A Dynamic, Online Library for Historical Documents

Drew McDermott, David Gelernter, Jeffrey Westbrook
Lenor Zuck, Donald Waters

Research Report YALEU/DCS/RR-1047
September 1994

YALE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

A Dynamic, Online Library for Historical Documents

Drew McDermott
David Gelernter
Jeffery Westbrook
Lenore Zuck
Yale Computer Science Department

Donald Waters
Yale University Library

September 13, 1994

Abstract

Yale's Digital Library project is aimed at providing on-line hypertext-style access to a large database of actual documents, drawn from the domain of foreign-relations history. It is oriented around two main themes:

1. An important part of what historians do is retrieve primary documents related to past events, and weave them together into a structures that provide explanations for those events.
2. These documents can be stored in fault-tolerant, distributed databases that provide efficient access by a large number of clients.

The research on Theme 1 is oriented around providing views of document sets based on spatial, temporal, and associative dimensions. Professional historians should have the ability to build, share, and annotate the various contexts within which documents exist. Furthermore, they are already engaged in collecting and sharing documents, so a partnership between historians and computer scientists can lead to rapid development of systems to enhance this activity.

Theme 2 research will provide a stable and compact substrate for Theme 1. Documents will consist of abstract objects of various types, including images compressed using state-of-the-art techniques. These abstract objects will inhabit a "Persistent Associative Shared Object (PASO)" database, which generalizes and formalizes the concept of "tuple space," from the Linda family of languages developed at Yale.

The research will be evaluated in terms of its ability to impact on the activities of professional historians at Yale and elsewhere (including archivists at the U.S. State Department).

1 Introduction

Much of the research effort in history and political science is devoted to searching primary sources, both published and archival. An example of published material is the series *The Foreign Relations of the United States,* published by the U.S. Department of State, which reprints diplomatic memoranda for use by scholars. The published documents are culled from the archives, and often sanitized. The archives are indexed in a rough way, but there is such a huge volume of information that a large fraction of a scholar's time is devoted to searching through the archives for nuggets. This search involves not only the standard bibliographic tools — card catalogs, subject bibliographies, citation searching, and the like — but also the framing of research questions in ways that can make use of those tools.

Once the searcher has found some documents that might be relevant, he or she photocopies as much as possible, building up a private library of primary historical documents relevant to some thesis that he or she is pursuing. When the process is completed, a paper or book is written, whose bibliography refers to the sources by their archive index numbers.

A key insight is that researchers bring to their task a range of perspectives and background information that is not now captured in bibliographic tools, which simply do not deal with the complex of personal, organizational, and historical relationships that primary documents fit into. The library catalog model, the indexing and abstracting model, and the bibliography model all focus primary attention on the document as an independent entity that has only coincidental relationships to other documents, and whose relationships to the persons, organizations, and events through which it was created and has been used is superficial at best. Data elements that identify authorship, creation date, and publication responsibility are the limit to which current bibliographic tools aspire. Minimal indexing by standard subject terms provides some level of collation by predefined topics, but does not address the description of the provenance of the document that would provide the context within which the document could be identified or interpreted.

As a result, researchers must use the following methods:

1. translate their research interests into categories and terms that existing tools can address;
2. use their knowledge of the context within which documents have been created and maintained and of the documents' internal structures and arrangement to identify potentially relevant places to look for materials related to their interests; or
3. follow the paths and markers left by previous researchers in the field to identify relevant resources to explore.

The first of these methods, the traditional cataloging and indexing model, is already in place and the linking of these tools to digital images of documents is already being implemented in various systems, including the linking of Orbis, Yale's online catalog, to Project Open Book, the Yale Library's prototype imaging system.

The second method is particularly important for the identification and selection of primary source materials, especially those found in archival and manuscript repositories and museums. Archives, manuscripts, and museum artifacts do not arise from the same types of deliberate, targeted activity that produces published works. Rather, they tend to be the unconscious by- products of organizational or personal activities; primary sources are generally not created as ends in themselves (as published works often are), but as means to an end. As such, the researcher must have a firm appreciation of the organizational or personal context in order to locate relevant documents and interpret them faithfully. There have been attempts to make this sort of information explicit, but typically it is expressed as a series of narrative notes tied to a particular body of documentation. There is no structure to these notes and no guidelines for their construction. Given this situation, much of this information never makes it into these bibliographic tools, resulting in a high degree of reliance on human intermediaries — the reference and processing staff — who have accumulated this knowledge in the course of working with the collections and with researchers and who can translate the research request into the terms needed either to query the bibliographic tools, or, more likely, make the connections themselves from their accumulated knowledge.

