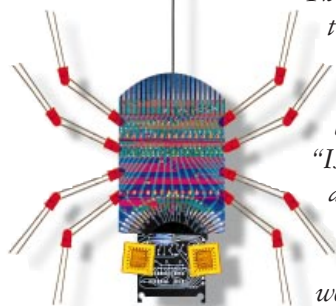


Document Management and Web Technologies: Alice Marries the Mad Hatter

Wedding the mature document management functionality (Alice) to chaotic, evolving Web technologies (the Mad Hatter) does not always come with a storybook ending.



The Hatter went on in a mournful tone, "And ever since then, time won't do a thing that I ask! It's always six o'clock now." A bright idea came into Alice's head. "Is that the reason so many tea-things are put out here?" she asked. "Yes, that's it," said the Hatter with a sigh: "It's always tea-time, and we've no time to wash the things between whiles." "Then, you keep moving round, I suppose?" said Alice. "Exactly so," said the Hatter: "as the things get used up." "But, what happens when you come to the beginning again?" Alice ventured to ask. "Suppose we change the subject," the March Hare interrupted [5].

Frequent prototyping during our early efforts to build a Web information system (WIS) reminded us of the Hatter's comment, "it's always tea-time ..." We often changed design strategies to take advantage of ever-changing Web technologies, making it difficult to move forward with "no time to wash the things between whiles." Also, the advent of easy-to-use Web authoring and management tools increased the desire for instant publishing, thereby trivializing the need for careful planning, foresight, and a systematic design methodology.

WIS designers must resolve issues of authoring, organizing, managing, and delivering large amounts of unstructured and timely information via the Web. They currently must integrate a myriad of third-party products with the Web. This article describes how we used a systematic design methodology and document management facilities to offset Web deficiencies in constructing a large-scale authoring and publishing system. The marriage of document management and Web technologies experienced problems and compromises.

A Marketing Information System Wonderland

Merrill Lynch provides financial management and advisory services to millions of households and businesses. Through its Trusted Global Advisor (TGA) initiative, Merrill Lynch is replacing its text-based, mainframe information systems with client/server and Web-based systems integrated under a single graphical user interface shell [6]. TGA enables internal marketing groups to deliver marketing information in many formats and media to financial consultants via an intranet. Subsets of this information may also be delivered to clients and the general public.

The systems development team received the following requirements:

- Manage and deliver large amounts of unstructured material in multiple media
- Provide a consistent and predictable information structure, user interface, and navigational mechanism
- Enable linking of related materials
- Ensure information is up-to-date
- Facilitate non-technical authors in creating the content
- Support well-defined roles, responsibilities, and access control for various stakeholders in various departments
- Enable workflow between authors, product managers, content administrators, editors, attorneys, and system administrators
- Enable the composing and publishing of different views of marketing information for different audiences: financial consultants, clients, the public

Document Management (Alice)	Web Technologies (Mad Hatter)
Manage large amounts of material	Deliver multiple media
Provide consistent and predictable structure	Provide user interface and navigation
	Enable hyperlinking
Ensure currency	
Facilitate non-technical authors with templates	Facilitate non-technical authors with WYSIWYG tools
Support roles, responsibilities and access control	
Enable workflow	
Publish multiple views	
Enable version control	
Provide document locking	
Enable recording of attributes	Enable attribute searching using metatags
Stable, well-defined functionality	Continuously evolving

Table 1. “Who brings what to the marriage?”

- Provide version control to support regulatory requirements
- Provide a locking or concurrency control mechanism to prevent two or more people from simultaneously updating the same content
- Enable searching and retrieval of content using predefined business characteristics of products and services

In their classic article, Malcolm et al. [8] specified similar requirements for an industrial strength hypermedia system within an engineering enterprise. We faced the same kinds of issues such as interactive authoring, templates, composite documents, object attributes, navigational aids, access control, version control, concurrency control, query mechanisms, interoperability, and collaboration.

