

# Personalized, interactive news on the Web

Krishna Bharat<sup>1,\*</sup>, Tomonari Kamba<sup>1,\*\*</sup>, Michael Albers<sup>2,\*\*\*</sup>

<sup>1</sup> College of Computing, Georgia Institute of Technology, Atlanta, GA 30332, USA

<sup>2</sup> School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

**Abstract.** We present *Krakatoa Chronicle*, an interactive, personalized newspaper on the World Wide Web implemented as a Java applet. The newspaper is similar in appearance to newspapers in the real world, with a multi-column layout and justified text. At the same time, it provides various interaction techniques for browsing the content of articles, giving relevance feedback, and dynamically changing layout. As users interact with the system, individual ‘user profiles’ are built up at the webserver site. These are used to tailor the newspaper’s content and layout to each user’s declared and inferred preferences. The system allows for a balancing of personal and community interests, allowing the user to navigate through a space of newspapers corresponding to a range of viewpoints.

**Key words:** On-line newspapers – Automatic layout – Community interest – Relevance feedback – Customization – Personalization

---

## 1 Introduction

The World Wide Web has become a powerful means of disseminating information to an audience of unprecedented size. Numerous information-gathering and -classifying robots or ‘infobots’ are at work, dedicated to the task of bringing the right information to interested people. Not surprisingly, various newspapers and publishing houses have started to look upon the web as a viable publishing medium, and have started placing some of their material on-line in a hyper-text format. The site, [7], lists several hundreds worldwide. Once the infrastructure for electronic commerce is in place,

---

\* Present address: Digital Equipment Corporation, Systems Research Center, 130 Lytton Avenue, Palo Alto CA 94031, USA; e-mail: bharat@pa.dec.com

\*\* Present address: C&C Research Laboratory, NEC Corporation, 1-1 Miyazaki 4-Chome Miyamae-Ku, Kawasaki Kanagawa 216, Japan; e-mail: kamba@mmp.cl.nec.co.jp

\*\*\* Present address: 2550 Garcia Avenue, MS UMTV21-228, Mountain View, CA 94043-1100, USA; e-mail: mca@eng.sun.com

Correspondence to: K. Bharat

both the supply and demand for organized information can be expected to grow.

On-line newspapers are arguably harder on the eyes and ergonomically less appealing than their hard-copy counterparts. Some studies [15] seem to indicate that raster displays of text compare favorably with printed text in reading tasks. However, on-line publications do have some advantages. They are well suited for dynamic news streams that evolve over time, and for the presentation of multimedia information. Electronic text affords browsing, querying and flexible reuse. Also, since the structure of electronic newspapers can be easily changed, it is possible to dynamically personalize the layout and composition of the newspaper. This can be done interactively by the end-user, and also by the system, based on a prediction of user’s interests. Since electronic newspapers are not subject to the economies of scale of newsprint, it is feasible to give each reader his or her own personal newspaper. Bogart observes in [1] that readers find the idea of a personalized newspaper appealing, but practical considerations have prevented this from being realized under the conventional, hard-copy publishing framework.

There are many advantages to placing a newspaper on the web: universal access, the presence of infobots which direct users to the service, and a commonly available presentation infrastructure. Unfortunately HTML is not rich enough to support custom user interfaces such as the one needed in an interactive newspaper. Although ‘frames’ and ‘tables’ can be used to partition the page into articles, HTML still lacks the support to display multi-column, justified text within individual articles (e.g., the Chicago Tribune [4] has a tabular layout without multi-column text). Hence, web-based newspapers have not bothered to reproduce the look of real-world newspapers. We felt this was an important part of the experience of reading news. The visual appearance of a document has a big influence on the manner in which it is perceived and the degree to which it is accepted. The format of present day newspapers has evolved over a long period of time with layout conventions that have a high degree of social acceptance.

While we wished to emulate real-world newspapers in matters of appearance, we wished to provide more in terms

of interaction. Currently web-clients send all their input to the server, incurring a large time penalty due to synchronous network communication. This large response time stifles interaction, limiting the set of interfaces that can be supported in this fashion. A richer drawing model and the ability to execute computations at the client are vital to providing quick semantic feedback and for implementing the custom widgets needed in a newspaper interface.

We present the *Krakatoa Chronicle* [8], an interactive, personalized newspaper on the WWW, which overcomes these limitations. The newspaper may be viewed using Java-compliant web-browsers, i.e., browsers which can display HTML documents containing executable Java byte-code. In this model of computation, a document such as our newspaper, will bring with it code to be executed within the user's browser. In our case the code renders a realistic newspaper on the user's screen and does input handling, behaving like a regular user interface application. Unlike in previous web-based newspapers, this allows for interactive personalization, browsing and layout control. It provides an open architecture for experimenting with various interaction, display and personalization strategies, and for embedding other interactive features within the framework of a newspaper.

## 2 Overview

In this section, we present an overview of our newspaper. This corresponds to the user's view of the system.

### 2.1 Connecting to the service

The Krakatoa Chronicle has a client-server architecture. Articles are collected and indexed at the server site, and organized into newspapers for a community of registered users. This happens off-line on a periodic basis. The server site is usually the same as the host running the web-server. Users connect to the system via the web and access their personal newspaper. This causes a web-document containing an embedded Java applet to be downloaded to the client-browser. The applet then displays the newspaper and handles input from the user.

