From NoteCards to the Web: The Role of Halasz's Seven Issues

Kate Dobroth
Mindframe Design
Jamaica Plain, MA 02130
kate@mindframedesign.com

Abstract

The issues that Frank Halasz describes in "Reflections on NoteCards" are relevant to both creating and browsing through hypermedia networks. In order to bring hypermedia out of research labs and into widespread use, it was necessary to first focus on issues related to browsing, specifically, in making information easy to find. It was also necessary to integrate computation with hypermedia in order to create worthwhile applications. Now that users are comfortable accomplishing basic tasks on the Web, there is increasing interest in developing more complex, collaborative applications. For this reason, the issues that involved authoring and support for collaborative work have new relevance today.

Keywords

hypermedia, World-Wide Web, user interface design, usability

In 1988, the Internet was widely used by Unix-running academics for email, newsgroups, and exchanging data. Apple was just about to release Hypercard, and it was about five years before the general public would know about the "information superhighway." Frank Halasz was a member of a team of researchers at Xerox PARC that developed NoteCards, a hypermedia authoring and knowledge representation tool. In his landmark paper "Reflections on NoteCards" (1988), Halasz summarizes seven key issues that he believed would be critical in the development of next-generation hypermedia systems. His discussion was based on a comprehensive review of the strengths and weaknesses of the hypermedia systems that existed at the time, as well as insights gained from his observations of the "use and misuse" of NoteCards.

It is remarkable to read this list of hypermedia issues in 2001. Simply put, Halasz summarizes seven issues that were critical in the evolution of early hypermedia systems into the Web as it is today. The first issue, search and query, makes it possible for users to find information in a network, even when the network is unfamiliar. Issue 2, composites, and issue 3, virtual structures, describe two constructs that make it possible to maintain large amounts of consistently changing information. Adding computational capabilities to hypermedia (issue 4) allowed the development of applications that went beyond simple storage and retrieval. Issue 5, versioning, and issue 6, support for collaborative

ACM Journal of Computer Documentation, 2001,25:88-91

work, are increasingly relevant as we move from an individual work model to a model of groups collaboratively and synchronously creating documents. And issue 7, extensibility and tailorability, stresses the importance of maintaining flexibility so that users are able to create the types of applications that they need.

The commentaries in this journal issue explore how well "Reflections on NoteCards" has stood the test of time. Of course, Halasz himself has already done this at least once, having updated the seven issues for a keynote presentation at Hypertext '91 (Halasz, 1991). In that presentation, he called for researchers and developers to consider how to move hypermedia out of labs and into widespread use. In keeping with that, Halasz introduced three new issues that were related to large, networked hypertexts. Perhaps anticipating that casual users might initially be more likely to want to find information than to create their own networks, the revised list seems to have a reduced emphasis on authoring issues. Issue 5 (versioning) was removed altogether. Issues 2 and 3 (composites and virtual structures), which concern the ability to easily manipulate networks, were combined with issue 1 (search and query) to create a single issue (ending the tyranny of the link).

In this paper, I examine the role that the seven issues played in making the Web widely accessible to casual, nontechnical users. In the early days of the Web, it was of utmost importance to make it easy for inexperienced users to find information. As users began to demand more sophisticated applications, developers introduced more advanced computational abilities. Most recently, there has been a surge in interest in the development of collaborative applications. For this reason, the issues that involved authoring and support for collaborative work have new relevance today.

Ending the Tyranny of the Link: Easy Access to Information

Although NoteCards included a limited search capability, finding information in NoteCards was usually a matter of moving through the network link by link until the desired information appeared. Halasz noted that, although there was "something alluring" about finding information in this way, next-generation hypermedia would need to provide more efficient methods of accessing information. For this reason, he proposed that the basic node and link model be augmented with search capabilities. He described two types of search: *content* search and *structural* search. Content search was the application of standard information retrieval techniques to hypermedia, so that users could enter queries and be brought directly to nodes containing matching text. Structural search would search for nodes located in a particular structural pattern in the network, regardless of content.

In 1991, Halasz noted that the structural search hadn't really caught on, and the same seems true today. Most commercial search engines do not explicitly offer structural search. One exception is www.google.com, which allows users to find all sites that connect to a particular site; however, this capability is hidden in the advanced search. It is not clear that structural search has much to offer to casual Web users who did not create the network that they are searching. Instead, structural search seems as though it would be more useful to an author or a person who was manipulating a network. Content search, on the other hand, is absolutely crucial to the Web as we know it today. If a user wants to find a particular type of information and does not have a specific site in mind, he or she must first use a search engine to find a list of potential sites. Halasz predicted in 1991 that there would be a tighter integration of hypermedia and information retrieval, and he was right. Search engines have progressed beyond keyword searching to include conceptual searching (such as www.excite.com) and relevance ranking to help users to sort through vast quantities of results. User interfaces for searching have also improved. For example, advanced search pages help users to use the full power of the query language more effectively by providing a graphical user interface. Increasing numbers of sites (such as www.askjeeves.com) also support natural language queries, where users can type questions using everyday language.

In addition, local search capabilities are integrated throughout individual sites. In the 1991 presentation, Halasz called for retrieval through search to be on "equal footing" with retrieval through links, and noted that there might be a blurred distinction between links and searches. This is certainly the case in the present-day Web. For example, when a user types in a zip code to see a weather report for a particular city, is this a search or a "dynamic link," as Halasz suggested in 1991? Does it matter? Personalized portal pages are even further removed from the tyranny of the link. Here, a user indicates the sorts of information that he or she wants to appear on a page, such as stock quotes or categories of news stories. The information is automatically retrieved when the page is loaded. With cookies, users may not even have to identify themselves before their information appears. An interesting question is whether retrieving information without following links of any sort still falls under the definition of hypermedia. But again, does it matter?

