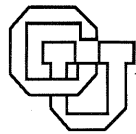


Personalizing the Web

Gerry Stahl

CU-CS-836-97



University of Colorado at Boulder

DEPARTMENT OF COMPUTER SCIENCE

ANY OPINIONS, FINDINGS, AND CONCLUSIONS OR RECOMMENDATIONS EXPRESSED IN THIS PUBLICATION ARE THOSE OF THE AUTHOR(S) AND DO NOT NECESSARILY REFLECT THE VIEWS OF THE AGENCIES NAMED IN THE ACKNOWLEDGMENTS SECTION.

Personalizing the Web

by Gerry Stahl

CU-CS-836-97

February 1997

**Department of Computer Science
University of Colorado at Boulder
Campus Box 430
Boulder, Colorado 80309-0430
(303) 492-3912**

For further information or comments, contact the author:

Gerry Stahl
3900 Pebble Beach Drive
Niwot, CO 80503
(303) 444-2792
Gerry.Stahl@colorado.edu

This Technical Report is ©1997 by Gerry Stahl

PERSONALIZING THE WEB

GERRY STAHL

Center for LifeLong Learning and Design
Department of Computer Science
& Institute of Cognitive Science
University of Colorado
Boulder, Colorado 80309-0430 USA
Gerry.Stahl@colorado.edu
<http://www.colorado.edu/~gerry>

ABSTRACT

The World Wide Web claims to address information needs of the whole world and therefore encourages knowledge to be expressed in universal formats. However, information is most meaningful and useful to people when it is adapted to their interests, backgrounds and situations. This paper presents a set of three mechanisms for adapting decontextualized information to personal needs: dynamic hypertext, personal perspectives and structural navigation. Five prototype systems illustrate how these mechanisms can be used in a variety of applications. Web servers incorporating this approach make the web more interactive, personal and adaptable.

THE NEED FOR A PERSONAL INTERFACE TO THE WEB

Excitement about the World Wide Web (the web) must be tempered by recognition of its unresolved contradictions. The new communication medium that strives to meet the future needs of the whole world fails to meet the needs of most people today. Security and censorship issues debated in the press and across the Internet point to conflicts concerning who should see what content in a medium designed to deliver everything to everyone.

If the medium is truly the message then we need to reflect on format as well as content. Should the web's world be one of homogeneous communication or should the world evolve into an arena where each person's individuality can shine forth and illuminate the perspectives of others? Ironically, the web's hegemonous technology may hold the key to a truly multi-personal world.

For information to be shared with everyone everywhere it must be represented in a highly decontextualized format, universally applicable. However, we know that information is the most interesting and useful when it corresponds in presentation style as closely as possible to our personal interests, conceptualizations, tasks and situations. When we are working on some problem, we want information that is directly applicable; when we are learning, we want information that matches our way of looking at things so that we can construct new understandings in our own ways.

The web poses an essential problem for computer-human interaction, not just superficial issues of user-friendliness. The computer wants computational (symbolic, universal) representations; people want personalized presentations. Web browsers and servers -- the software that mediates between data stored on the web and what web surfers see -- need to resolve this conflict.

Conventional web access lets users customize only the frills, like colors of link indicators. This is because the web is based on a simplistic model of hypertext, embedding link information in blocks of text where it

cannot be manipulated by readers. This imposed simplicity may have been necessary to spark the explosive growth of the web, but it is quickly becoming a barrier to achieving the real potential. Just as technical issues of name space and bandwidth, for instance, limit the quantity of users, the web's hypertext structure limits the quality of interactions.

If the web is to become more interactive so that people can actively communicate as well as passively consume and more personal so that it serves the needs of readers as well as the agendas of publishers, then a new generation of web servers is needed. The model of hypertext underlying the web has to be extended. Furthermore users have to be shielded from any consequent increase in complexity. The kind of hypertext content envisioned in this paper requires very thoughtful and tedious structuring. In many cases -- like the applications discussed in this paper -- such an effort on the part of content authors may be warranted. However, it is always important to avoid overburdening the reader.

The following sections first describe a set of three mechanisms to support the personalization of universalized information spaces: *dynamic hypertext*, *personal perspectives* and *structural navigation*. These mechanisms work together to make the web more interactive, personal and adaptable. Then applications of personalizable hypertext are illustrated in a series of five prototypes. These systems include knowledge-based software environments for designers, teachers, students, workers and network managers. They show how the three basic mechanisms can be used to support a variety of human-computer interactions to recontextualize information stored in digital repositories.

MECHANISM 1: INTERACTIVE WEBS

Comparisons of the web with other models of hypertext show several limitations of embedding links in static pages of text with HTML tags [2]:

- Readers cannot add annotations easily because they cannot modify the text in which they would have to add their comments or embed links to their pages.
- The links only point out from the text, so there is no way to avoid dangling references when target pages change.
- Contents cannot be restructured for different purposes because they are fixed pages -- the linearity of large texts has only been broken down to the page level.

