Cyberbrowsing: Information Customization on the Web

Hal Berghel, Daniel Berleant, Thomas Foy, and Marcus McGuire
Department of Computer Science and Computer Engineering, University of Arkansas, Fayetteville, AR 72701. E-mail: hlb@acm.org; djb@engr.vark.edu
Metro Information Services, E-mail: thomas@opmaxx.com
Raytheon E-Systems, E-mail: MMCGUIRE@cyberramp.net

The ability to discriminate and distinguish among individual documents in the ever-increasing volume of information available through digital networks is becoming more and more difficult. With websites being added to the 100 million installed base by tens of thousands per month, information overload is inevitable (H. Berghel, 1997). There are two basic paradigms for dealing with this information overload: filtering (Information filtering, 1992) information before it reaches the end-user, and customizing the information after it arrives (Berleant & Berghel, 1994a, 1994b). Filtering remains primarily a server side activity since filtering at the client-side would necessitate unnecessary downloads. Information customization is a client-side activity designed to pick up where information filtering leaves off. In this article, we describe our vision of information customization and, along the way, chronicle the development of our proof-of-concept prototype, Cyberbrowser, for customizing information on the Web.

The Information Customization Conjecture

The long-term effectiveness of the techniques of information customization described here and elsewhere is related to the "information customization conjecture," which holds that information filtering technology will never be able to keep up with the volume of possibly relevant information. Put another way, this conjecture claims that information filtering and related automated techniques may never be able to reduce the magnitude of available data to levels which are within the bounds of the typical end-user's personal "bandwidth." Our experience with the Internet (e.g., the large numbers of hits produced by a typical Web search engine query) thus far tends to confirm this conjecture.

Thus our belief is that, in most cases, there will always be the risk that the digital networks will send more information "downstream" than the end-users can consume. Even deployed technologies that expressly intend to send only information that is of interest to the user—listservs and other e-mail lists; WWW push technology (Berghel, 1998; DeMocker, 1998)—often seem to have the effect of increasing rather than decreasing the information overload. Such existing information retrieval and filtering technologies help a user get only documents with a high likelihood of being interesting, but a further step is needed to help users deal with an individual document, customizing a user's view of it to the user's immediate interests and needs, thereby speeding up our ability to deal with the individual documents that listservs, push, etc., bring to our attention at ever-increasing rates. We have referred to this process as *Information Customization* in earlier articles (Berleant & Berghel, 1994a,b). In this article, we will discuss how the techniques of information customization may be built into network clients.

The Information Customization Metaphor

The goal of information customization is streamlined access to, and absorption of, needed information by users. While information customization can be done by humans for other humans (information brokerage), the future of information customization lies in automation because of the enormous volumes of information accessible through the digital networks.

In our view, information customization should be viewed as the latest element in the evolutionary chain of digital information handling technologies. Tools, techniques, and operational metaphors have been developed for information storage, transfer, distribution, acquisition, agency, and brokerage through the 1990s. Information customization assumes that all of these techniques, taken together, still produce information overload for the end-user, and that the optimal solution of the remaining problem is highly interactive, client-side, network-enabled software. This strategy is conveyed in the sequence of graphical transformations which appear on the splash page (Fig. 1).

Received January 9, 1998; revised May 22, 1998; accepted August 17, 1998.

© 1999 John Wiley & Sons, Inc.

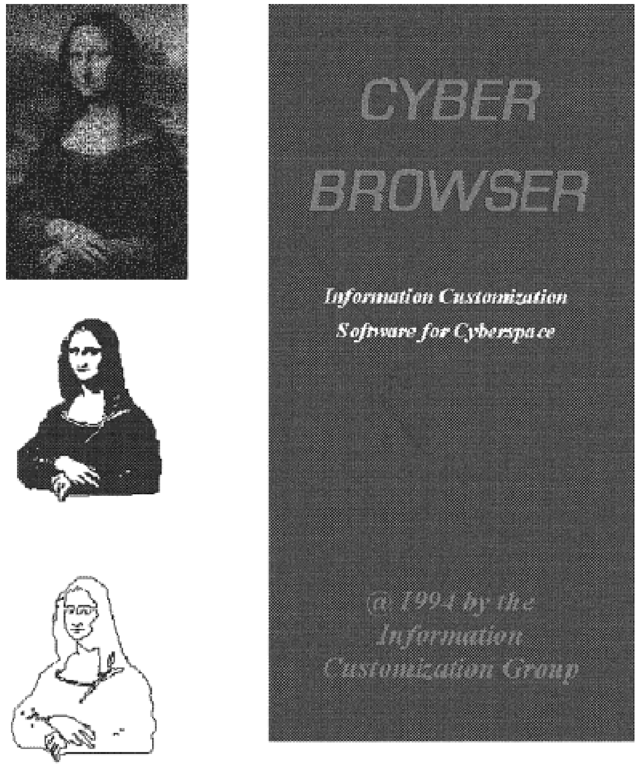

FIG. 1.