Integrating Document Management and the Web

Alice Marries the Mad Hatter

The traditional Web model of authoring and managing content on a file system could not support our requirements, with over 10,000 documents. One goal was developing a low-maintenance information system, easily managed by end users. Responsibilities would be delegated and dispersed to various stakeholders in the organization. Marketing departments, products and services groups, editorial, legal, and technical staff would collaborate through a centralized document repository, minimizing systems support personnel.

Due to the highly unstructured nature of the material, a relational database management system (RDBMS) was insufficient. Information about prod-

ucts and services such as mutual funds and cash management accounts are too dissimilar for any generalized data model. We did not have enough expertise to explore OODBMS. Document management functionality satisfied many requirements. Documentum™ most closely matched our needs. It is an object-oriented client/server system residing on a relational database facilitating the

storage, management, and retrieval of documents in multiple formats. It also can store documents as components, assemble them into different views, provide privileged access to authors, and support workflow.

When starting the project in early 1996, document management and Web technologies were not integrated. We were challenged with building a hybrid system that combined and leveraged the strengths of both. We did not realize that integration would be a daunting task. Although the two technologies complemented each other (Table 1), this was not a marriage made in heaven.

System Architecture, Challenges, and Issues

The Newlyweds

To build successful WIS, Bieber et al. [2] urged developers to observe systematic design methodologies, integration, and evaluation techniques. As we describe the highlights of our design methodology

and system architecture, we will also present the challenges we faced. At the heart of the architecture (Figure 1) is a centralized document management system (DMS).

Hypermedia functionality concerns structuring, linking, and navigating related material. We reviewed existing hypermedia design methodologies such as the relationship management methodology (RMM) and the object-oriented hypermedia design method (OOHDM) [7,10]. Both require the application domain to be abstracted into entities or classes and relationships. As mentioned earlier, we could not arrive at a generalized entity-relationship model. Unable to apply existing methodologies to such a

within that structure. A challenge was in defining roles and ACLs without becoming too complex. We had to balance strict control and flexibility.

We started with a sample product area and prototyped the organization and information structure of its products. At a physical storage level, all product information was organized within a flat hierarchy, but logically could be organized in any way. Hyperlinks allowed the logical organization to differ from the physical structure. This simplified authoring and linking, facilitating automation of the publishing process. Information about products were separated into components (Figure 2) such as Description, Client Suitability, Client Benefits, Performance, Sales Charges, Phone List, Risk, Marketing Materials, Sales Ideas, and Competition. However, not all products shared the same set of components.

We also prototyped the user interface and navigation mechanisms. The left frame (Figure 2), generated by the system, would function as a local navigation mechanism while the top frame would function as a global navigation mechanism providing access to other areas of interest. The body frame contains product information assembled out of components that were authored separately by product marketing experts.

While the business liaison insisted on a consistent information structure and interface,

product marketing groups wanted more autonomy and creative license. A continuous debate raged between financial consultants who wanted an organized product reference guide and product marketing groups opting for a flashy Webzine.

HTML component templates included authoring instructions as comments. With WYSIWYG authoring tools, authors do not have to learn much about HTML. Due to the infancy of authoring tools, we could not provide foolproof templates as requested by the business team.

HTML authoring tools were not truly WYSIWYG and were being upgraded continuously, making it difficult for us to select a standard. Though we provided an HTML extension to a familiar word processor, authors still required hands-on training. An author's lack of familiarity with the Web and resistance to change existing authoring practices posed problems.

