesis 1

Users want "one-button access" to their routine web destinations

As a result of our study, users could access their routine destinations in four different ways: follow a bookmark; follow a link on the links toolbar (a small toolbar that displays only four to six bookmarks); follow a link on a links-only montage; and view content or follow a link on the embedded-content montage. Although we did not directly compare the montage approach against bookmarks and the links toolbar,

Visualization	Average score
Links-only	4.40
Embedded-content	2.98
Model	**Average score**
Simple	2.64
Complex	3.79

Table 3: Results by variable.

Visualization	Model	Grp 1	Grp 2	Grp 3
Links-only	Simple	3.32		
Links-only	Complex		4.97	
Embedded-content	Simple			1.88
Embedded-content	Complex	3.08	4.00	2.50

Table 4: Results by study group. Each study group viewed two different montage styles, switching halfway through the second week of the experiment.

users' qualitative feedback indicate that they value the montage higher than the manual approaches. Some users suggested a hybrid, where they can manually add links to the automatically-generated montage. We are considering such an extension to our system.

Between the two visualization approaches, we were surprised to find that the links-only montage was unanimously preferred over the embedded-content montage (see Tables 2 and 3). *A priori*, we had expected users to prefer their target content embedded directly on their montages. Instead, users apparently don't mind following at least one link to view their destination.

We believe that several factors may have influenced users' opinions on this issue. The links-only montage loaded nearly instantly as it is only formatted text. The embedded-content montage, however, would take up to 30 seconds to load completely (downloading all the distal content embedded on the page). It's not clear how much of this delay was caused by our particular experimental setup (*e.g.*, was the proxy a bottleneck?) and how much delay is inevitable (*e.g.*, network delays). We plan to investigate this issue in a future study by prefetching the content embedded on the montage. We also found that the potential full value of the embedded-content montage was hard to attain with the limited screen real-estate associated with typical screen resolutions. The default size of the web clippings and the limited browser area allowed only one or two clippings to fit in a montage unless the user customized the clipping size. We plan to investigate other means for providing greater numbers of content clippings, including the approach of rendering scaled-down views of portions of pages.

5.3 Hypothesis 2

Tools for routine web browsing are enhanced by tailoring links and views to a user's current browsing context

The test for this hypothesis is the comparison between our complex model and the simple one. The complex model conditions the links and content on the montage on the user's current context, while the simple model ignores context. The latter half of Table 3 shows this comparison: the model conditioned on context outscored the simple model by well over a point. Of course, this result only scratches the surface—context is clearly useful, but *what* context should we use? In a future study, we plan to evaluate how much predictive lift each feature offers to determine the most valuable set of features for the user model.

5.4 Hypothesis 3

Past web access patterns can be successfully mined to predict future routine web browsing

The proof of this hypothesis lies in how often users found their target content on or using their montage. Based on a post-study survey, many users agreed that they found the montage helpful in suggesting where they truly wanted to visit. However, there were several cases in which MONTAGE suggested pages that were clearly never going to be revisited. In particular, MONTAGE tends to suggest search engine results pages to users, as MONTAGE estimates the user's interest in these pages highly (MONTAGE estimates interest in part as the number of links followed from the page). We plan to refine MONTAGE's interest estimate in a future implementation and test this hypothesis further.

Quantitatively, our study showed that 73% of users felt that MONTAGE selected appropriate links and content at least occasionally. We feel that this result supports our initial approach to mining routine web browsing. Moreover, we can improve this value in two ways. First, due to a technical issue, MONTAGE does not suggest intranet content; omitting these pages accounted for 18% of the sessions in which MONTAGE failed to suggest the correct target. With the experience of the user study, we are confident that we can overcome this technical detail and expand the universe of content MONTAGE can offer. Second, 45% of the missed sessions led to pages that users had visited sometime earlier in the study. With a longer history of web usage, and perhaps a more sophisticated user model, MONTAGE could suggest useful content for many of these sessions.

5.5 Discussion

Overall, it appears that the MONTAGE user modeling components work well—the context-sensitive model scored higher than the simpler model. Additionally, users are concerned that their start pages load quickly; 64% of our users indicated speed of loading the start page as "very important." This concern is more important, in fact, than having the target of their session display in the start page. We are thus interested in incorporating MONTAGE with a web prefetching system to greatly improve the load time of the embedded-content montage.