The point to be made is that continued simplifications, first from full-color to monochrome, and subsequently from monochrome to scalable outline, preserve the basic information content—at least sufficient to identify the basic theme of the original object. We encourage the user to think of the text extracts produced by Cyberbrowser as "iconic" of the entire document in much the same way as the simplified images are of the Mona Lisa.

As we write, network oriented document acquisition is redefining itself. Early strategies were modeled after techniques used to distribute digital information on magnetic media. By the late 1970s, network deflectors were serving in the role that magnetic media duplication had been serving, and early bulletin boards satisfied the same needs as archived collections of programs and data had done previously. File Transfer Protocol (FTP), together with FTP-compatible document indexing and downloading tools like Gopher and Wide Area Information System (WAIS), marked a transition in the redefinition of network acquisition tools. Although these tools were useful, they were fundamentally impaired for general-purpose network information exchange because "browsing" involved working with information which was peripheral to the document content (e.g., long file names, directory names, path structures). In retrospect, we now see that FTP indexing was a clumsy technique based on the same metaphor as the physical distribution of magnetic information—fetches based not on the content of the document, but on its label, title, file name, or location.

More than any other single technological event, Telnet showed us the way to achieve our information retrieval objectives. Telnet gave the digital networks a virtuality they had not previously enjoyed, and enclosed networked computers in a unifying cyberspace. With Telnet, networked computers became extensions of our desktop, and it was not long before Tim Berners-Lee and his colleagues at CERN came up with specifications for Internet protocols that would provide platform-independent support for distributed multimedia on the Internet. The World Wide Web was born, and with it came the concept of a navigator-browser. By early 1995, Merit NIC reported that the Web had become the leading packet hauler on the Internet in terms of both byte count and number of packets.

While the multimedia, hyperlinked structure of the Web allows users to search for information more efficiently than before, the Web alone is still suboptimal with respect to information acquisition and distribution. One of the reasons for this is that the search and filtering processes were added as afterthoughts, and not built into the original Web design. As an illustration, the content descriptor tag did not become part of the HTML standard until 1995! This tag is, in some ways, the "business part" of the HTML header and helps drive many Web spiders, wanderers, and worms which are the heart of modern search engines. In addition, requirements for information filtering and information customization are independent of the specifications for either HTTP or HTML.

Further, search engines are inherently ill-equipped to deal with high-recall, high-precision information distribution and acquisition. They work most efficiently when information is indexed, graded, and categorized prior to posting, which is rarely done. Even the simple META CONTENT = tag seems to be ignored in most documents. Since the Web did not grow out of a philosophy of preprocessing before posting, there is a definite practical limit to the performance that one may expect of search engines in the foreseeable future, no matter how finely tuned. Yet after-the-fact natural language understanding utilities remain elusive.

In general, effective document location and identification technology is becoming an increasingly indispensable link to the world of information for the modern professional. But as powerful as these tools are becoming, they are intrinsically limited in their ability to support the information consumer once information has arrived. Thus transfer of knowledge from the computer to the user is more of a bottleneck than ever before. Even client-side systems such as Bellcore's SuperBook Browser, and Digital Equipment's Lectern system, remain oriented to the information provider. It should be emphasized at this point that information customization attempts to deal with this problem by orienting itself to the information consumer.

Information customization complements existing information acquisition, distribution, and agent/broker tools, and increases their effectiveness. It has five basic characteristics: (1) It is best performed on the client side; (2) it is specifically designed to maximize information uptake, rather than filter or retrieve; (3) it "personalizes" documents by such techniques as extraction; (4) it is normally done interactively, enabling the user to have a "dialog" with a document; and (5) the software provides the capability of *non-prescriptive*, nonlinear document traversal. Condition (2) sets information customization apart from traditional information filtering and retrieval, while (4) sets it apart from information agency, and (5) distinguishes it from traditional nonlinear document traversal systems (e.g., hypertext).