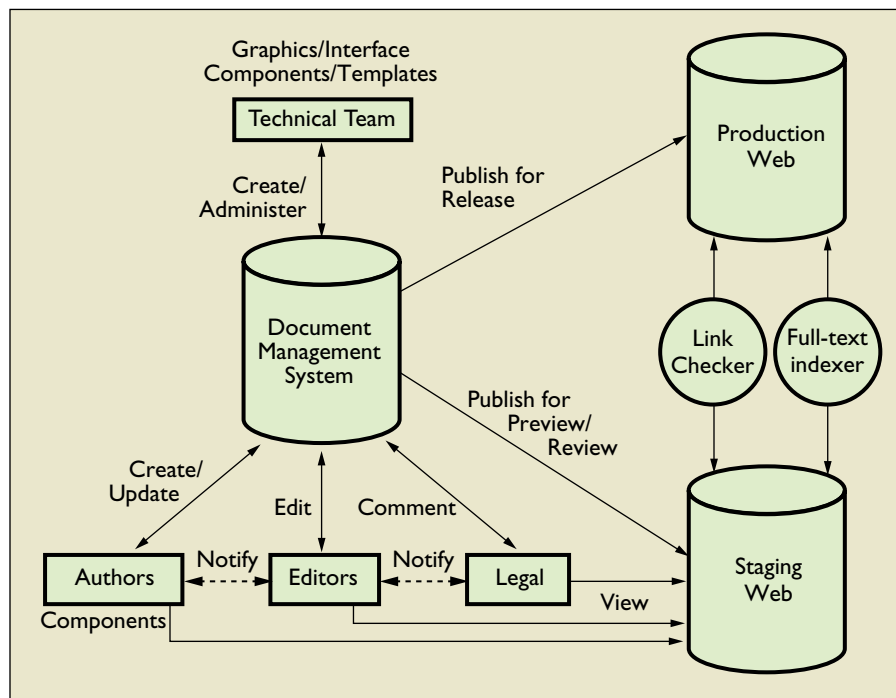


Figure 1. Software architecture of the authoring and publishing system

large-scale, unstructured, document-oriented WIS, we decided to derive our own by extending Isakowitz et al.'s RMM [7]. Our development methodology included seven iterative stages: information architecture, user interface and navigation design, content creation and authoring, workflow and document management, publishing, document review and link management, and search and retrieval [1].

Based on roles, client interfaces to the DMS were created for various stakeholders in the organization. This required significant customization and training. Privileges and access control lists (ACLs) were defined, providing authors secured access to permitted documents. Administrators define the product information structure with templates while authors create content

In addition, we faced backward compatibility problems even between versions of the same tool, only months apart, as well as major incompatibilities between different vendors' tools. In the end, we added a publishing process to perform some minor HTML formatting adjustments.

The authoring and publishing cycle begins when an author checks-out a predefined component of a particular product. The DMS locks the document, preventing two or more people from simultaneously

integral part of the authoring process, there would be less room for error. Some authors found it difficult to understand the concept of linking. It is a nonlinear activity and it is not how people are accustomed to writing and presenting content.

Each DMS component has standard attributes such as Title, Keywords, Author, Creation Date, Modified Date, and Version Number. Authors assign component-specific attributes such as Suitable Target Audience. This later guides assembling different