Web pages should be dynamic. Web servers should adapt displays of data to users. This is accomplished by storing the content and its links in a database on the web server. When a browser requests a page, the web server assembles the page in real-time and sends that instead of sending the contents of a fixed file. The contents can be stored at a finer granularity (sentence or paragraph, rather than page), allowing selectivity in choice of content and linking, depending on the nature of the request to the server. This solves the problems that plague fixed web pages:

- Readers can annotate existing pages by adding supplementary contents or links to the database, to be included when the page is recreated by the server. (This process can be hidden behind natural seeming interactions and restricted under program control within the forms which allow authoring by readers.)
- Since the database stores references to both ends of links, the server can check that references are valid and that their inclusion is relevant and authorized for a given request. (The server can also check for links to a page before deleting that page to prevent problems in advance.)
- Most importantly (from this paper's perspective), the server can assemble and display contents based on who wants to see what how.

The importance of dynamic web pages is widely recognized and now increasingly supported by server software and tools like Hyper-G, Frontier, Cold Fusion, HyperNews, SuperNova, Tango and IntraBuilder.

MECHANISM 2: PERSONAL PERSPECTIVES

Given dynamic web pages, perspectives are a way of determining what is displayed for whom [3, 4, 6, 15]. The mechanism of personal perspectives builds on the dynamic assembly of pages. Atoms of content (e.g., brief paragraphs or multimedia items) are assigned to primitive perspectives (such as "intermediate level electrical issues for residential house construction") and this designation is stored with those content nodes. Then a hierarchy of useful perspectives is built up by grouping primitive perspectives into higher-order perspectives. This is all done as part of the large initial effort of seeding an information space with hypertext content organized into perspectives.

Individual information seekers can, in turn, define one or more personal perspectives reflecting their own interests. For instance, an architect might construct several distinct perspectives in order to view designs alternatively by "intermediate structural concerns," "advanced aesthetic issues," etc. Or a company can define a hierarchy of perspectives for its employees, allowing each to view corporate documents that are appropriate to their status and roles in the organization, combining the perspectives of a work group in the perspective of the group's manager. Authorization of read, annotate and modify permissions can be defined through perspectives, facilitating a fine-grained security system without cumbersome passwords.

As people annotate and edit contents, the information space evolves in a controlled way. The perspectives mechanism incorporates the advantages of contexts [9], virtual copying [10] or transclusion [12]:

- Changes to contents only show up in the perspectives in which they were made or in perspectives that inherit from them; original contents are always available. Perspectives can therefore serve as a versioning and archiving system.
- The mechanism conserves memory space because only the brief elements actually modified have to be duplicated for different versions, not whole pages.
- If charges are levied on copywritten materials, the necessary information for this can be associated with elemental contents and preserved even in highly annotated or edited multi-source documents.

The user's choice of perspective controls the adaptation of all information to the user. When a dynamic web page is assembled, only the versions of contents associated with the active perspective are displayed.

MECHANISM 3: NAVIGATING THE NETWORK ADAPTIVELY

It is possible for a content provider to create a nuanced information space in which detailed contents are organized by perspectives and inter-linked using a rich semantics. Given such an information space, a structure-based query language can provide power and generality in navigating this space to generate an unlimited variety of web presentations adapted to different needs [11, 16]. For instance, in a design rationale repository one could display on a page all the issues added in the past year that have three or more answers with no justifications stated.

In earlier systems developed by the author and collaborators (see [Acknowledgments](#)) we tried to incorporate common syntactic structures from English into our navigation language to make it readable [17]. It was like a combination of SQL and HyperCard's HyperTalk. The sample query of the previous paragraph can be expressed like this in that language: "issues that have creation date after 1/1/96 and have more than 2 answers that have no justifications." In our future systems we want to embed the language in a visual programming interface (Visual AgentTalk [14]) to allow users to construct and test queries using direct manipulation from graphical representations of syntax and content options.

A structural navigation language that defines patterns of link traversal can be used internally to the web server software to assemble pages. It can also be used within the information space, e.g., to define virtual

links that point to contents selected dynamically. Such a language can be made arbitrarily powerful. The language we implemented includes commands to create and manipulate lists of contents, to traverse combinations of associative links, to filter out various contents, to quantify and to perform arithmetic or logical computations.

While power-users of applications for professionals might want access to the full power of a hypertext navigation language to explore information spaces and to format reports, other users might be thankful to have the language formulations hidden from them. In some of our systems, for instance, we have encapsulated useful standard queries behind buttons in the user interface or in special structured displays, as illustrated in the following sections.