Users appreciated MONTAGE's efforts to automatically select and place content on the screen, but users still want some manual authoring mechanism. A number of users point to their links toolbar as what they feel MONTAGE must compete with. Although it is true that MONTAGE was able to identify a number of these preselected shortcuts automatically, it is clear that users would be more inclined to adopt a hybrid system that also allows for direct manipulation and authoring of content to include.

6. RELATED WORK

The MONTAGE work comes in the spirit of related work in the User Modeling community on adaptive personalization of hypermedia. Prior efforts on adaptive hypermedia have relied largely on the use of logical rules, policies, and templates to adapt content, annotations, and user interfaces to different users and user classes. A review of prior and current research on adaptive hypertext can be found in [9].

Among research on personalizing web portals, the most similar in spirit to MONTAGE is *MyOwnWeb* [2]. The proposed *MyOwnWeb* architecture relies on *site descriptions*, which are essentially programs that run on a web site (following links, filling in forms, *etc.*) and produce a block of HTML as output. A system harnessing the *MyOwnWeb* concept would allow users to select the site descriptions desired on a start page, and would execute the site descriptions and concatenate the results for display. MONTAGE improves on this approach in two ways. First, MONTAGE automatically selects the content to display, freeing the user from the need to manually maintain the personalized page. Second, MONTAGE embeds web clippings—an approach that we believe captures a more intuitive and robust approach to viewing distal content than filtering content based on the HTML markup of the page.

Also similar to MONTAGE is work on automatically building bookmark lists. PowerBookmarks [10] automatically builds a Yahoo!-style web directory of the pages each user visits, selecting which pages to include by how often the user visits them and by their link structure. The Bookmark Organizer [11] is a semi-automated system that maintains a hierarchical organization of a user's bookmarks while letting the user keep control of certain aspects (such as "freezing" nodes in the hierarchy to prevent them from being changed). These systems reduce the effort required to maintain the bookmark lists, but they do not address all the drawbacks of such lists. PowerBookmarks and the Bookmark Organizer are insensitive to the user's browsing context, and may require substantial user effort to find the target link (navigating a hierarchical menu structure or drilling down through a web directory). In contrast, MONTAGE leverages a sophisticated user model to automatically select high-quality bookmarks, and to display the most appropriate bookmarks for the user's current context. MONTAGE also displays both embedded content and links in a single page, requiring no scrolling and at most one link to follow.

Building a web montage is related to the more general interest in building web-based document collections. Systems such as Hunter Gatherer [16] assist users in building collections of web content, either of web pages or of within-the-page blocks of content. In contrast to these systems, MONTAGE builds its collection automatically and dynamically, but is effectively "hard-wired" to only one collection (the "material for a start page" collection). The techniques in MONTAGE and collection building systems nicely complement each other, and it would be interesting to apply the user-assisted assembly techniques to MONTAGE. For example, both MONTAGE and the user could add components (links and embedded content) to the start page collection, and MONTAGE would automatically group and display the components as well as it could. The user could provide MONTAGE assistance for components that are incorrectly displayed or that are added manually. The participants in our study agreed that this type of mixed-initiative system would be their likely favorite.

7. CONCLUSIONS AND FUTURE WORK

The goal of MONTAGE is to improve the experience for routine web browsing—the browsing that users tend to repeat over and over in similar situations. We implemented the MONTAGE prototype system by coupling proxy-based monitoring and predictive user modeling machinery with layout and display techniques that compose web montages. We posed several hypotheses about routine web browsing and addressed them in a user study. We found that users appreciated having an automatic system that could suggest links for them to follow, given a broad view of their current browsing context. However, we found that users particularly wanted the system to operate as transparently as possible. We found that loading the montage page should not take more than just a second or two, and users wanted some manual authoring capabilities. These findings encourage us to push forward in a number of directions with MONTAGE, as we have highlighted in this paper and outline below.

In one extension of MONTAGE, we are interested in providing users with more control of the learning and reasoning. For example, we would like to allow users a means of adjusting how navigation costs and information value are combined. We also are exploring the use of dynamic *topic leveling* for structuring and segmenting web pages. In this refinement, we vary the level of detail in a content ontology at which different topics are represented, based on a user's specified preferences or on the amount of content in each topic. Beyond employing data only from a single user, we are interested in the use of collaborative filtering in MONTAGE. We can exploit the browsing habits of multiple users and build different kinds of portals within and across enterprises. Such montages would allow users to inspect portals of information that people with similar interests and patterns of access find useful. For example, there may be value in harnessing collaborative filtering methods to build an intranet portal for assisting new employees of an organization by examining content on human resources that is accessed at different periods of time after hiring, and presenting montages based on the time that a new employee has been at a company. We also seek to persue user interface design challenges highlighted by MONTAGE. For example, we are interested in integrating the automated discovery of favorites with methods that capture user-selected favorites. Finally, we are exploring a better understanding of and addressing potential problems with the instability of layout associated with adaptive procedures. Solutions to challenges with stability may center on designs that mix a superstructure of an overall persistent layout, as specified or locked by users, with adaptation within particular adaptive regions of the layout.