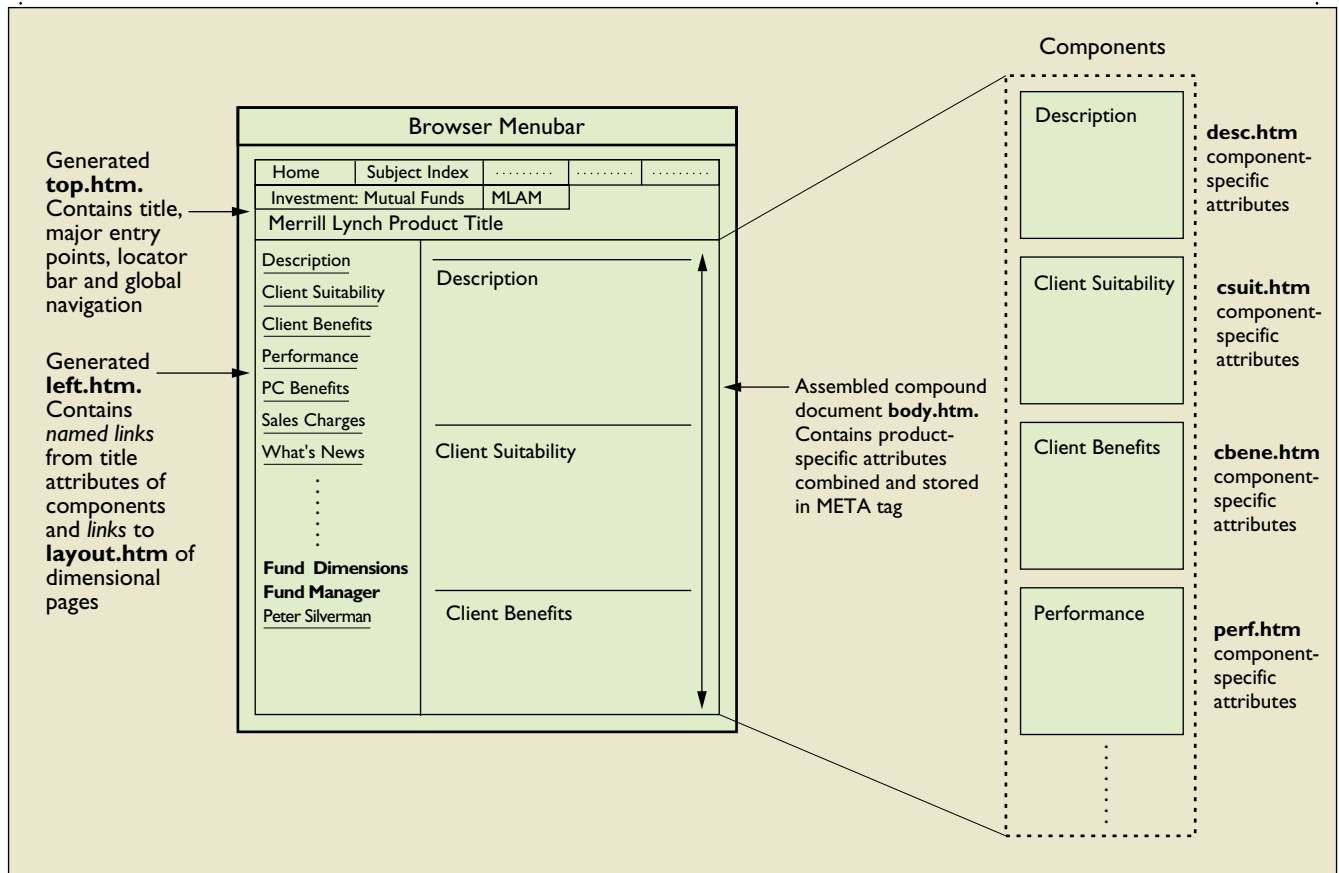


Figure 2. Component-based authoring and publishing

working on it. After editing, the author checks the document back into the DMS, assigning it an appropriate version number. Authors and system administrators have access to older versions for regulatory and legal reasons.

Authors can create links to related products. They browse through a dynamically generated list of URLs to all information currently in the system, displaying each item's contents in a different frame. The link's relative URL can be copied and pasted. Thus, we minimized dead link problems by ensuring that authors could correctly create links only to documents that already existed. This relies heavily on authors following our linking procedures. If link verification were an

components into views for financial consultants, clients and the public. For example, components such as Description, Client Benefits, and Client Suitability can be delivered to all three audiences. Components such as Financial Benefits, Phone List, Ordering Information, and Competition pertain to financial consultants only. Theoretically, combining various components to create multiple views avoids reauthoring. As it turned out, audiences were so different that authoring a component once for all readers was not appropriate.

Authors also define product-specific characteristics such as the title, client suitability, and risk level. During publishing, these attributes are written as HTML

metatags, used for search and retrieval. Web servers running search engines continuously index content. This enables users to retrieve content using attribute-based and full-text searches. The initial attribute list applied to a majority of products, but became irrelevant for others. For example, an equity may have a stock symbol and a risk attribute, which are not applicable to an account. As the attribute list grew to cover all products classes, it became difficult to control and provide meaningful search results.

An author previews work by publishing to a Staging Web server and viewing with a browser. The publishing process generates the user interface, global and local navigation components (top and left frames in Figure 2), thereby maintaining consistency. An appropriate view (body frame of the layout shown in Figure

2) is constructed by combining components based on the target audience attribute of each component defined by the author. Versions of these generated documents are also stored in the DMS for future reference. Efforts to standardize the user interface at the corporate level required us to change it a few

times. This was easily accomplished by making minor changes to our publishing procedures, without rewriting all documents by hand. All product documents can be republished easily to reflect any changes.