It is important to remember that content searching is only one component of information accessibility. Web designers have also made good progress in understanding how to develop usable navigation schemes for any given Web site. Among its limited navigational capabilities, NoteCards included *browser pages*, which seem similar to present-day site maps. Halasz noted in 1991 that there would need to be different approaches to navigation for very large information spaces. A site map may accurately reflect the content structure of a Web site; however, a successful navigation scheme is based on users' tasks, and may or may not resemble the content structure.

Computation over Networks: User Interface Design Challenges

In the 1988 paper, Halasz recognized that computational capabilities would be necessary for hypermedia applications to progress beyond passive storage and retrieval systems. He envisioned that computation could occur either locally, where the computational engine is integrated with the hypermedia, or remotely, where data is sent to a remote computational engine that processes results and sends them back. Hypertext Markup Language (HTML) and simple scripting languages are limited in their computational abilities, which means that in most sites computation occurs remotely.

Although the addition of computational abilities resulted in more powerful Web applications, it is interesting to note how remote data and computational abilities have influenced user interface design for Web application as compared to desktop applications. Users bring to the Web a set of expectations that is based on their experiences with desktop applications, where computation and storage are both local. At times, it can be confusing to use a Web browser that runs locally to manipulate data that is stored remotely. Web designers have had to develop new user interface models for accomplishing tasks on the Web that users are accustomed to doing in desktop applications. For example, saving data on the Web often requires an explicit action on the user's part (e.g., clicking a Save or Submit button). A remote server cannot always track whether users have made changes, and so it is not always possible to prompt users to save changes if, for example, they close a browser window. Direct manipulation (e.g., drag and drop, highlight and delete) has only recently begun to be available in Web applications through the use of Java applets. Without direct manipulation, some tasks that are simple on a desktop, such as renaming, deleting, and re-ordering a list of items, tend to be cumbersome and unintuitive. Java makes it possible to temporarily download additional computational abilities to the client side. Presently, not all users have fast enough connections or powerful enough processors to support Java and other advanced features. Nevertheless, Java may prove to be an effective tool for creating more powerful, intuitive user interfaces.

Support for Collaborative Work

Now that users have become more comfortable with accomplishing simple tasks on the Web, there is increased interest in releasing applications that allow groups to collaborate over the Web. For example, Groove Networks (www.groove.net) recently released Groove, a collaborative platform that is being marketed to ordinary consumers as well as to companies and knowledge workers. In his discussions of collaborative work, Halasz was primarily concerned with two issues: technologies for shared databases and support for social interactions during collaborations. In the 1991 presentation, he stated that, while there had been some progress on the "mechanics of multi-user access," there were very few tools that supported social interaction. Specifically, he found that there were almost no tools that supported *procedural* collaborative activity, that is, activities that control the mechanics of the collaboration. Since then, real-time interactive tools such as instant messaging, chat, and shared whiteboards have become commonplace. More recently, speech functions have been introduced (for example, in Microsoft's NetMeeting), so that groups of users can speak to each other. Groove includes a "Navigate Together" function that allows groups of collaborators to ensure that they are all looking at the same page. In addition, asynchronous reviewing tools, such as the reviewing and commenting functions available in many Microsoft applications, are also relatively common, providing support for annotative activities.

Although asynchronous collaboration has become relatively common, it is less common to see applications that allow groups of users to collaborate on single documents synchronously. Synchronous collaboration presents many user interface challenges: when multiple users are allowed to edit a document concurrently, there is a risk of conflicting operations, such as when one person is trying to edit an item that another person is deleting. Halasz suggested several methods for avoiding conflicts, such as concurrency control, where only one user at a time can modify a piece of information, and notification, where users are notified about important changes to a document. As synchronous collaborative tools become more common, one challenge will be to design user interfaces that help users to avoid collisions. Even though real-time interactive tools may help users to avoid conflicts, these tools may not always be integrated into collaborative tasks in a way that helps users to achieve mutual intelligibility. For example, a user may be focusing on editing a piece of text in one region of the screen, and not see an instant message that appears in another area of the screen that indicates that someone else is about to delete that text. Integrating interactive tools into collaborative tasks will probably still require a rhetoric of hypermedia, as Halasz suggested. And until then, versioning (issue 5) will be extremely important. In the 1991 paper, Halasz expressed doubt as to whether versioning was an important issue anymore. Whether single users make use of versioning is a matter of individual work style. When groups collaborate, however, versioning may be more important, if only because of the increased risk of conflicting operations. Even with real-time interactive tools, miscommunications are bound to occur in group work, which increases the likelihood of erroneous changes. Easy-to-use versioning tools will be necessary to help groups recover from errors.

Conclusion

A common element that runs through all of Halasz's issues is a desire for all types of users to be able to use hypermedia to create and navigate through networks of information. The Web has made great strides toward achieving that goal. The issues that Halasz raised were all necessary to attract wide numbers of users to the Web. First and foremost, users had to be able to find information. As users grew more comfortable with finding information on the Web, it became necessary to offer applications that did more than passive storage and retrieval of information. Applications that allow groups to collaboratively create their own networks of content are becoming increasingly popular. Nearly a decade later, all of the issues that Halasz discussed are still active areas of development, as developers and designers continue to work to develop worthwhile, easy-to-use applications for the Web.

References

Halasz, F. (1988). Reflections on NoteCards: Seven issues for the next generation of hypermedia systems. *Communications of the ACM*, 31(7), 836-852.

Halasz, F. (1991). "Seven issues": Revisited(keynote address). Hypertext '91