In operation, information customization programs transform an information entity—such as a document—into a form that suits the needs of a particular user at a particular moment. This central intuition has considerable currency. For example, information customization is similar to Englebart's (1995) view control, as well as having strong connections to data mining (Fayyad, Piatetsky-Shapiro, Smyth, & Uthurusamy, 1996) and knowledge discovery (Piatetsky-Shapiro & Frawley, 1991).

One can think of the ideal information customizer as taking as input a triple containing a purpose, a cognitive context, and information to customize, and producing as output that processed form of the information which is best attuned to the user. The purpose may be fleeting. The cognitive context changes continually. Only the information to customize is likely to have some constancy, and when that information is a mailing list archive (e.g., Hypermail) or an institutional knowledge base (e.g., Lotus Notes) even that is no longer a given.

Since the point of information customization is to help people absorb the right information more quickly, an obvious strategy is to provide them with the information they need, withholding the information they do not need, and to provide them with that information in a user-friendly way that promotes its absorption. Furthermore, since information customization becomes more important as useful information artifacts become more accessible, information customization becomes most important in an age in which large quantities of relevant documents or other information artifacts are electronically available to the user. Thus, the information filters, Web-based search tools, and digital libraries of today and tomorrow make information customization tools increasingly indispensable.

Cyberbrowser

Our earliest work with information customization began in the late 1980s with investigations into digitally "simplifying" both images (Berghel, Roach, & Cheng, 1991) and text (Berleant & Berghel, 1994a,b). Our goal was to find ways in which we could make the important parts of volumes of data more accessible to the consumer via digital networks. Part of this work led to a prototype information customization desktop utility called "Keychain" (circa 1990), which allowed end-users to transform the presentation of a particular document according to intersecting keyword chains detected in the document by pre-processing. The successor prototype, "Schemer" (circa 1993) added hypertext capabilities and expanded the range of control that the user would have over the presentation, while at the same time adding on-the-fly preprocessing.

Throughout the development of Keychain and Schemer, we viewed information customization as a desktop-centric utility for accelerated content discovery. By 1993, when the World Wide Web was starting to become popular, we had changed our view of information customization to include network-centricity as well. This included a change of design from a stand-alone application to one which was executed from within a browser's launchpad as a "helper app."

These systems built upon a long history of text analysis and extraction, based not upon meanings and multiword structures (typified by the well-known Natural Language Understanding approach), but upon words appearing in texts and their frequencies (typified by the statistical approaches) (Edmundson & Wyllys, 1964; Furnas, Landauer, Gomez, & Dumas, 1988; Luhn, 1958; Rush, Salvador, & Zamora, 1971). The use of individual words for extraction and use of frequencies of words can be based on the actual unprocessed words appearing in texts, or can be based on stemmed versions of those words or n-gram analyses of them (see Salton, 1989). In Keychain and Schemer, we rated the relevance of words in documents by their frequencies. We implemented tabulation of both actual word frequencies and relative word frequencies. The relative word frequency