Once an author is satisfied with his or her work, an email router within the DMS sends a notification to a group of editors. An editor reviews the document, adds comments, republishes it to the staging server, and forwards the document URL to a company attorney. If approved, the editor notifies the author and requests an administrator to promote the document for release to the Production Web server. There was resistance to adapt to this new workflow. Editors and attorneys were not very enthusiastic about reviewing and approving content online. Additionally, attorneys were supported by secretaries and relied on handwriting for authenticity. Also, problems arose due to incompatibilities in network connectivity and email software, which are slowly being resolved.

System administrators review the documents on both Staging and Production Web servers. Link

verification engines are run on both servers to report any dead links. Hence, dead links are only detected after the fact. There is no immediate notification of document deletion, even if it results in dead links. This completes the authoring and publishing cycle. Periodic notifications are sent to authors by marketing groups to reexamine content for currency.

Both the user interface and the search interface have been evaluated through informal usability studies with mixed results. While some financial consultants were familiar with the Web, others expected the system to function like a typical event-driven application. We observed the same even with usability specialists. Technical staff and authors hold weekly feedback sessions called "The Authoring Cafe" to exchange notes about the system.

Some authors were overwhelmed by simultaneous training in multiple areas such as information mapping, authoring tools and guidelines, client interface to the DMS, and new work processes. Delays due to the development of such an integrated system had quite an impact on the organization. The desire to

Although the Web has simplified information delivery, WIS development is equally, if not more, challenging than traditional IS development. IS managers and users generally do not recognize this fact.

be out there quickly drove some well-funded marketing groups to break away from the standards imposed by centralized control of content towards a more Webzine-like approach. They tried to adopt their own tools for authoring and version control, however, they have not fared well with internal regulatory staff and corporate standards.

Using the DMS, authors and marketing groups update and publish Web documents as information changes. Primarily, we adopted a static publishing model. Information is published from the DMS to the Web site since much of the content is static and must be legally approved. However, information derived from dynamic data sources can also be integrated with this publishing model using dynamic HTML and active server pages.

We evaluated other Web-enabled document management products, including one from Documentum, which provide gateways to a dynamic publishing model. Early product releases were not robust. Some had retrofitted their client/server systems to the Web without fully integrating document management

functionality. Current DMS/Web gateways address Web integration better.

Related Work

Supporters of the Marriage

Bieber et al. [2, 3] compare the existing Web infrastructure to second-generation computing languages. They recommend incorporating third- and fourth-generation hypermedia features into the Web. In addition, the Web lacks in other equally important functionality pertaining to document creation, management, delivery, and retrieval. This concern is also shared by Rein et al. [9] who iden-

tify the need for enhanced functionality to manage collections, support document management functions such as access control, structural and content version control, concurrency control, change notification, and integrate link management functions within Web servers. They call for integrating document management functionality into the Web infrastructure.

Efforts of the IETF Working Group to define standards for a distributed authoring and versioning protocol for the Web (WEBDAV) are a step in the right direction. Whitehead [11] describes the proposed extensions to the HTTP protocol to provide meta-

Alan R. Dennis

Lessons from Three Years of Web Development

In 1994, we began developing our Web-based groupware system called TCBWorks (tcb-works.cba.uga.edu) [2], which was installed by 300 organizations around the world and used by thousands of people at our public site [3]. In 1997, TCBWorks became the commercial product called Consensus anyWARE (www.softbicycle.com). We learned four key lessons from this project.

Don't Chase New Technologies.

The rapid evolution of Web technologies can create a nightmare. When one version of a system is complete, a new round of technologies emerges. Suddenly, there is pressure to redevelop the system to use the latest and greatest technology. This is a fundamental process. New technologies will continue to emerge that make systems obsolete soon after they are introduced. Continually redeveloping systems to take advantage of new technologies is an irresponsible waste of resources because it comes at the expense of other systems. Information systems are devel-

oped because they add value to the organization. Unlike Web sites, whose goal is often marketing, information systems continue to add value whether they use obsolete technology or the most advanced techniques. The only reason to redevelop systems is if new technologies offer a clear business value (see the Kambil and Ginsburg article in this section).