FIVE PROTOTYPE SYSTEMS

We have designed five applications that illustrate different approaches to and uses of personalizable information systems. None of these systems has yet been deployed to users; they are research prototypes for exploring alternative implementations and applications of the three mechanisms. The applications target different user communities as shown in Table 1. They emphasize different approaches and design issues.

		target users	system approach	focus for paper
1	Hermes	NASA designers	DODE substrate	assemble rationale
2	TCA	school teachers	digital library	curriculum planner
3	PLM	physics learners	personal medium	custom display
4	CIE	corporate teams	collaboration	merge versions
5	WebNet	LAN managers	evolving info space	information medium

Table 1. Five applications of personalized information delivery.

1. The three mechanisms described above were first developed for the Hermes system [16]. Hermes is a substrate for building domain-oriented design environments (DODEs) like Janus [7] and Phidias [9], desktop applications for designers to do their work with knowledge-based computational support. A design environment for lunar habitat designers at NASA was built on top of Hermes. It is still being developed to maintain design rationale at NASA. In particular, the perspectives mechanism is being exploited to display multiple versions of manuals using shared hypertext contents.

2. The Teacher's Curriculum Assistant (TCA) was designed as a web browser for digital libraries of educational resources and curriculum [19]. TCA's interface was prototyped to demonstrate how fine-grained hypertext stored in an indexed database could be assembled into curriculum tailored to the needs of classrooms [18]. We plan to use TCA for sharing over the web educational games developed by students using WebQuest [13].

3. A Personalized Learner's Medium (PLM) was originally conceived for texts on high energy particle physics. The three mechanisms were combined with additional structuring techniques behind the scenes to present technical reading materials consistently at different levels of difficulty for different students. This application illustrates the generality and synergy of multiple customization mechanisms.

4. The Collaborative Information Environment (CIE) was prototyped to help workgroups develop and maintain documents for ISO 9000 certification. It uses the three mechanisms to organize versions for

individuals and groups and to enforce read, annotate and write permissions. This business application includes practical tools for comparing and merging information from several perspectives.

5. WebNet is our current effort to provide rich information sources on the web for LAN managers. With it, we are starting to explore ways to use the three mechanisms and other techniques to support fluid evolution of information by communities of practice that need up-to-date information. We want to allow information authored by users to be automatically structured for personalization by others.

The following sections review these five prototypes to illustrate how the mechanisms discussed in this paper can help to personalize the delivery of decontextualized information.

SYSTEM 1. HERMES: TAILORING TO THE TASK

The theory motivating *Hermes* is based on hermeneutic philosophy, design methodology and observation of NASA lunar habitat designers [15]. It notes that professionals engaged in design tasks are continually interpreting their work while tacitly *situated* with the design artifact (e.g., a CAD drawing), viewing their problem from some particular personal *perspective* (e.g., privacy concerns), and articulating their interpretation in explicit *language* (e.g., a query about design rationale). The core technology is intended to support design by:

- providing a dynamic version of hypertext for representing the design *situation*,
- structuring the knowledge base into personal *perspectives* to provide alternative interpretive viewpoints, and
- providing a computational *language* for navigating through and manipulating the represented artifacts, domain knowledge and design rationale.

Design rationale is accessible in *Hermes* as context-sensitive hypertext -- but in a different sense from the usual help systems. That is, design knowledge is adapted to the current design task undertaken in the application's workspace. Typically, a designer is alerted to the need to review design rationale by an automated critic rule being triggered by the state of the design as it is developed in the CAD (Computer-Aided Design) component of the *Hermes* design environment. The CAD graphics system is itself constructed with hypertext nodes and links so that the navigation language interpreter can analyze it. Critic agents are computational rules defined in the language; they check the designs and provide suggestions for improvement by retrieving relevant domain knowledge and rationale. Information for delivery to the working designer is assembled from fine-grained textual elements linked together in accordance with the argumentation structure (e.g., design issues are linked to their alternative answers and subissues) [17].

The designer can navigate through the rationale using an interface that in effect builds queries whose results are assembled, formatted and displayed. There is always a personal perspective that is active; it determines what contents are filtered out and which are displayed. In *Hermes*, perspectives are used to differentiate different professional domains (life support; privacy concerns; noise and vibration; micro-gravity; electrical; etc.).

Note that the design rationale presented in the center of Figure 1 is displayed in a custom outline format, indenting alternative answers and their rationale under the issues. The dynamic approach to all displays of content from the hypertext database allows different materials to be arranged differently. The dialog to the left allows the designer to navigate links that have selected semantics -- implicitly using the language without having to program it explicitly. These are two examples of how mechanisms (dynamic hypertext, structural navigation) can be hidden behind displays or controls that are natural and appropriate to a domain (e.g., design rationale).