We are excited about the challenges and opportunities captured by the research on MONTAGE. We believe that an elegant intertwining of automated analysis and user control will lead to efficient methods for building and maintaining information resources based on a user's browsing history and preferences.

Acknowledgments

The authors thank Jonathan Grudin, David Heckerman and Mary Czerwinski for feedback on the MONTAGE project, and thank the user study participants for their effort, feedback, and insights. This paper was improved by comments on earlier drafts by Cathy Anderson and Zack Ives and by comments from the anonymous reviewers. This work was conducted during at internship at Microsoft Research and is funded in part by NSF grant #IIS-9872128.

8. REFERENCES

[1] C. R. Anderson, P. Domingos, and D. S. Weld. Personalizing web sites for mobile users. In *Proceedings of the Tenth International World Wide Web Conference*, 2001.

[2] V. Anupam, Y. Breitbart, J. Freire, and B. Kumar. Personalizing the web using site descriptions. In *Proceedings of the 10th International Workshop on Database and Expert Systems Applications (DEXA)*, 1999.

[3] S. Chakrabarti, B. Dom, R. Agrawal, and P. Raghavan. Scalable feature selection, classification and signature generation for organizing large text databases into hierarchical topic taxonomies. *The VLDB Journal*, 7(3):163–178, 1998.

[4] C. Chekuri, M. H. Goldwasser, P. Raghavan, and E. Upfal. Web search using automatic classification. In *Poster Proceedings of the Sixth International World Wide Web Conference*, 1997.

[5] H. Chen and S. T. Dumais. Bringing order to the web: Automatically categorizing search results. In *Proceedings of ACM CHI 2000 Conference on Human Factors in Computing Systems*, 2000.

[6] L. Fan, P. Cao, W. Lin, and Q. Jacobson. Web prefetching between low-bandwidth clients and proxies: Potential and performance. In *Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '99)*, pages 178–187, 1999.

[7] E. Horvitz. Continual computation policies for utility-directed prefetching. In *Proceedings of the Seventh International Conference on Information and Knowledge Management*, 1998.

[8] R. L. Keeney and H. Raiffa. *Decisions with Multiple Objectives: Preferences and Value Trade-Offs*. Wiley, New York, NY, 1976.

[9] A. Kobsa, J. Koenemann, and W. Pohl. Personalized hypermedia presentation techniques for improving online customer relationships. *The Knowledge Engineering Review*, 16(2):111–155, 2001.

[10] W. Li, Q. Vu, D. Agrawal, Y. Hara, and H. Takano. PowerBookmarks: A system for personalizable web information organization, sharing, and management. In *Proceedings of the Eighth International World Wide Web Conference*, 1999.

[11] Y. S. Maarek and I. Z. Ben-Shaul. Automatically organizing bookmarks per contents. In *Proceedings of the Fifth International World Wide Web Conference*, 1996.

[12] Microsoft Corporation. *My MSN*. `http://www.msn.com/`.

[13] D. Mladenic. Turning Yahoo into an automatic web page classifier. In *Proceedings of the 13th European Conference on Artificial Intelligence*, 1998.

[14] V. N. Padmanabhan and J. C. Mogul. Using predictive prefetching to improve World-Wide Web latency. *ACM SIGCOMM Computer Communication Review*, 26(3), July 1996.

[15] M. Perkowitz and O. Etzioni. Towards adaptive web sites: Conceptual framework and case study. *Artificial Intelligence Journal*, 118(1–2), 2000.

[16] m. c. schraefel, D. Modjeska, D. Wigdor, and Y. Zhu. Hunter Gatherer: Interaction support for the creation and management of within-Web-page collections. Technical Report CSRG-437, Department of Computer Science, University of Toronto, October 2001.

[17] Yahoo! Inc. *MyYahoo!* `http://my.yahoo.com/`.