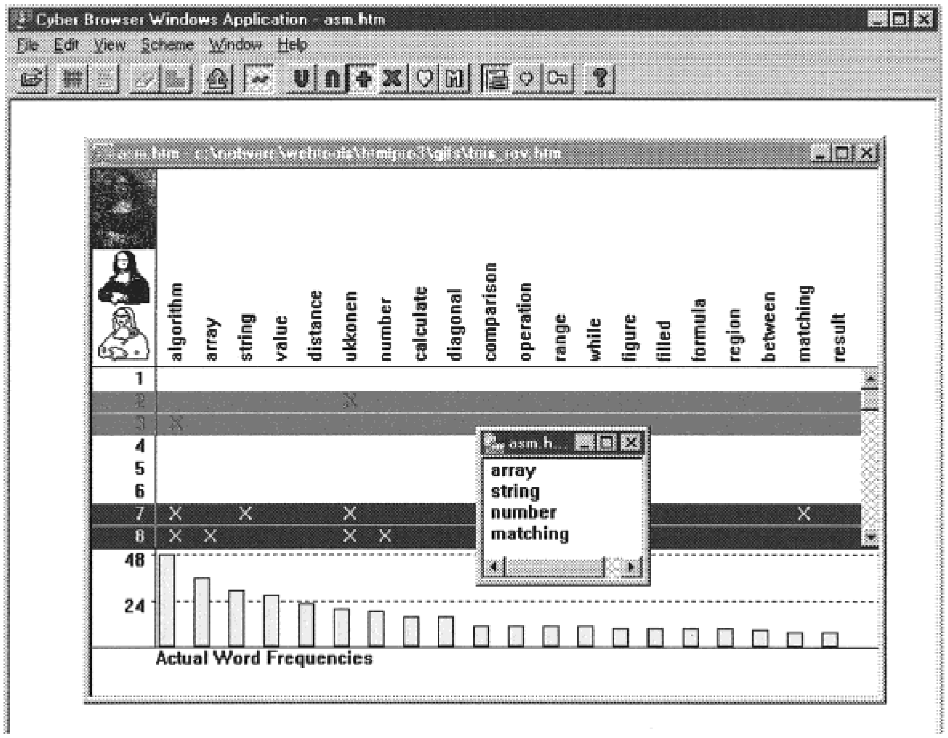

FIG. 2. Keyword-oriented document extraction.

measures modify an actual frequency of a word in a document by factoring in its background frequency, which is its frequency in some set of documents. Relative frequency measures we implemented were based on a quotient (actual word frequency in the document divided by its background frequency) and a difference (actual word frequency in a document minus its background frequency). The initial version of Cyberbrowser (Foy, 1995) allowed users to choose which of the frequency counting methods is used. Whichever one was chosen, it was used to pick out the most relevant words and present them as columns in a chart so that the user could extract and browse chains of sentences and their contexts in the document based on those words. The underlying idea of sentence chaining was also useful in constructing a different system, described next, indicating empirically that the sentence chaining approach is rather diverse in its applicability (as is, of course, the keyword-based extraction paradigm in general).

The current version of Cyberbrowser (v. 5) extends our initial design philosophy to include:

- accommodating nonpreprocessed text and HTML files
- adding a simple, intuitive, Windows-look-alike interface
- implementation of additional passage extraction operations
- compatibility with Netscape and Explorer so that documents retrieved by these Web clients are displayed using Cyberbrowser
- use of a histogram, instead of text, to display keyword frequencies
- the option to view either the customized extract of the original document by itself, or the complete original document with the customized portion highlighted
- implementation of additional logical operations including the “extract kernel” and “extract meta-kernel.” The kernel of a document is the collection of sentences which contain the greatest number of keywords. The meta-kernel sentences of a document are the kernel sentences of the *k* most frequently occurring keywords. The concept of document kernel is simply a place-holder for results of document analysis algorithms which would be available in a commercial-grade product.

The figures below show typical views of Cyberbrowser in action. Figure 2 depicts a typical keyword-based document extraction. Horizontal bands cover the affected sentences (numbers on left); keywords across the top represent the most frequent (or most significant) words in the document.

Figure 3, on the other hand, depicts a typical sentence-based document extraction. The keyword-based extraction specifically includes sentences with some keywords (dark grey bands) and specifically excludes sentences with others (light grey bands).

Note that in both figures, the presence of a keyword (top row) in a sentence number (left column) is indicated by an “x.” The bar chart at bottom plots the absolute frequency of the keywords. These keywords were extracted automatically from the document, in contrast to Hearst’s (1995) TileBars which are generated based on terms in a user’s query. (We have experimented with relative frequency measures based upon comparisons with standardized corpora, e.g., the Brown Corpus, as well as various weighted mea-