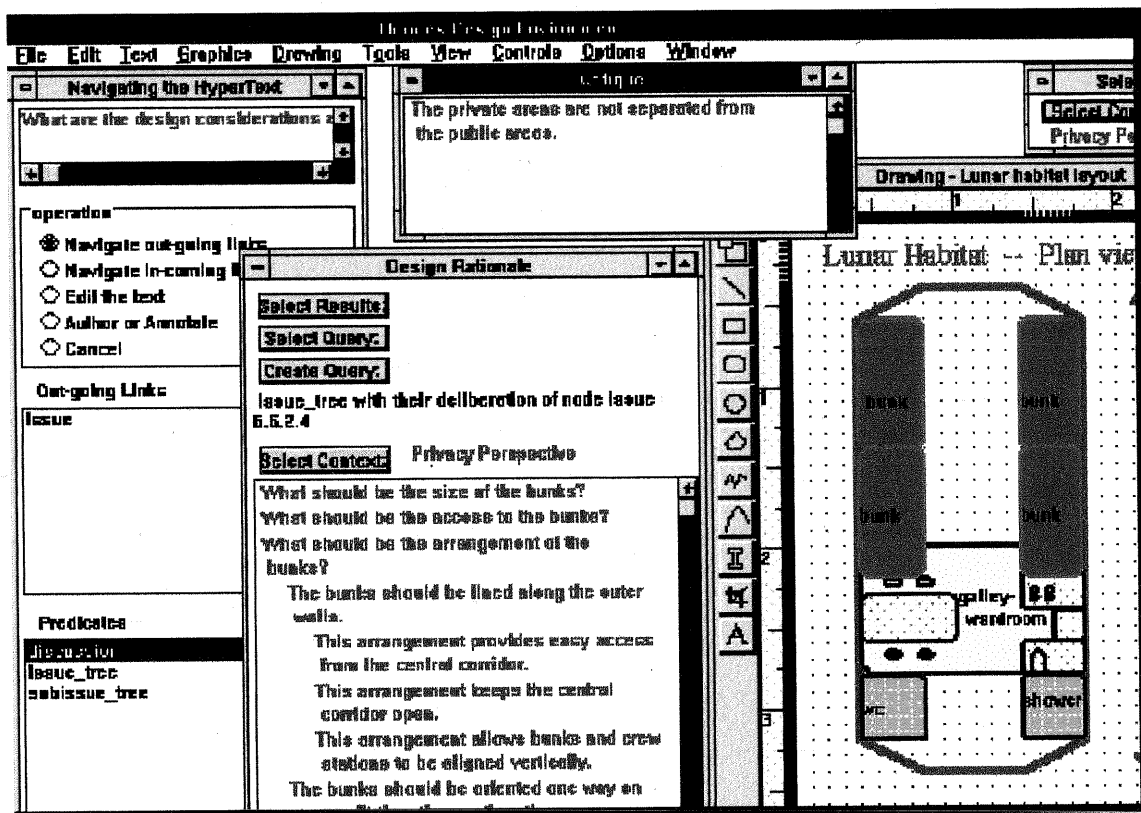


Figure 1. Hermes. From left to right the windows illustrate navigating the hypertext by choosing from a list of link types; a presentation of design rationale dynamically constructed from related texts by a query; an automated critique of the current design; the CAD workspace; a button for selecting a new perspective.

SYSTEM 2. TCA: CUSTOMIZING FOR THE CLASSROOM

The Teacher's Curriculum Assistant (TCA) was designed as a web browser for teachers to access digital libraries of constructivist curriculum. When teachers try to use browsers like Netscape Navigator to take advantage of the educational ideas that are beginning to be posted to the web, they meet with the following problems:

- There are no effective methods for locating relevant curriculum sites.
- It is too hard to search for items of interest.
- There is no choice of versions to select for different situations.
- There are no tools for adapting what is found to local needs.
- There is no support for organizing scattered ideas into workable curriculum.
- There are no ways for teachers to share their experiences widely.

These problems can be overcome with centralized repositories of carefully structured curriculum and indexed resources. The repositories should support two-way communication, so that teachers can share their experiences using materials in the repositories and can "grow" the repositories.

TCA's Profiler, Explorer and Versions components work together for information retrieval. The Profiler helps teachers define perspectives for a particular classroom and locates curriculum and resources that match the perspective. The Profiler prompts a teacher to build a simple user model of curricular interests, classroom resources and pedagogical preferences of the teacher and students.

The Explorer displays relevant items and allows the teacher to search through them to find related items. The Explorer uses an interface similar to the Mac Finder or Win95 Explorer to navigate across links between a curriculum, its weekly units, their lesson plans and their individual resources.

A Versions dialog then helps the teacher select from alternative versions that have been adapted by other teachers. Through these interfaces, teachers can locate the available materials that most closely match their personal needs; this makes it easier to customize the materials to classroom requirements.