### 2.2 The newspaper

Figure 1 shows a typical screen of the Chronicle, viewed in Sun's *HotJava* web-browser. The page displayed is the first of 13 pages. It is partitioned into a mosaic of articles. Each article is an interactive widget, containing a title in bold font, a scrollable text region containing the article's text in justified Times Roman font, and other widgets to support browsing and feedback.

In hard-copy newspapers, text which cannot be displayed in the main article is placed in a continuation on another page. In an electronic newspaper it makes more sense to allow the user to browse the article in place, without losing context by having to move to another part of the newspaper. For this purpose, single column articles have vertical scrollbars and multi-column articles have horizontal scrollbars. Scrolling causes the article's text alone to scroll without

displacing the title and other widgets. In addition, the *Peek Button* (labelled 'P'), allows articles to be temporarily maximized over the entire page, while the mouse button is held down. If necessary, the maximized article can be scrolled by a horizontal drag. Clicking on the *Save Button* (labelled 'S'), causes the article to be saved to the user's *Scrapbook*, which is a web-page containing links to various articles the user wished to preserve.

The *Score Bar*, shown immediately below the title, is a custom widget that displays the importance of the article to the user and to the community. This is in fact a slider.

It was our intention to make widgets as unobtrusive as possible, receding into the background when not in use, to avoid distracting the user from the task of reading the newspaper. Figure 2 illustrates this behavior in the case of a *Score Bar*. The importance of the article to the user (or *personal importance*) is shown quantitatively by the location of the *Score Bar*'s thumb, and qualitatively as a textual rating (at a coarser granularity). Ratings range from 'Totally Irrelevant' to 'Very Interesting' through 'No Comment.' The vertical line shows the importance of the article to the community (or *community importance*) on the same scale. This cannot be manipulated by the user. Later, we discuss how these scores are computed.

Initially, personal importance is the system's estimate of the user's interest in an article, based on feedback given during the browsing of previous editions. If the rating seems incorrect, the user can change the article's score on the *Score Bar*. This feedback has two effects: (a) it could alter the rank of the article in the current newspaper, and (b) it causes the server to be notified of the updated score. Users have individual 'interest profiles' which track their interest in various topics. As we shall explore in more detail, once users have provided feedback on the articles they have read, their profiles are automatically updated.

The newspaper's content and layout are based on the importance of each of the articles to the user and to the community. The manner in which these values are combined in deciding the article's importance can be controlled dynamically using the *Tendency Bar*. Figure 1 shows this slider in its passive state. The *tendency* factor ranges from 'Very Personal' (where only personal importance is considered) to 'Fully General' (where only community importance is considered) through 'Average' (where the scores are averaged). Articles of high importance will appear earlier in the newspaper, have titles with larger font, and be allocated more space than articles of lesser importance. Also, within a given page, the more important articles will be closer to the top-left corner than other articles. These layout considerations are similar to those employed in the typesetting of hard-copy newspapers.

Two other controls are provided to manipulate the layout. The *Density Bar* controls the density of articles within a page, and hence the number of pages in the newspaper. The amount of screen real-estate allocated to each article is a function of both the article's importance and the amount of text in the article. The *Sensitivity Bar* controls the degree to which these parameters affect space allocation. In the 'Totally Insensitive' state the space in the page is divided evenly among the articles it contains. As sensitivity is



Fig. 1. View of an edition of the Krakatoa Chronicle (6 articles, *sensitivity*: high, *tendency*: very personal). The animated logo at the top represents *Krakatoa*, an active volcano in the Java sea

increased, articles of greater importance or size will tend to expand at the expense of smaller, less important articles.

The user can customize the layout by changing article scores and manipulating global layout parameters such as packing density, variance in size allocation, and the ratio in which the personal and community importance factors are combined to decide the ranking of articles. Since layout often causes global reorganization, it is both time consuming and potentially confusing to the user. Hence, users are given a *ReLayout Button* to explicitly initiate relayout, and are expected to batch changes to layout parameters and article scores between invocations of the command. Our initial implementation performed automatic relayout after each change, which proved to be distracting.

Figures 1, 3 and 4 illustrate the effect of varying tendency on a given edition. The articles in Fig. 1 (Very Personal) have high personal ratings, while articles in Fig. 3 (Fully General) tend to have low personal ratings, but are displayed because they are relevant to the community (i.e., their community ratings are high). Figure 4 illustrates the average case and it can be seen that there is a mix of articles drawn from the two previous cases. In Figs. 1 and 3 sensitivity is high, showing a fair amount of variance in the space allocated to

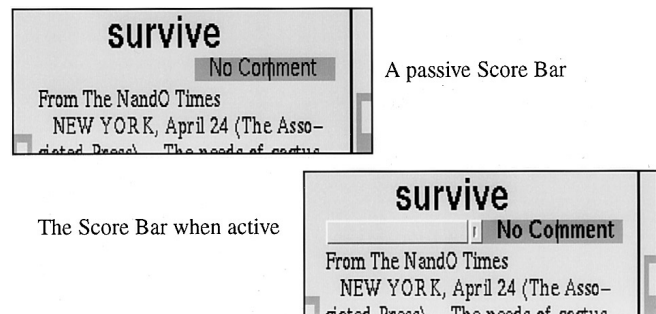


Fig. 2. When not in use, widgets recede into the background to avoid distracting the user. They 'wake up' when the mouse is within 'sense range'

articles. In Fig. 4, since sensitivity is low, article sizes are approximately the same.