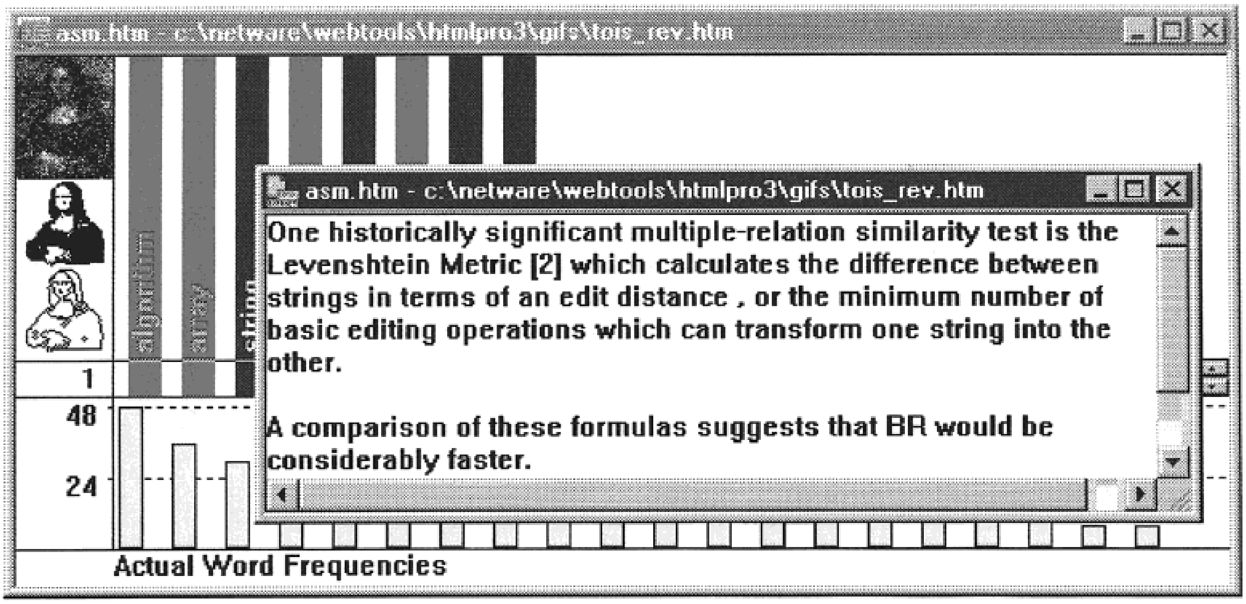

FIG. 3. Sentence-oriented keyword extraction.

sures, but deprecated these functions in the latest version of the prototype because it did not appear that the additional computational overhead was justified.) The display of frequency measures was invoked by clicking the histogram button (see below).

The light grey and dark grey bars which identify keywords (vertically) or individual sentences (horizontally) show that a manual document extraction is being performed to find sets of keywords or sentences. In use, one might read a particularly interesting paragraph and then wish to do a projection of the keywords contained in that paragraph. Conversely, one might see some keywords which correspond to current interest and wish to extract those sentences which contain them. It should be noted that sentence-oriented document extraction produces sets of keywords, whereas keyword-oriented extraction produces sets of sentences. This is analogous to the selection and projection operators in relational databases. Similarly, the text algebra operators perform in much the same way as Boolean queries (e.g., k1 & ~k2 means find the set of sentences which have keyword k1 but do not have k2; s1 v s2 means find the set of keywords which appear in either s1 or s2; etc.)

Cyberbrowsing functionality may be categorized in terms of presentation schemes, the underlying text algebra, and document extraction techniques. We define them in Table 1, below, by reference to the items on the button bar.

Cyberbrowser supports a rich set of extraction operations based on the familiar, "button bar" interface. An interesting alternative is presented by Golovchinsky (1997, e.g., Fig. 3-1), who allows words in the actual text being viewed to be highlighted via mouse clicks and used to construct a query that way. Words that are highlighted and connected via arcs to other highlighted words are ANDed, and the set of disconnected graphs of highlighted words form a set of ORed clauses. His system was intended for use in extracting distinct news articles from a large set while our intention was to browse within a single document. Both interface strategies represent quite different approaches to satisfying similar goals in an information extraction interface: Ease-of-use, understandability, and effectiveness.

Alternative Models of Information Customization

There are a wide variety of alternatives to information customization which differ from our model in certain respects. One recent approach involves the dynamic creation of HTML documents from databases. On this account, each consumer could, in principle, receive a different version of the same document(s) based upon current interests. This has some interesting similarities and differences with Cyberbrowser. The most obvious differences are that dynamic page creation from databases remains prescriptive and server-oriented. Another major difference involves the nature of the input. In the database model, the database information would be the input to the customization process, rather than the actual document(s) themselves. The customized result might differ according to the extent to which the database version of a document is different from the original document (e.g., through stripping of format, structure, or syntactical information). One additional advantage of client-based customizers is that the computational cost of customization may be amortized over several machines.

Ted Nelson has also defined an alternative. Since 1960, he has promoted progressively more advanced hypertext concepts (cf. Nelson, 1993). His Xanadu architecture is instantiated in part in the Hyper-G and HyperWave systems. A key idea in this body of work is transclusion, the including of virtual copies of passages from diverse sources into a single document (Nelson, 1995). OSMIC (Nelson, 1998), a

TABLE 1. Cyberbrowsing functionality by category.