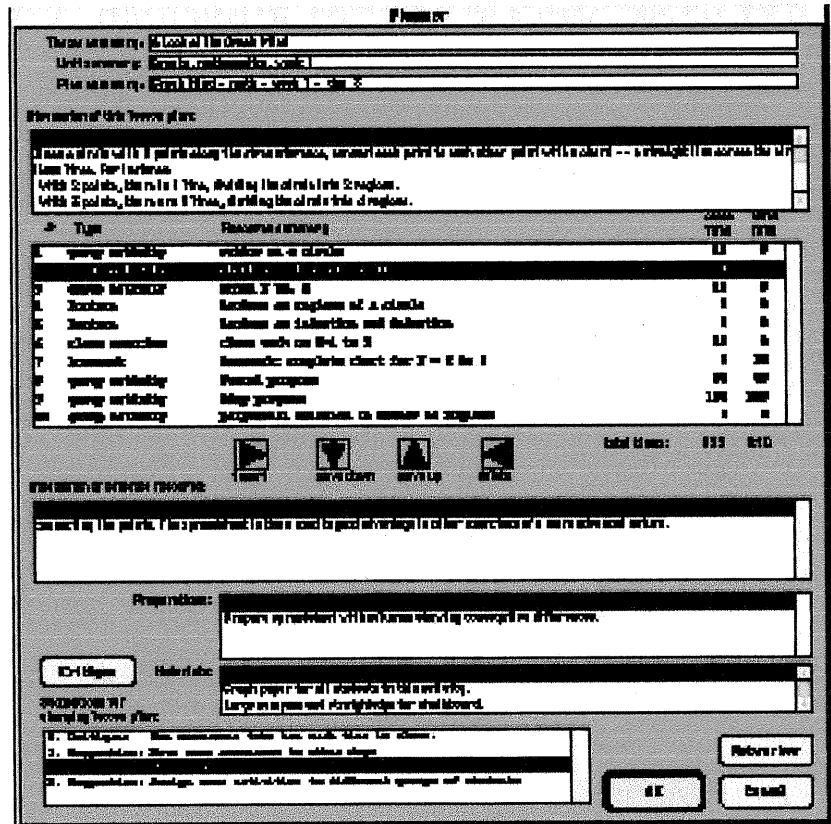


Figure 2. The TCA interface for the Planner.

TCA's Planner, Editor and Networker components help the teacher to prepare resources and curriculum for use and to share the results of classroom successes. The Planner (Figure 2) is a design environment for reusing and reorganizing lesson plans. Note that the Planner interface displays a variety of linked information in a useful format for teachers to study and manipulate. All the educational resources linked to a lesson plan are listed together where they can be rearranged or selectively deleted; their recommended classroom and homework times are added up as a guide; lists of required materials, prerequisites and preparation steps are compiled. In addition, a critiquing component displays suggestions and warnings within the plan. Here, relevant inter-related information is gathered together in a display that can print out the teacher's familiar lesson plan.

The Editor allows the teacher to modify and adapt individual resources (texts, spreadsheets, drawings, quizzes, collections of web pointers). This is a primary means for personalizing the curriculum. Finally, the Networker supports interactions with the Internet, providing a two-way medium of communication with a global community of teachers. Using the Networker, a teacher can share successfully customized versions

of standard curriculum with other teachers who might have similar needs. Contributed versions are described and indexed so that they can be retrieved when relevant to future requests.

SYSTEM 3. PLM: READING FOR READERS

The Personalized Learner's Medium (PLM) illustrates how several mechanisms can be used synergistically to transform generalized information in an educational digital library on the web into personalized information in a student's browser. The PLM technical approach integrates customization ideas, mechanisms and industry standards from TCA, Hermes and HTML. This approach can be applied to classroom applications or to learning-on-demand features in software for professionals, where information relevant to the current task is tailored to the person and to the task.

PLM was originally proposed for people learning high energy particle physics. This is a field where students have to go over the same material several times, at different levels of mathematical sophistication. There is little continuity in the available textbooks. To start, there are a number of popular books with no mathematics. Then there are some requiring substantial background in mathematical physics. Next come highly technical texts and then the very specialized notation of the journals. The problem is that students come with different backgrounds and the texts use an assortment of seemingly incompatible formalisms.

The idea behind PLM is to provide one consistent and comprehensive hypertext source with paths through it at different levels. By specifying a perspective, a reader determines the level of material to be presented, including the desired amount of historical background, of mathematical elaboration or of experimental description. The hypertext format allows a reader to explore related material to supplement the perspectival path -- e.g., to get a refresher on some forgotten mathematical formalism.

This approach uses computational power to go beyond the limits of static printed books. Of course, it requires substantial effort to plan and structure material to be applicable for a broad range of reading levels. However, a textbook publisher might find this extra effort worthwhile if it saves printing, updating, maintaining and marketing a whole sequence of distinct volumes. Similarly, a curriculum developer creating an educational digital library might judge such an approach more worthwhile than just putting a collection of redundant, inconsistent and soon-out-of-date books on-line. A major advantage of the integrated hypertext database is that updates to knowledge permeate all views consistently.