Users Aren't Like the Rest of Us.

The Web is dominated by computer enthusiasts. They—we—love technology and the exciting things it enables us to do. Most users aren't like us. They don't like change and don't care that a system uses the Web, any more than they cared that its predecessor used C. While most users were quite enthusiastic about our Web groupware system, some weren't. Those users had the same expectations for the Web system as they did for their Windows systems. They expected the standard interface concepts such as double-clicking to open topics and dragging-and-dropping to move items. The fact that

the system used the Web was unimportant compared to the lack of standard Windows features. Interface design is important, as discussed by Lohse and Spiller in this section.

Training is Critical. There is often an expectation that users will learn to use Web information systems the same way they learned to use the Web: by trial and error. Most users of our system learned in this way. However, this haphazard approach to training was a key factor in the failure of the system in several organizations (see Maurer sidebar). The most successful organizations that adopted our system provided training to all users through a series of classes or one-on-one sessions either face-to-face or over the telephone. The training focused on both the technology (that is, which buttons did what) and the new work processes needed. They provided specific written guidelines for use, gave examples, and discussed how the user's work habits could change. While we have no

data, name space management, overwrite protection and version management. While this group is defining a new architecture for next-generation HTTP servers, we had to integrate a variety of tools to address our immediate practical needs to provide these capabilities.

Lessons Learned

Marriage Counseling

Although the Web has simplified information delivery, WIS development is equally, if not more, challenging than traditional IS development. IS managers and users generally do not recognize this fact. Man-

agers and developers can easily be misled by tools that convey the impression that looking cool is of sole importance for Web pages. These tools, often in beta mode, have lured people away from recognizing the need for a systematic design approach. (A similar view is expressed by Dennis in this sidebar.) The real challenge is in managing information on a regular basis with updated content and links. Without enough planning and design, organizations can quickly walk into document maintenance nightmares.

WIS development involves a radical shift in structuring and presenting information. Enabling marketing departments to create content immediately ready

for intranet publication involved training authors and product managers in many new skills. Authors are now much closer to the creation and delivery of multimedia and hyperlinked content, previously the responsibility of art departments and publishers. This changes not only authors' job functions in terms of new processes and tools, but also their thought processes in terms of structuring and presenting information.

Authoring tools must be enhanced to support more error-proof templates similar to the ones popular word processing programs provide. Link management should become an integral part of the authoring environment as suggested by Carr et al. [4]. There should be instantaneous recording and verification of links in a link database. In addition to ensuring 100% link integrity, this would also eliminate the need for manual intervention, facilitate visualization of the information space, enable link change notifications, and provide link traversal privileges. (See Takahashi and Oinas-Kukkonen in this section.)

Rein et al. [9] state that Web technologies must support existing work processes just as work practices need to adapt to emerging Web technologies. We believe both sets of changes are required to fully enable collaboration among distributed groups. Document management

direct evidence, we believe that this training, particularly on the new work processes required, is an important factor in successful adoption. While this may seem obvious, we did not provide adequate training when we assisted two organizations that ultimately failed in the adoption of the system.

Web Information Systems are Systems, not Pages. WISs are information systems first, and Web systems second. Unlike Web pages that are designed to be browsed like magazines, WISs enable users to perform work—work that is inherently more complex than the rest of the Web. Understanding requirements is more difficult for Web information systems than for traditional Web development [4] because users interact with the system much more than with simple Web pages. Unlike Web page development, in which users play small roles, user participation is as critical to Web system development as it is for traditional system development. WISs should use the same disci-

plined system development principles, hard-nosed business-value assessments, and user-centered approaches that are required to build successful non-WISs [1, 5]. **C**