Presentation Schemes

View Control - keyword mode. Display the n most common keywords in the document and plot them against the sentences (by number) in which they appear.

View Control - extract mode. Display the document extract called for by the query

View Reset. Erase all keyword selections and restart document analysis in current window

View Histogram. View the absolute frequency of keyword distribution in current document. (Earlier releases of Cyberbrowser offered relative and weighted frequency measures as well.)

Adjust Weighting. Change weights assigned to keywords which were found in HTML <TITLE> tags, HTML <META> tags, or document <BODY>

Text Algebra

Complement or inverse operation.

Produce the set of those sentences which contain at least one of the selected keywords, and none of the complemented keywords

Produce the set of those sentences which contain all selected keywords, and no complemented keywords

Project the set of all keywords which occur in at least one selected sentence, but do not occur in any complemented sentences

Project a set of all keywords which occur in each selected sentence, but do not occur in any complemented sentences

Extraction Modes

Extract kernel sentences - e.g., the union of the sentences which contain the greatest number of different keywords, <u>and none of the complemented keywords</u>

Extract meta-kernel sentences - e.g., the kernel sentences for the k most frequently occurring keywords

Context toggle - if enabled, only extract will be displayed; else, full text is displayed with kernel sentences highlighted

Set granularity - modifies the number of sentences on each side of a target sentence which will be included in the extract - more means added context.

Key highlighting mode - if enabled, keywords will be highlighted in the display of document, and all hyperlinks will be active.

design for a recent version of Xanadu includes a royalty payment scheme that encourages use and reuse of passages from other sources. Xanadu and its derivative works support manual generation of new documents from passages of existing documents. The scope of this project is wide and far-reaching, however it does not address the specific issues we deal with in our work here: *Automatic* creation of chains of passages in a single document. Our work is complementary to the Xanadu project, and a passage chaining interface could be used with documents created via transclusion or those written in the usual way.

XML, an eXtensible Markup Language, is another environment which enables information customization.[1] In this case, a robust system of user-defined markup tags allow documents to be "customized" before download. For example, a phone number might be written in an XML document as ⟨phoneNumber⟩501-555-1212⟨/phoneNumber⟩. Such semantic information about content might be of considerable use, since the range of the markup tags is virtually unlimited. Of course, unlike Cyberbrowser, the markup would remain.

Information Customization Architectures

In general, information customization involves an interactive process whereby users interactively, and in real time, control the means by which documents are reduced in size or transformed into a more useful form, and then displayed. Figure 3 illustrates this process in our current proof-of-concept prototype, Cyberbrowser, which behaves as either a stand-alone application or a browser-compliant, spawnable peruser (i.e., a helper app).

We have identified a number of design heuristics and principles that appear to contribute positively to the quality of an information customization system, listed below. This list attempts to codify those heuristics and principles commonly employed, implicitly or explicitly, in this and in closely related work. Continued progress in design guidelines for information customization systems requires more research and empirical observation to better understand these and other issues, including tradeoffs and other interactions. Here are these desirable design features of information customization tools:

- *Document Dialog.* Interactive information customization helps the user guide the information traversal or transformation process so that there is a fine-grained sequence of user queries followed by system responses that would ideally seem to the user almost like a dialog with the document. Cyberbrowser meets this criterion via a real-time, interactive interface.
- *Interface Transparency.* Users of information customizers should concentrate on information absorption, not on manipulating the interface. Unfortunately, the mental energy required to switch back and forth between separate interfaces for reading and querying appears to be significant, and are more likely to use information relevant to a given task if it is present on the screen than if they have to click to get it (Wright, 1991). Therefore, it is important that the interface be unobtrusive (Weiser, 1994), to avoid interfering with the goal of maximizing the efficiency of information transfer to the user. Such a transparent interface should be intuitive, and provide a small number of powerful options to avoid distracting the user from the primary pursuit of information uptake. This principle guided our Cyberbrowser interface, which provides intuitive Boolean and other operations as described earlier.
- *Input Format Independence.* The user should not need to be concerned with the data format (TXT, HTML, DOC, etc.) of a document or other information artifact to be customized. Making format considerations invisible to the user is implied by the basic requirement that information customization provide information in a way that is suited to the user. Cyberbrowser processes documents in either HTML or TXT formats. Substantive compliance with this principle is more of a commercial than a research issue, so we have not attempted to extend our systems to such common formats as postscript, PDF, MS Word, WordPerfect, etc.
- *Multiway Lookahead.* What a user most needs to see next is likely to be related to what the user is seeing currently. However, there are likely to be a number of related items in the document. Multiway lookahead means computing several related items and displaying them simultaneously. There is then a relatively high likelihood that what the user will want to see next is present without the need for another query. Then, the user can simply go on to read one or more of the preprocessed documents without the mental overhead of thinking about what to request to see next.