Personalization in PLM takes place in eight sequential stages within the process of selecting materials from the digital library, analyzing them into hypertext nodes and links, and synthesizing selected contents into a personalized display. The stages typically take place automatically, but the user can intervene when desired.

Stage 1. Searching for relevant materials. The learner defines a Profile of the materials sought in an educational digital library. This profile includes characteristics of the learner as well. The profile is used by PLM to formulate a query that retrieves a selection of materials from the library. The profile functions as a user model for the software, but one that is under the control of the user because the user can define and revise the model at will.

Stage 2. Browsing among related resources. Descriptions of selected materials are displayed in an Explorer window. The learner uses this interface to browse among related library resources, such as textual selections, software tools, historical background, useful mathematical techniques, relevant video clips. By providing for browsing within the confines of the Profile search, PLM gives the learner freedom to explore with limited danger of becoming lost on irrelevant and inappropriate tangents.

Stage 3. Selecting the best fit version. The digital library may include multiple versions of a given resource. For instance, a physics lesson might be approached using a graphical simulation, equations or computer programming. The learner can select the most appealing approach or use different versions at different times.

Stage 4. Parsing into nodes and links. Documents are broken down into their elements (as hypertext nodes), connected by typed hypertext links. The link types are based on the element's HTML markup type (e.g., `title`). This is done automatically by PLM. Custom node and link types may be defined by learners -- or by their teachers. Here, the structuring that already exists in well-organized on-line documents is exploited.

Stage 5. Viewing from a perspective. The learner's profile defines a perspective. The currently active perspective selects which nodes and links can be viewed. This allows multiple, redundant forms of information to be present in resources in the library, of which all but one form will be filtered out. For instance, many people may have annotated a particular resource, but a learner may want to filter out all annotations except her own, her teacher's and her classmates'. Then she would define her own perspective and have it incorporate her teacher's and her class' in order to view what they view in their perspectives.

Stage 6. Querying with the language. All PLM displays are created dynamically by queries in the hypermedia navigation language. Statements in the language in effect specify starting nodes, types of links to traverse and characteristics of nodes to filter out. Execution of a statement takes place within a selected perspective and results in a collection of linked nodes. This collection is the material selected for display. The language can be extended by end-users and terms in the language can have different meanings in different perspectives. Readers, their classmates and their teachers can build up special queries to display materials of interest.

Stage 7. Synthesizing a display. PLM constructs a document from the collection of nodes and links. The document is marked up using a version of HTML. At this stage, the information retrieved from the library has been personalized.

Stage 8. Formatting the presentation. The final stage is to display the information. The display format can be personalized by adjusting the mark-up definitions. For instance, the hierarchical information can be indented by level. Alternatively, different levels could be italicized or text size and color adjusted to individual preferences.

Through these eight stages, standardized materials are selected and displayed in a way that can be tuned extensively by individual learners to their needs. A thoughtfully prepared hyper-document in the global library can be personalized differently by each learner in the world. Ironically, the key to *individuating* the presentations of the materials is to exploit the *standardized* HTML structuring of documents to break them into semantically meaningful fine-grained hypertext that can then be re-assembled by software to adapt to personal preferences. In this way, the technology of the global web can be used to personalize the web's contents.

SYSTEM 4. CIE: COORDINATING COLLABORATION

The Collaborative Information Environment (CIE) is a tool for workgroups to collaboratively formulate policy and procedure documents in accordance with the Total Quality Management organizational style and ISO 9000 documentation standards. This groupware tool facilitates the development, review, critiquing, annotation, editing, versioning and auditing of shared documents. It supports efficient group communication about documents and sophisticated version control of them.

Each group member using CIE has a personal notebook that allows document versions to be viewed, annotated and edited in ways appropriate to the person's position in the organizational workflow. The notebook typically opens to the person's calendar. The Calendar lists documents with deadlines coming up soon. The user can then go to an Overview of the company's documents to begin reviewing one to work on. The Overview may display a variety of information about documents, such as how many comments or versions currently exist of each document, so that a manager can determine immediately where concerns lie and which documents are in need of revision. The Calendar and Overview contents are, of course, computed dynamically from an underlying hypertext, based upon options selected by the user, the user's perspective, queries defined by the user or by the system, and the current information state.

The development of documents primarily involves people editing existing drafts and commenting on each other's proposed changes. CIE provides interfaces for viewing different versions by one's co-workers and annotations that have been made to them, as well as means for easily adding one's own ideas. A number of features are also available for manipulating one or more versions. For instance, the chair of a workgroup might merge multiple versions together to arrive at a consensus document for the group. A CIE editor helps the chair to merge versions by redlining terms deleted in each version and color-coding additions suggested in different versions. Alternatively, one person's version can be promoted to the group's perspective.