The button labelled 'B' is used to switch *bold mode* on and off. In bold mode (shown in Fig. 5) the article currently under the cursor is rendered in bold font causing the text to expand at the expense of the margins of neighboring articles. This is a preliminary attempt to provide focus within the newspaper page. A better option would be to use a fish-eye (focus+context) technique such as the 'Document Lens' [16]. We have an additional motive for wanting users to

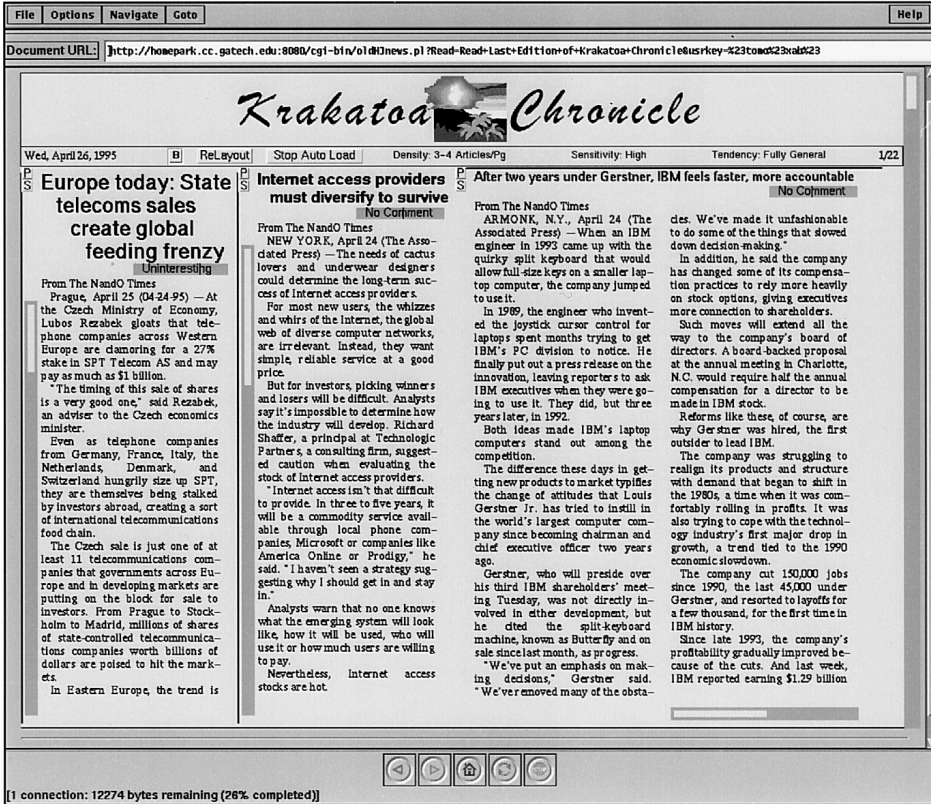


Fig. 3. Another view of the same edition (three articles, sensitivity: high, tendency: fully general)

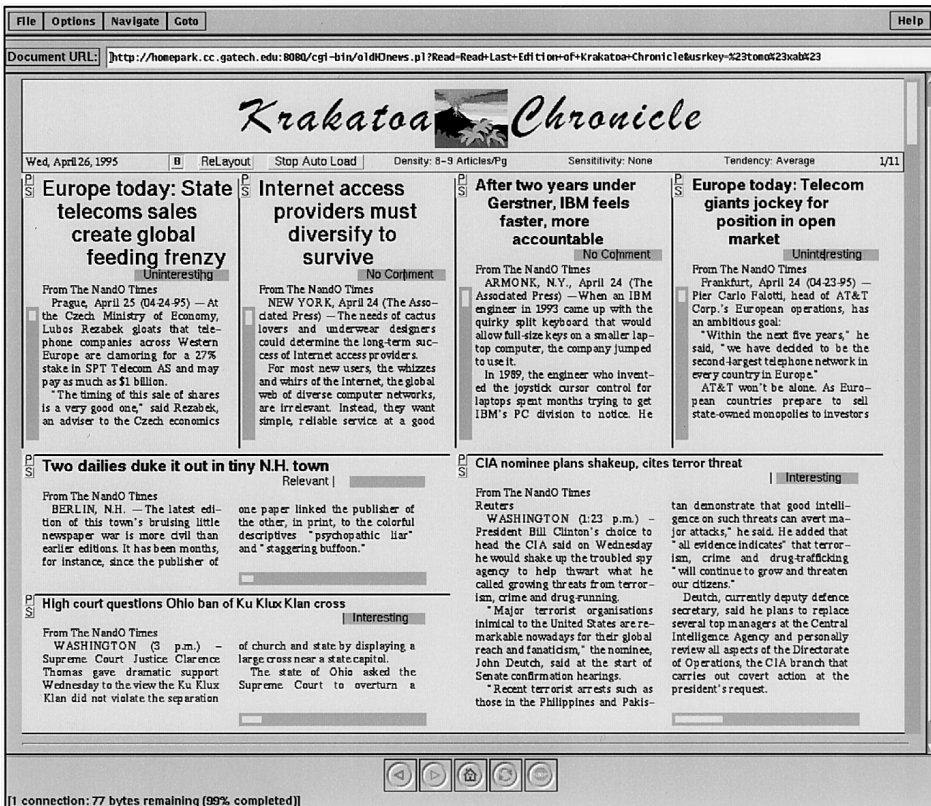


Fig. 4. Third view of the same edition (eight articles, sensitivity: none, tendency: average)

p://homepark.cc.gatech.edu:8080/cgi-bin/oldHJnews.pl?Read=Read+Last+Edition+of+Krakatoa+Chro...