  Ranked lists of retrieved titles (as in Web search engines) take a first step in the direction of multiway lookahead by presenting multiple alternatives, although these alternatives are not meant to be read without additional retrieval. As Golovchinsky and Chignell (1997) point out, electronic newspapers and paper newspapers display different articles on the same page and are easy to read in part for this reason. Thus, newspapers are an example of multiway lookahead, and this is one reason why they chose the newspaper metaphor for their information exploration interface design. We are currently investigating the multiway lookahead paradigm with prototype "multibrowsers" based on a variety of different design strategies for both client and server (cf. Kamba et al., 1995).
- *Non-Insularity.* Information customizing services will be most useful when used with other systems that provide information. When mature, information customization tools will be menu items of everyday word processors, desktop publishing software, Web navigator/browsers, and so forth. They will complement the existing client-server base, including a wide variety of client-server browsers, locators, e-mailers, and transfer programs (cf. Berghel, 1997). Those programs will provide back ends to customizing modules. This is why we emphasize Web compatibility in Cyberbrowser.

[1] XML is defined by the World Wide Web Consortium XML at http://www.w3.org/XML; see also, Fitch, K., XML, http://www.csiro.au/itsb/kent/imtc97/presentation/sld010.htm

- *Nonlinearity.* By nonlinearity, we mean that the order of presentation of information from a document or other information artifact is not determined by its physical or digital layout. A paradigm nonlinear viewing environment is hypertext, which is an essential element of the Web's HTML language definition and document layout strategy. Another popular nonlinear information presentation model is the newspaper (e.g., multicolumn pages present information in parallel, nearness to the front signifies article importance facilitated by liberal use of "continued on page n" pointers, etc.). A hypertext/newspaper hybrid combining the best nonlinear characteristics of each has been explored in Golovchinsky (1997) and Golovchinsky and Chignell (1997).
- *Nonprescriptiveness.* By nonprescriptiveness, we mean the ability to transform or traverse an information artifact in ways which were not prescribed by the information provider. In contrast, hypertext is typically a prescriptive environment: the anchors and links in a hypertext document are typically prescribed by the author and hard-coded into the document. While this allows for nonlinear traversal, it is prescriptive.

  Nonprescriptiveness means that the document may be traversed or processed in useful ways that were unforeseen, and perhaps even unintended, by its creator. This makes it possible for an information artifact to be more flexibly customized. Nonprescriptive links are possible if they are generated as needed, in response to retrieval queries. This approach combines the nonprescriptiveness of retrieval queries with the utility of hyperlinks in a hybrid that blurs the distinction between queries and links (see also Golovchinsky, 1997).

  Web search engines form another example of nonprescriptive linking. The response of a search engine is a list of dynamically generated links, built in direct response to the user's needs. A book's index, on the other hand, is generated by the author and is prescriptive to some degree (a poor index is very prescriptive, and a long and comprehensive one is less so). Nonprescriptiveness is akin to Englebart's (1995) "Every object intrinsically addressable" concept. Chaining systems like Cyberbrowser are nonprescriptive in that chains are found in the text irrespective of the intent of the author.
- *Real-Time Performance.* The whole point of information customization is to speed up the transfer of useful information to a user's mind, so making a user wait for the system to compute obviously detracts from the performance of an information customizer. Real-time performance is all the more important in highly interactive settings where information to be presented is transient and must be recomputed frequently. Cyberbrowser's interactive interface exemplifies this criterion.

Improved understanding of these and perhaps other architectural principles, their interactions, and their conditions of application, relate not only to our own immediate research goals but also to important related work such as visual data navigation, Web resource locators, database mining, and others.

Conclusion

Information customization is becoming increasingly important as modern information access methods make overwhelming quantities of information electronically available to the individual user.

In this article, we have outlined design heuristics and principles for information customization systems. We also overview our own information customization prototype, Cyberbrowser, which is the latest of three generations of our client-side, information customization programs. Cyberbrowser is designed to integrate fully with Web navigation/browsing clients.