REFERENCES

1. Byrd, T.A., Cossick, K.L., and Zmud, R.W. A synthesis of research on requirements analysis and knowledge acquisition techniques. *MIS Q.* 16 (1992), 117–138.
2. Dennis, A.R., Poothari, S.K., and Natarajan, V. TCBWorks: A Web-groupware system for telecooperation. In F. Lehner and S. Dustdar Eds., *Telecooperation in Organizations*, Springer Verlag, Berlin, 1997.
3. Dennis, A.R., Poothari, S.K., and Natarajan, V.L. Lessons from the early adopters of Web-groupware. *J. Manag. Info. Syst.* To be published.
4. Dennis A.R. and Wheeler, B.C. Groupware and the Internet: Charting a new world. In *Proceedings of the 30th Hawaii International Conference on System Sciences*, IV, 287–296, 1997.
5. Holzblatt, K. and Beyer H.R. Requirements gathering: The human factor. *Commun. ACM* 8, 38 (May 1995), 30–32.

ALAN DENNIS (adennis@uga.edu) is an associate professor in the Terry College of Business at The University of Georgia.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

technologies with workflow facilities, combined with Web technologies, can reengineer complex and time-consuming authoring and publishing phases. We have seen resistance to change existing work practices either due to organizational culture or due to costs of a common technology platform across various departments. Developing a WIS is not just a technological issue; it is also an organizational, political, and cultural issue. Despite all our systematic efforts, our initial perception is the system is now less-than-successful due to non-technical factors [9].

We believe the best approach to address a majority of the issues we've mentioned is to seamlessly integrate document management and hypermedia functionality into the existing Web infrastructure. We need standards to avoid the myriad of tools and techniques being proposed almost daily by software vendors. This proliferation creates an interoperability nightmare for

application developers and end users, similar to those we experienced during the client/server era of the late 1980s and early 1990s. Product-differentiation wars between vendors could undermine the very distributed, platform independent, interoperable nature of the Web, further causing WIS design, construction, and maintenance problems. We urge designers and developers to remember this when adding new features or enhancing existing infrastructures. Just as the Web needs ubiquitous hypermedia support [2], it also needs ubiquitous document management support to design and construct large-scale, distributed, industrial-strength, platform-independent WISs.

Returning to the story of Alice and the Mad Hatter, we see the strengths of one compensate the weaknesses of the other, just like any marriage. Alice will keep the tea party in order so that the Mad Hatter can have his fun. ■

Hermann Maurer

Modern WISs

Neither virtual reality nor push technology nor any other fancy high-tech stuff will change Web life. The revolutionizing features will be easier information location and tools that allow a high degree of personalization. These two elements will provide breakthroughs in Web-based training (WBT) and knowledge management.

We all agree that we want good search facilities in our WIS. Moreover, our lives are primarily structured hierarchically. Aren't we happy when we go to a library to find medical books in one spot and physics books in another? Truth is, sometimes it would be nice to have a book in two spots (such as the biochemistry book in both the chemistry and biology sections). At other times we would like to have the books arranged by author or by title.

Thus, hierarchical approaches in a WIS must allow overlapping multiple classifications. Further, ideas from relational databases are entering the Web via metadata: documents should have standard attributes associated with them like author, creation date, date of last modification, expiration date (to end obsolete information!), access rights, and all other kinds of information useful to find information, to sort documents on, to restrict access to certain user groups, and so on.

In a WIS we surely want links. But we need links that can also be followed backward, that are stored in a separate database, and that can have attributes. After all, if you have a document you may want to see all documents pointing to it; and if you erase a document

then you want automatic deactivation of links in documents pointing to the deleted one to avoid broken links. For example, links should have access rights: you can have and make your private links (or links for your group of coworkers) that are visible to no one else. And if you can also introduce your own hierarchical structure on top of documents, the Webmaster is no longer a dictator. When you enter the WIS and log in, you see your view of the information, structured how you like it, and you can rearrange or make additional links at your pleasure.