**B** ReLayout Density: 5-6 Articles/Pg Sensitivity: Average

**ology, Aztech of sound card**  
No Comment  
Times  
il 25 (04-24-95) —  
ogy Ltd. and Az-  
d. both expanded  
e market for com-  
s last year after a  
filed for ban-  
from industry  
quest show.  
apore-based com-  
e the worldwide  
cards, which give

**Indonesian police seize 'communist' video games**  
Interesting  
From The NandO Times  
JAKARTA, April 26 (04-25-95) —  
Indonesia, still fearful of communist  
influence three decades after it was  
outlawed, has seized Nintendo  
video-game cassettes it says contain  
the hammer and sickle symbol, the  
Jakarta Post said on Wednesday.  
It quoted police in the central  
Java city of Semarang as saying the  
US\$25 cassettes had been seized at  
several outlets.  
Police were quoted as saying the  
communist symbol was hard to find

**People of Decker, Mich., talk ab**  
From The NandO Times  
WASHINGTON (7:41 p.m.) —  
Residents of Decker, Mich., the  
hometown of one of the men under  
investigation in connection with the  
Oklahoma City bombing, offered  
contrasting views on the govern-  
ment in a televised town meeting  
late Tuesday.  
The meeting, organised and  
broadcast by ABC's "Nightline"  
programme, underscored the strong

**s. declares  
Oklahoma City**  
No Comment  
Times  
N (3:08 p.m.) —  
nton declared Ok-  
disaster area on  
ring the way for  
relief, and joined  
a former Secret  
rd killed in the  
k ago.  
prepared to host a

**Pacific Telesis may buy \$150 mln  
in Scientific-Atlanta gear**  
No Comment  
From The NandO Times  
San Francisco, April 24 (04-24-95)  
—Pacific Telesis Group said it may  
buy up to \$150 million of video  
equipment from Scientific-Atlanta  
Inc. during the next three years.  
The companies want to deliver  
video services to four large areas of  
California in the beginning of 1996.

**Bombing echoes extremists' tac**  
From The NandO Times  
Whoever blew up the federal  
building in Oklahoma City used  
tactics that are strikingly similar to  
those urged by far-right advocates  
of "leaderless resistance" against  
the government, according to civil  
liberties experts who keep track of  
militant groups.  
The idea was developed by two  
former leaders of the Ku Klux Klan  
and has been promoted by some of  
the self-styled citizen militias that

Fig. 5. 'Bold' mode to provide focus

explicitly indicate their point of focus within the newspaper. We track interaction events to allow us to correlate them with the user's explicit relevance feedback. The hope is that, if a good correlation is detected, we may be able to estimate the user's interest in articles largely by observation, reducing the need for explicit positive feedback. They may still need to provide negative feedback where applicable. Preliminary studies [17] show this to be a worthwhile approach. A focus-oriented interaction technique would be an excellent source of 'interest events.'

### 2.3 Maintaining user profiles

In addition to the part of the user profile which is implicitly maintained by the system based on user feedback, there is an explicit part which the user creates and maintains. This consists of two lists of keywords – a list of keywords the user is interested in and another containing keywords they wish to avoid. When users connect to the system they are given three options: 'Create New Newspaper', 'Read Last Newspaper' (if applicable) and 'Edit User Profile.' The last option takes them to a form where the explicit keywords lists can be edited. When the user selects 'Create New Newspaper' (and also every time the document set changes), the feedback log since the last session is used to update the implicit portion of the profile. In creating a newspaper, the keywords explicitly specified by the user are given precedence over any other keywords the system may have added to the profile on its own accord.

## 3 System architecture

In this section, we describe salient features of the Krakatoa Chronicle's architecture. Figure 6 illustrates the various activities involved and the entities responsible for each.

### 3.1 Server site processing

#### 3.1.1 Gathering articles

We download articles on a periodic basis from the *News and Observer* newspaper (NandO [14] based in Raleigh, NC) with their permission. Most of the information available online (such as Usenet news) is not suitable for inclusion in a newspaper, because it is either lacking in newsworthy content or lacks the objective, professional style of newspaper articles. Our focus was not on finding information on the Internet to compose a newspaper. Rather, we wished to develop a mechanism that would allow publishing houses and commercial newspapers to present their articles to a discerning audience. Hence, one of the assumptions we make is that there is a readily available collection of quality articles, possibly classified into sections. In practice, such a newspaper could combine articles from various news sources.

#### 3.1.2 Indexing and formatting

Perl scripts translate the source documents to plain text (in our case from HTML). The text of the articles is indexed by the *SMART* document indexing engine [2] to produce a set of document feature vectors. In the process, the engine