In CIE, perspectives enforce company security concerns and allow people to create and share personal versions of company documents without affecting other people's views of those documents. Permissions for reading, annotating and actually editing policies and procedures can be associated with individual perspectives. The development of CIE required substantial refinement of the perspective mechanism as used in the previous systems. Now, the perspectives enforce access permissions to shared corporate intranet information. Interfaces were needed to assign and administer perspectives. Also, intuitive mechanisms were needed for sharing, merging and exchanging information between perspectives.

Because perspectives inherit changes to documents from their ancestors, people can effectively be assigned perspectives that mirror their role in the organization hierarchy. Assume my perspective inherits from the organization's and from my workgroup's, while my supervisor's perspective also inherits mine. Then my view of a company policy reflects any changes adopted by the company or by my workgroup. Moreover, my supervisor can view changes I propose and compare them to those of others in our workgroup. In this way, everyone can try out different ways of wording documents and then recommend adopting certain specific changes. This facilitates the coordination of collaborative document maintenance, which can otherwise raise a serious barrier to ISO 9000 level documentation in an organization.

SYSTEM 5. WEBNET: ENABLING EVOLUTION

At the Center for LifeLong Learning and Design we are interested in knowledge-based systems to support working and learning in professional domains. In our current WebNet project we focus on supporting the *evolution* of domain-specific knowledge using web-based information repositories. Our work is increasingly taking place within the context of rapidly changing web technologies. Advanced web server software like Hyper-G [1] and HyperNews [8] as well as intranet front-ends to databases provide important functionality that needs to be integrated with our own mechanisms.

The view of the web as an information medium is often traced back to Vannevar Bush's vision [5]. At the close of the last world war, Bush emphasized the need to develop technology to support effective access to the burgeoning "record" of scientific knowledge. We are exploring the use of dynamic hypertext, personal perspectives, structural navigation and other approaches to achieve our extended vision of an evolving knowledge medium. One domain of knowledge we are trying to support is our own research "record": papers, slides, dissertations, proposals, workplans, email, glossaries, bibliographies, etc. that form an organizational memory for our research group.

Another target domain is that of Local Area Network (LAN) designers and managers. This domain involves supporting a virtual community of practice that relies on knowledge of rapidly changing technology. WebNet has to allow for evolution at many levels: growth of content, restructuring of content, multiplicities of versions, proliferation of retrieval queries, diversification of information displays, changes in the domain and emergent system needs.

Our strategy is to store all the information in WebNet in an integrated database. Data is interrelated via hardware devices, LAN sub-networks and users that are referenced in the data. Conventional information tools for LAN managers such as trouble queues, host tables, device diaries or trouble-shooting procedures exist as views onto this rich information space. Familiar input and report forms are created by a structural query language. Personal perspectives are used to filter through only information about devices, etc. that are of interest to particular individuals.

In WebNet, dynamic hypertext is implemented with recent intranet technologies including Frontier, Tango and IntraBuilder. These mechanisms serve to structure information entering the evolving information base (as well as displayed reports) so that new data can be processed by the mechanisms for personalization in the future. In addition, we try to implement the language and perspectives mechanisms in ways that maximize their extensibility, so that system users can create new categories as necessary to reorganize the data to meet emerging needs.

THE FUTURE

The web has not yet fulfilled Bush's vision; meanwhile the stakes have risen. We believe that:

- The web must be an *interactive* medium rather than the repository of a static record, so that knowledge in different domains will be "grown" by communities of practice interacting through the web, gradually overcoming the spotty coverage of information currently on the web.
- People using the web -- whether for information retrieval, knowledge evolution or collaborative communication -- must be empowered to control, adapt and extend the computational mechanisms of the web to serve their *personal* needs.
- Knowledge on the web must be *adapted* to the work situations and learning perspectives of individuals so that decontextualized, explicit information can be meaningfully integrated into people's largely tacit understandings.

The prototype systems described above illustrate how *dynamic hypertext* with *personal perspectives* and *structural navigation* can mediate between universal representations of knowledge and personalized presentations of relevant information. This approach has the potential to make the web significantly more interactive, personal and adaptable.

We are currently pursuing user-centered design efforts to fashion computer-human interactions that can shield author/readers from the complexities resulting from our approach:

- Authoring of comprehensive information spaces will require careful structuring of content nodes and their links to make information applicable to a wide range of applications; publishers of such information will need tools to support this work and to automate it where possible.
- We would like to automate some of the defining and refining of personal perspectives so that it happens behind the scenes, although we also want people to be able to knowingly review, modify and switch their computational perspectives.
- Finally, we want to expose the structure-based query language as an end-user scripting language with visual programming supports for power users who want to exceed the functionality of pre-programmed queries, while providing standard queries transparently for common tasks.