The features mentioned help with the retrieval of information and allow the user to customize information. The same documents can be grouped and linked together in completely different fashions for different

REFERENCES

1. Balasubramanian, V., Bashian, A., and Porcher, D. A large-scale hypermedia application using document management and Web technologies. In *Proceedings for Hypertext '97*.
2. Bieber, M., and Vitali, F. Toward support for hypermedia on the World Wide Web. *IEEE Computer* 30, 1 (Jan. 1997), 62–70.
3. Bieber, M., Vitali, F., Ashman, H., Balasubramanian, V., and Oinas-Kukkonen, H. Fourth generation hypermedia: Some missing links for the World Wide Web. *Intern. J. Human-Computer Studies* (July 1997), 31–65.
4. Carr, L., Davis, H., De Roure, D., Hall, W., and Hill, G. Open information services. In *Proceedings of the Fifth World Wide Web Conference*. (1996); diana.ecs.soton.ac.uk/~lac/WWW96/Overview.html.
5. Carroll, L. *Alice's Adventures in Wonderland, The Complete Works of Lewis Carroll*. Oct. 1995, Gramercy Press, New York, N.Y.
6. Hopper, S., Hambrose, H., and Kanevsky, P. Real world design in the corporate environment: Designing an interface for the technically challenged. In *Proceedings for CHI '96*.
7. Isakowitz, T., Stohr, E., and Balasubramanian, P. RMM: A methodology for structuring hypermedia design. *Commun. ACM* 38, 8 (Aug. 1995), 34–44.
8. Malcolm, K. C., Poltrock, S. E., and Schuler, D. Industrial strength hypermedia: Requirements for a large engineering enterprise. In *Proceedings for Hypertext '91*.
9. Rein, G. L., McCue, D. L., and Slein, J. A. A case for document management functions on the Web. *Commun. ACM* 40, 9 (Sept. 1997), 81–89.
10. Schwabe, D., Rossi, G., and Barbosa, S. D. J. Systematic hypermedia application design with OOHDM. In *Proceedings for Hypertext '96*.
11. Whitehead, E. J. World Wide Web distributed authoring and versioning (WEBDAV): An introduction. *StandardView*. (1997) ACM, New York, N.Y.

V. BALASUBRAMANIAN (bala@e-papyrus.com) is a research-oriented practitioner. He heads E-Papyrus, Inc., an emerging technologies consulting firm in Edison, N.J., specializing in hypermedia design, document management, workflow, and user interface design.

ALF BASHIAN (abashian@njaost.ml.com) is a Systems Architect and Assistant Vice President at Merrill Lynch, Princeton, N.J. His professional work over the last 15 years includes developing client/server applications and Web-based systems.

Documentum is a registered trademark of Documentum, Inc.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

© 1998 ACM 0002-0782/98/0700 \$5.00

user groups without any danger of mutual interference.

Those ideas have been around for a while. However, the first industrial strength system that does all of this and more is Hyperwave (www.hyperwave.com).

A WIS with all these features provides the ideal basis for WBT, one of the key applications of the future. Modern WBT systems will have high-quality multimedia material with digital background libraries so users can add notes, start and participate in group discussions, and ask questions to tutors. The dialogs are recorded so that subsequent groups of learners may have no need to ask a question. They may find their question has already been answered some time ago. All major organizations will use such WBT systems sooner rather

than later. (A set of minimal features is described in the GEneral Training and Learning Environment (GENTLE); www.iicm.edu/gentle.htm).

Another breakthrough goes beyond GENTLE. WISs will be used more often for knowledge management. Have you ever heard this sigh from the CEO of a big company?: "If my employees only knew what my employees know." In any large organization very little of the valuable knowledge housed in the brains of employees is accessible to others. Thus, it is imperative to capture and store the knowledge of people in a group for the benefit of others. The WIS of the future will go a long way in this direction: it stores versions of documents, it provides recordable synchronous and asynchronous communication, it uses information

from quality control processes, and it helps with digital libraries appropriate for the organization. Such background libraries are also a solid platform for WBT material. The gap between the total knowledge in an organization and the knowledge accessible to individuals will become significantly smaller provided the appropriate modern WIS is used and the necessary organizational steps are also taken (see Turoff and Hiltz and Dennis). **C**

HERMANN MAURER (hmaurer@iicm.edu) is head of the Institute for Information Processing and Computer Supported Media (IICM), at Graz University of Technology, in Graz, Austria.

© 1998 ACM 0002-0782/98/0700 \$5.00