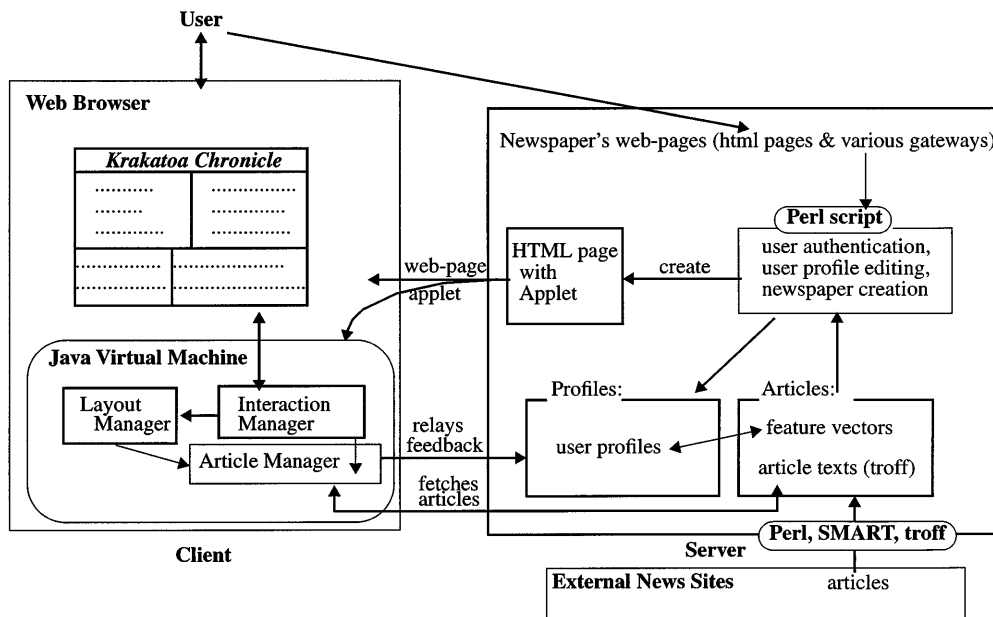


Fig. 6. System architecture of the Krakatoa Chronicle

eliminates common words (such as ‘a’ and ‘an’), eliminates suffixes by stemming and generates a set of document feature vectors, one for each document. Each *document vector* contains a list of keywords which were found to be highly representative of the document within the document set, and corresponding weights. The weight of a keyword is proportional to its merit as a representative of the document. Keywords that occur frequently in a document and occur in few documents within the document set are considered good discriminants. Specifically, weights are computed using the ‘term frequency times inverse document frequency’ (or TFIDF) metric. Salton et al. ([18] p. 105) note that TFIDF is quite effective in keyword-based information filtering.

Next, a Perl script separates the title, credits and content portion of each of the articles, and formats the content into justified Times Roman using the UNIX utility, *troff*. We needed to tailor our scripts to the source format. This is the only component of the system that is source specific. Preformatting the text saves the client code the burden of formatting text in the client at run time.

### 3.1.3 Interaction with users

User interaction at an administrative level is handled by CGI scripts. These are invoked when a user interacts with one of the newspaper’s web-pages to create their account, launch their newspaper or update their explicit profile. A script is involved also when feedback from a user updates their profile. Each user’s *user profile* is a vector of keywords and weights similar to document vectors. Indeed, it is derived from document vectors by a process of weighted averaging. Explicitly specified keywords are also a part of the user profile and receive the maximum positive or negative weight as the case may be. When users provide feedback on the relevance of articles, the scores they supply are transmitted to the server and logged. At a subsequent point in time, the final scores assigned to each article are used to update the profile.

If  $dS$  is the change in the relevance score due to user feedback for an article with normalized document vector,  $\vec{d}$ , the implicit part of the user profile,  $\vec{u}_i$ , is updated as follows:

$$\vec{u}_i = \vec{u}_i + \vec{d} \cdot dS \cdot mutability,$$

where *mutability* is a scalar in the range [0..1] which decides the degree to which the incoming feedback will affect the profile. We used a *mutability* value of 0.75.

### 3.1.4 Creating newspapers

When articles are first indexed, a newspaper is created for each of the users. Subsequently, users may ask for fresh newspapers to be created at any time when there is outstanding feedback. The process of creating a newspaper involves computing the goodness of the match between the user profile and each of the documents in the document set. This is computed in each case as the scalar product of the user profile and the document vector, normalized by document size. This quantity is scaled to the range [-1000...1000] and represents the document’s personal importance for the user.

In the absence of a good measure of community importance, we chose to average the personal importance of each of the users in the community. Unfortunately, this produced uniformly low scores for all articles, since, given any article, a large fraction of the users will not have a strong opinion about it. Later, we decided to average the scores of users who have ‘significant’ opinions about the article (in the system’s estimation), determined using a pair of significance thresholds. This was more effective. This is still not as effective as having a human editor rate the importance of the articles to the community. Often, there are developments that do not match anyone’s list of favorite topics, but are still important enough to be on the front page of the community newspaper. The ranking used by the source of the articles (e.g., the location of the article in NandO’s newspaper) could also be used for this purpose.

The newspaper is transmitted as an HTML document containing a Java newspaper applet, with the titles, scores and the number of lines in the various articles as parameters. The contents of the articles are not transmitted until necessary. The initial parameters enable the applet to compute an initial layout before beginning to download the formatted contents of articles.

### 3.2 Client side processing

The applet consists of three logical parts: the Interaction Manager, the Article Manager and the Layout Manager. The *Interaction Manager* builds and manages the user interface of the newspaper. In the process, it uses the *Layout Manager* to: (a) compute the allocation of articles to pages, and (b) compute the layout of each page in the newspaper. It then builds the user interface by embedding article widgets within page widgets. Since the layout is subject to constant change, the *Layout Manager* is invoked frequently to reorganize the user interface.

The *Article Manager* is given the task of asynchronously downloading the formatted text of the articles from the server. This is done asynchronously to avoid freezing the user's screen until all the articles are loaded, since it may take a while. For example, the size of formatted text for 92 articles (a sample document set) is about 2 MB. The order in which articles are fetched from the server is driven by the current layout criteria. As the tendency of article selection is changed or article scores change due to feedback, the ordering of articles within the document will change. Every time this happens the *Article Manager* thread is halted, while the new order is computed. This involves sorting the article set based on the weights of articles, where an article's weight is computed thus:

$$weight = personal\_score \cdot \beta + community\_score \cdot (1 - \beta),$$

where  $\beta$  ( $0 \leq \beta \leq 1$ ) is the tendency factor,

Then, the *Article Manager* thread starts once more from the beginning, but only downloads articles that have yet to be loaded.