As of this writing, we are developing prototypes of another information customization strategy, multibrowsing. Since what a user will want to read next is obviously not fully predictable, the multibrowsing strategy is to present several likely alternatives simultaneously. The operative design principle, we term multiway lookahead.

We expect that information customization, in due course, will take its place along complementary information technologies and play a useful role in dealing with information overload.

Acknowledgments

We thank the anonymous JASIS reviewers and the Editor-in-Chief for thoughtful comments on earlier drafts of this article.

References

Berghel, H. (1996). The client side of the Web. Communications of the ACM, 39(1), 30–40.

Berghel, H. (1997). Cyberspace 2000: Dealing with information overload. Communications of the ACM, 40(2), 19–24.

Berghel, H. (1998). The new push for push technology. Networker, 2(3), 28–36.

Berghel, H., & Berleant, D. (1994). The challenge of customizing cybermedia. Heuristics: The Journal of Knowledge Engineering & Technology, 7(2), 34–43.

Berghel, H., Roach, D., & Cheng, Y. (1991). Expert systems and image analysis. Expert Systems: Planning, Implementation, Integration, 45–52.

Berleant, D., & Berghel, H. (1994a, September). Customizing information: Part 1, Getting what we need, when we need it. Computer, 96–98.

Berleant, D., & Berghel, H. (1994b). Customizing information: Part 2, How successful are we so far? Computer, 76–78.

DeMocker, J. (1998, March 23). Enthusiasm fades for browser channels. Internet World.

Edmundson, H.P., & Wyllys, R.E. (1964). Problems in automatic abstracting. Communications of the ACM, 7(1), 259–263.

Englebart, D.C. (1995). Toward augmenting the human intellect and boosting our collective IQ. Communications of the ACM, 38(8), 30 ff.

Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., & Uthurusamy, R. (1996). Advances in knowledge discovery and data mining. Cambridge, MA: MIT Press.

Foy, T. (1995). Cyberbrowser. (MS Project Rep.). Fayetteville, AR: University of Arkansas, Department of Computer Science.

Furnas, G., Landauer, T., Gomez, L., & Dumas, S. (1988). Statistical semantics: Analysis of the potential performance of keyword information systems. Bell Systems Journal, 62(6), 1753–1806.

Golovchinsky, G. (1997). From information retrieval to hypertext and back again: The role of interaction in the information exploration interface. Unpublished doctoral dissertation, Department of Mechanical and Industrial Engineering, University of Toronto.

Golovchinsky, G., & Chignell, M. (1997). The newspaper as an information exploration metaphor. Journal of Information Processing and Management, 33(5), 663–683. Online preprint available: http://anarch.ie.utoronto.ca/people/golovch/publications/ipm_en

Hearst, M. (1995). Tile Bars: Visualization of Term Distribution Information in Full Text Information Access. Proceedings of the ACM SIG CHI Conference on Human Factors in Computing Systems, (pp. 59–66). New York: ACM Press.

Information filtering [Special feature]. (1992). Communications of the ACM, 35(12), 26–81.

Kamba, T., Bharat, K., & Albers, M.C. (1993). The Krakatoa chronical—An interactive, personalized, newspaper on the Web. Fourth International World Wide Web Conference. Online preprint available: http://www.w3.org/Conferences/www4/Papers/93

Luhn, H.P. (1958). The automatic creation of literature abstracts. IBM Journal, 159–165.

Nelson, T. (1993). Literary machines. Sausalito, CA: Mindful Press.

Nelson, T. (1998). OSMIC open standard for media interconnection. Available: http://www.sfc.keio.ac.jp/~ted/osmic.w1/osmic.d1m.webconv.html

Nelson, T.H. (1995). The heart of connection: Hypermedia unified by transclusion. Communications of the ACM, 38(8), 31 ff.

Piatetsky-Shapiro, G., & Frawley, W. (Eds.). (1991). Knowledge discovery in databases. Cambridge, MA: MIT Press.

Rush, J., Salvador, R., & Zamora, A. (1971). Automatic abstracting and indexing. Journal of the American Society for Information Science, 260–273.

Salton, G. (1989). Automatic text processing. Reading, MA: Addison-Wesley.

Weiser, M. (1994). The world is not a desktop. Interactions, 1(1), 7–8.

Wright, P. (1991). Cognitive overheads and prostheses: Some issues in evaluating hypertexts. Proceedings of Hypertext '91 (pp. 1–12). New York: ACM Press.