Although we do not believe that basic tools for professionals to use in exploring web resources will need to be walk-up-and-use in the future, we want to avoid cognitive burdens that distract from the immediate tasks of communicating, working and learning.

As the sources of our information become increasingly mediated by computers it becomes correspondingly important that people be able to maintain control over that manipulation when they want to. Web servers that make the web empowering for personal expression and authentic knowledge construction can prevent the reduction of the web to a world wide digital mall of mass-produced information commodities.

ACKNOWLEDGMENTS

The ideas and systems reported here came from collaborations with Ray McCall, Gerhard Fischer, Bob Owen, Doug Swartz, Tammy Sumner and Jonathan Ostwald. Hermes, the author's dissertation system, grew out of research on the Phidias system at CU's College of Environmental Design with support from CASI. TCA was developed at Owen Research with support from NSF. PLM and CIE were designed at Personalizable Software, a consulting firm founded by the author. WebNet is a project of CU's Center for LifeLong Learning and Design under support from ARPA. Hermes Ver 2 is © 1994 by Gerry Stahl; CIE is © 1996 by Personalizable Software.

REFERENCES

1. Andrews, K., Kappe, F., Mauer, H. Hyper-G and Harmony: Towards the next generation of networked information technology. *Proc CHI '95 Companion*, (1995), 33-34.
2. Bieber, M., Vitali, F., Ashman, H., Balasubramanian, V., Oinas-Kukkonen, H. Fourth generation hypermedia: Some missing links for the World Wide Web. Submitted to *International Journal of Human-Computer Studies*. Special issue on HCI & the Web.
3. Bobrow, D., Goldstein, G. *An Experimental Description-Based Programming Environment: Four Reports*. Technical Report CSL-81-3. (1980). Xerox Palo Alto Research Center.
4. Boland, R., Maheshwari, A., Te'en, D., Schwartz, D., Tenkasi, R. Sharing Perspectives in Distributed Decision Making. *Proc CSCW '92*. (1992).
5. Bush, V. As we may think. *Atlantic Monthly*, 179, 1 (1945), 101-108. Reprinted in *Interactions*, III, 2 (1996), 35-46. Available as <http://www2.theAtlantic.com/atlantic/atlweb/flashbks/computer/tech.htm>
6. Fischer, G., Nakakoji, K., Ostwald, J., Stahl, G., Sumner, T. Embedding computer-based critics in the contexts of design. *Proc InterCHI '93*, (1993), 57-164.
7. Fischer, G., Nakakoji, K., Ostwald, J., Stahl, G., Sumner, T. Embedding critics in design environments. *The Knowledge Engineering Review*, 4, 8 (1993), 285-307.
8. LaLiberte, D. About HyperNews. Available as <http://union.ncsa.uiuc.edu/HyperNews/get/hypernews/about.html>
9. McCall, R., Bennett, P., d'Oronzio, P., Ostwald, J., Shipman, F., Wallace, N. Phidias: Integrating CAD graphics into dynamic hypertext. *Proc ECHT '90*, (1990), 152-165.
10. Mittal, S., Bobrow, D., Kahn, K. Virtual copies: At the boundary between classes and instances. *Proc OOPSLA '86*, (1986), 159-166.
11. Nardi, B. *A Small Matter of Programming*. (1993). MIT Press.
12. Nelson, T. H. The heart of connection: Hypermedia unified by transclusion. Available at <http://www.cs.uct.ac.za/Local/CS300W/HCI/nelson.htm>.
13. Perrone, C., Clark, D., Repenning, A. WebQuest: Substantiating education in Edutainment through interactive learning games. *Proc WWW5* (1996).
14. Repenning, A., Ambach, J. Tactile programming: A unified manipulation paradigm supporting program comprehension, composition and sharing. *Proc Visual Languages '96*. (1996).
15. Stahl, G. Supporting situated interpretation. *Proc Cognitive Science Society '93*, (1993), 965-970.
16. Stahl, G. *Interpretation in Design: The Problem of Tacit and Explicit Understanding in Computer Support of Cooperative Design*. Ph.D. dissertation. (1993). Department of Computer Science. University of Colorado at Boulder.

17. Stahl, G., McCall, R., Peper, G. Extending hypermedia with an inference language: An alternative to rule-based expert systems. *Proc IBM ITL Conference: Expert Systems*, (1992), 160-167.
18. Stahl, G., Sumner, T., Owen, R. Share globally, adapt locally: Software to create and distribute student-centered curriculum. *Computers and Education*, 24, 3 (1995), 237-246.
19. Stahl, G., Sumner, T., Repenning, A. Internet repositories for collaborative learning: Supporting both students and teachers. *Proc Computer Support for Collaborative Learning '95*, (1995), 321-328. '93