#### 3.2.1 Computing layout

Layout computation is completely independent of article downloading and is based solely on the initial information received about each article (namely title, number of lines, and scores). The layout process involves the following two steps: (i) assignment of articles to pages, and (ii) structuring of articles within a given page.

First, based on the user-controlled density factor, the average number of articles per page is computed. We use the number of lines in each article to partition the sorted list of articles into segments corresponding to each page. Articles lengths (subject to a threshold) are used to decide how many articles should go on a given page. Pages containing small articles may have more than the average number, and pages with large articles may have less.

Then, to lay out articles within a given page, we employ a randomized algorithm to recursively partition the page real-estate using vertical and horizontal splits. With each split,

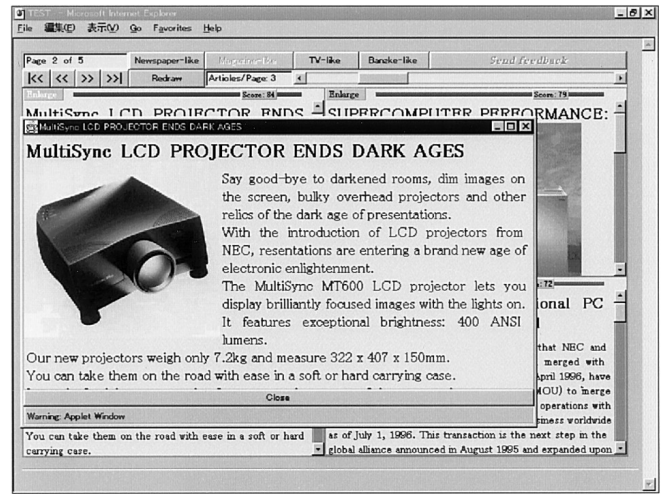


Fig. 7. Anatomomy

the sorted list of articles under the split is partitioned into two halves of approximately equal cumulative weight. The decision to split vertically or horizontally is based on a random number weighted by (i) the availability of columns, (ii) the lengths of the title lines, (iii) the dimensions of the space available. The ratio in which the split partitions the space is based on (i) the sensitivity factor (user controlled), (ii) the cumulative weights of the articles on either side of the split, (iii) the number of lines on either side of the split.

This creates a partition of the page into rectangular regions. Each region embeds an article widget. Although hard-copy newspapers sometimes use non-rectangular embedding, we found this to be quite adequate. The use of *in situ* browsing techniques simplifies our task by reducing the need for more elaborate embedding schemes. For instance, newspaper typesetters are careful not to make continuations too small because the reader would be forced to turn to a new page for very little compensation. This does not affect us, since we use scrollbars. Indeed, the need to scroll very little can be a blessing.

## 4 Discussion and related work

### 4.1 Performance

A typical batch of articles would contain about 75–100 articles. Extracting feature vectors took about 8 min on a *HyperSparc* and was done off-line. The web-server ran on a *SparcStation 20* and took about 5 s to compute the newspaper for each user. This tends to be a function of the size of the user profile. We limited the growth of user profiles by keeping only the top 5000 significant keywords. One could also employ a decay scheme to progressively weaken weights unless reinforced by feedback. This would help cope with the changing interests of users.

The time taken to download an article from the web-server was about 8 s over our Ethernet LAN. The first page took about a minute to come up (for a six-article page). Thereafter, the user was able to interact with the page while articles were downloaded in the background. Random access to a page later in the newspaper or a change in tendency

could cause the newspaper to block while the relevant articles are fetched.

#### 4.2 Personalization

*Fishwrap* [3] maintains a user profile which keeps track of the user's personal interests and community affiliations. They also generate a front page consisting of frequently accessed articles. However this does not necessarily mean the users liked those articles; nor is there a means of mixing community and personal articles in varying proportions. Ours is the first newspaper to attempt a realistic, multi-column rendering with personalization of layout based on predictions of user interest. We believe that a totally personalized newspaper is inadequate, since it fails to expose the user to important developments and does not encourage the growth of new interests. Our experience shows that averaging the opinions of other users to compute community importance is not as good as having an editor's viewpoint, but is a reasonable substitute. In the future, we hope to help users benefit from the ratings of others with similar tastes using *collaborative filtering* strategies, as in [6, 19]. This helps effectively bootstrap a user's profile, and allows users to benefit from the feedback given by others who read the newspaper earlier (e.g., those in an earlier time zone).

#### 4.3 Computing user profiles

Many schemes have been employed for this purpose. The simplest is to let users maintain the profiles themselves [20]. Besides being tedious, this method fails to track the subconscious and transient interests of the user. Users sometimes cannot find appropriate words to express their interests. In *Newsweeder* [10], the user assigns a score to each article via a score bar, as in our case. This is effective, but still a bit burdensome. Morita et al. [11] tried predicting the interest of users based on the time spent in reading Usenet news articles and got promising results. They required users not to be engaged in other activities while reading news. This may not be reasonable requirement.

In an interactive system such as ours, there is plenty of opportunity to record events as the user interacts with the newspaper. Scrolling, peeking at, and saving an article to scrapbook are reliable indications of interest. One alternative would be to have the user's Score Bar incremented automatically when the number of interest events within an article crosses a certain threshold. This will allow the newspaper to periodically revise its prediction of the user's interest. Such changes to the predicted score would also transmit feedback to the server, in the same way as explicit feedback. This approach has been studied in a reimplementing of Krakatoa Chronicle called *Anatagonomy*, discussed in the next section.

### 5 Recent work

Things move quickly on the Internet. Since this paper was written, our newspaper has been reimplemented by the authors at their respective companies, namely DEC and NEC.

The chief motivation for reimplementing was to move from Java(alpha) to JDK 1.0, to allow the applet to run on all Java-compliant browsers. The newer implementations are more efficient. They use the same architecture as Krakatoa Chronicle and inherit most of its features.

The NEC implementation, known as *Anatagonomy* [9] (Japanese for "As you like it"), is shown below (Fig. 7). *Anatagonomy* uses a better matching and learning engine than the Chronicle, based on the work of Nakamura et al. (described in [12]). Global layout controls for density, sensitivity, etc. have been removed; instead, the user is given the option of a set of predefined layouts. In addition to displaying articles in a newspaper format, they can also be presented with a magazine-like appearance, as a passive scrolling display, and as an index view (called "banzke"). One of the goals of implementing *Anatagonomy* was to explore the possibility of using implicit feedback (based on actions on the user interface) to compute a relevance score for articles. In a user study [17], it was discovered that there is an appreciable correlation between scrolling and enlarging articles and the user's perception of the article's relevance. Not performing these actions was taken as a sign of lack of interest and interpreted as negative feedback.

*X-Press-O* [13], a commercial realization of this system by NEC and *The Yomiuri Shimbun* newspaper, is publicly accessible.

*EachNews* is a port of Krakatoa Chronicle to JDK 1.0 within DEC, and preserves most of its features. The relevance feedback mechanism was improved to make the user profile update mechanism a closer approximation to the true inverse of the matching function. In the equation presented earlier for updating the user profile,  $\bar{u}_i = \bar{u}_i + \bar{d} \cdot dS$ , errors due to the cumulative effect of terms occurring in multiple documents were ignored. Such terms are relatively infrequent in the set of significant keywords (typically less than 10%), because the TFIDF metric tends to reduce their weights. However, when they do occur in the document matrix, their scores tend to be updated in the profile for each of the documents they occur in. *EachNews* compensates for this by computing user profile scores for multiply occurring terms first, and then adjusting the scores of the singly occurring terms, based on the remaining portion of the relevance feedback.

*EachNews* also provides the notion of sections. The newspaper's administrator provides a set of sections, and articles drawn from the various sources are assigned to the sections based on a set of mapping rules. The user is provided with a selection dialog for choosing sections to view (see Fig. 8). Only articles in the selected sections get displayed in the resulting view. This allows the user to browse a slimmer newspaper, leading to a lightweight and more responsive applet.

### 6 Conclusions

We have described our experiences with Krakatoa Chronicle, the first newspaper on the World Wide Web to provide a layout similar to that of real-world newspapers. Unlike other newspapers on the web, it is interactive and supports end user customization article layout. Also, the user is allowed



The screenshot shows a web browser window displaying a news page titled "News Du Jour". The page layout includes a header with the date "Tue 18 Mar, 97", navigation links like "FP", "B", and "ReLayout", and a status bar showing "3-4 Articles/Pg" and "Sensitivity: None". The main content area features several articles:

- Cable modems come into vogue for new and experienced web surfers**: A lead article about cable modems, with a sub-headline "NEW ORLEANS (Mar 17, 1997 5:31 p.m. EST) -- John Swiokla jokes that his Newton, Mass., neighbors have 'cable envy.' But it's not because of the shows he can get. Swiokla is using his cable TV line to surf the Internet, shop, download work files, exchange e-mail and help his five children with homework. He is one of several thousand people who are using powerful devices called cable modems to link their personal computers to cable TV lines. Users can watch TV while they tour cyberspace over the same cable line." A "Show in New move to 100 time than star" link is visible.
- Pakistan tensions will remain in South Asia if Kashmir issue lingers on**: A sub-headline "Very Relevant" above the article text. The text starts with "ISLAMABAD (Mar 18, 1997 10:49 a.m. EST) - Pakistani Prime Minister Nawaz Sharif Tuesday vowed his government would not compromise national interests during upcoming talks with old foe India. Sharif told reporters all outstanding issues between the two countries would be discussed in the March 28 to..."
- Digital launching chips for PCs in challenge to Intel**: A sub-headline "Relevant" above the article text. The text starts with "NEW YORK (Mar 17, 1997 11:02 a.m. EST) -- Digital Equipment Corp. Monday is introducing a new microprocessor chip that can double the performance of personal computers, posing a challenge to widely used Pentium chips from Intel Corp. Digital said that PCs running its new-version Alpha chips will start at under \$2,600 -- comparable to Pentium-based machines. Digital is aiming its new microprocessor at corporate desktop PCs that run on Microsoft's Windows NT operating system. Hewlett-Packard is one of the..."
- Microsoft, Hewlett-Packard computers cheaper to**: A sub-headline "Very Relevant" above the article text. The text starts with "(Mar 18, 1997 00:43 a.m. EST) -- Microsoft Corp. is expected to announce Wednesday a major alliance with Hewlett-Packard aimed at making workplace computers cheaper to operate and maintain. The two industry giants will integrate their products more tightly so that, for example, an administrator can fix other computers on the network from a central machine without touching the others, according to reports in computer industry trade publications. Hewlett-Packard is one of the..."

On the left side, there is a "SECTIONS" menu with checkboxes for: Business (20), Information Technology (27) (checked), National News (20), Sports (20), Opinions (8), and World News (23) (checked). Below the menu is an "Update" button and a "Unsigned Java Applet" warning.

Fig. 8. EachNews, with a mix of articles on information technology and world news

to control the extent to which public and personal interests are mixed in composing the newspaper. We are grateful to the News and Observer for allowing us to use their on-line articles for our newspaper.

In the future, we expect to include dynamic components into the newspaper framework, such as a shared whiteboard, crossword puzzles, animated comic strips, etc. These are starting to appear in on-line publications. For example, the *Editor and Publisher News* site [5] includes on-line discussions on current events. Embedded interactive features are easily implemented in the Java framework and seem natural in a newspaper setting and could use a similar scoring/personalization mechanism for layout. This would allow an electronic newspaper such as ours to play a more useful role on the desktop.

## References

- Bogart L (1989) Press and Public: Who Reads What, When, Where, And Why In American Newspapers. Lawrence Erlbaum Associates, Hillsdale, New Jersey
- Buckley C (1985) Implementation of the SMART Information Retrieval System. Cornell University, CS Department Technical Report: TR85-686
- Chesnais PR, Mucklo MJ, Sheena JA (1995) The Fishwrap Personalized News System with Mathew Mucklo and Jonathan Sheena. Proceedings of the 1995 2nd International Workshop on Community Networking. Princeton, NJ, June 1995, pp 275-282
- Chicago Tribune (1995) <<http://www.tribune.com/>>
- Editor and Publisher News (1995) <<http://www.mediainfo.com/ephome/news/newshtm/news.htm>>
- Goldberg D, Nichols D, Oki B, Terry D (1992) Using Collaborative Filtering to Weave an Information Tapestry. CACM 35(12):61-70
- Intercom News Index (1995) <<http://www.intercom.com.au/intercom/newspr/index.htm>>
- Kamba T, Bharat K, Albers M (1995) The Krakatoa Chronicle: An Interactive, Personalized Newspaper on the Web. In: Fourth International WWW Conference, Boston, Mass.
- Kamba T, Sakagami H, Koseki Y (1997) ANATAGONOMY: A Personalized Newspaper on the WWW. Int J Human-Comput Stud (Special Issue on Innovative Applications on the World Wide Web) 46 (6): 789-803
- Lang K (1995) NewsWeeder: Learning to Filter Netnews. In: Proc. of the 12th International Conference on Machine Learning, pp 331-339. San Francisco, CA, Morgan Kaufman
- Morita M, Shinoda Y (1994) Information Filtering Based on User Behavior Analysis and Best Match Text Retrieval. In: Proc. of SIGIR'94, Dublin, Ireland, pp 272-281
- Nakamura A, Mamizuka H, Toba H, Abe N (1995) Learning Personal Preference Functions using Boolean-Variable, Real-Valued, Multivariate Polynomials. In: Proc. of the 52nd National Convention of the Information Processing Society of Japan (in Japanese),
- NEC and The Yomiuri Shimbun (1996) X-Press-O. Tokyo, Japan. <[http://pnews.cplaza.or.jp/start\\_pnews\\_e.html](http://pnews.cplaza.or.jp/start_pnews_e.html)>
- News and Observer (1996) The Nando Times <<http://www.nando.net/newsroom/nt/nando.html>>
- Osborne DJ, Holton D (1988) Reading from Screen versus Paper: there is no difference. Int J Man-Mach Stud 28: 1-9
- Robertson GG, Mackinlay JG (1993) The Document Lens. In: Proc. of UIST '93, pp 101-108, Atlanta, Georgia
- Sakagami H, Kamba T (1997) Learning Personal Preferences On On-line Newspaper Articles From User Behaviors. In: Proc. of the Sixth International WWW Conference, Santa Clara, Calif,
- Salton G, McGill MJ (1983) Introduction to Modern Information Retrieval. McGraw-Hill, New York
- Shardanand U, Maes P (1995) Social Information Filtering: Algorithms for Automating "Word of Mouth". In: Proc. of SIGCHI'95, Denver, Colorado
- Yan TW, Garcia-Molina H (1995) SIFT - A Tool for Wide-Area Information Dissemination. In: USENIX Technical Conference, pp 177-186, New Orleans, Louisiana



KRISHNA BHARAT received his B.Tech from the Indian Institute of Technology, Madras in 1991 and his Phd in Computer Science from Georgia Institute of Technology in 1996. At present he is a software engineer at DEC Systems Research Center in Palo Alto. His research interests include user interface toolkits, distributing computing, online information services, and data mining on the web.



MICHAEL C. ALBERS is a User Interface Designer at JavaSoft. His interests include human-computer interaction, the use of sound in the interface, and cognitive science. Michael can be reached at [michael.albers@sun.com](mailto:michael.albers@sun.com).



TOMONARI KAMBA received his B.E. and M.E. in Electronics from the University of Tokyo in 1984 and 1986, respectively, and joined NEC Corporation. He was a visiting scientist at the Graphics, Visualization & Usability Center at the College of Computing, Georgia Institute of Technology from 1994 to 1995. Currently, he is an assistant research manager of C&C Research Laboratories, NEC Corporation. His research interests include multimedia user interfaces, mobile computing, and online information services.