The third method, following the trail left by previous workers, is crucial to current practice; it is the equivalent of "repeating experiments" in the study of history, and it often is the point of departure for fresh explorations. Researchers report on their data as if it were another world, through which they traveled using dogsleds and native guides. If someone else wants to revisit the data, they can either come see the private collection of photocopies, or recreate the expedition to the archives.

We propose that in the future this picture will be quite different. An increasing fraction of the primary data will be in machine-readable form. When a scholar explores a certain area, he or she will find indexing structures left by previous visitors. In fact, when appropriate two or more scholars can be made aware of each other's "presence" in a certain part of the information space, and carry on conversations. Software surrogates can be left behind to signal when new data or new people enter the area. And, most important, the users of the system can, under supervision, make changes to the shape of the information space. Over time, the information space will evolve to match more closely the concepts and relationships in the heads of the researchers.

1.1 Subject Focus

Yale has world-class programs that draw on both the historical and social science disciplines to study international relations. In support of these programs, the Library has built a major collection of both published governmental documents and unpublished manuscript and archival material documenting the development of United States foreign policy and the activities of the United Nations. In addition, Yale has recently received major foundation funding for a study of the history, organization, and future of the United Nations, timed to coincide with the 50th anniversary of its founding in 1995. Because of these strengths and the importance of introducing the digital library into the humanities and social sciences, Yale proposes to use the history of U.S. foreign policy as the subject focus of its proposed system.

There is no dearth of material that needs to be stored in an accessible form. As we mentioned above, the series *The Foreign Relations of the United States,* published by the U.S. Department of State [33], reprints diplomatic documents for use by scholars. It consists of some 300 volumes published continuously since 1861. The principal sources for the series are records of the President, national security advisers, and the State Department. These official records are sometimes supplemented with documents from private collections and from interviews with U.S. officials. FRUS publishes documents that are at least 35 years old, so that we are now seeing documents from the late fifties. Documents selected from the official record are transcribed and often differ in content from the original, reflecting national security concerns at the time of publication.

In addition to this basic published resource, Yale holds a significant collection of unpublished resources documenting U.S. foreign relations and diplomacy, including the papers of Colonel Edward House, Woodrow Wilson's adviser; Henry Stimson, Secretary of War in the Taft administration, Secretary of State for Herbert Hoover, and Secretary of War again under Franklin Roosevelt; Dean Acheson, Secretary of State in the Truman administration; and Cyrus Vance, Secretary of State for Jimmy Carter. Yale also holds a large number of collections of materials from their associates and other prominent figures in U.S. foreign policy. These papers, including diaries, memoranda, minutes of meetings, speeches, and writings, are in heavy demand by scholars around the world and are vital records of U.S. activities from the Wilson administration to the present. The linking of these materials with those represented by the Foreign Relations series would provide scholars with an unparalleled assemblage of materials to study the evolution of United States foreign policy in the 20th century. This linkage follows naturally from the unified navigational structure we propose.

Yale has a large collection of official records of the United Nations General Assembly and United Nations Security Council. These serve as the basic resources for understanding the workings and evolution of the world body. They include verbatim meeting records, draft resolutions, letters relating to their agendas, and adopted resolutions and decisions.

There is an existing effort, Project Open Book [43], whose goal is to begin converting these materials to digital images. We now have funding from the National Endowment for the Humanities to create a large corpus (3000 volumes next year) of online material in image form from Yale microfilm collections and anticipate further grants to extend that work, possibly by a few thousand more volumes per year. We feel that by the time that the NSF grants are awarded we will have well underway the creation of a world class online collection of international relations materials of national importance.

Scholars from around the country already make heavy use of Yale's special collections in the area of history and political science. As we digitize these materials and their indexes, we will recruit scholars at other institutions to make use of the emerging on-line database, and to incorporate indexes to some of their collections (if not the collections themselves). The proposed system will provide scholars studying United States foreign policy with a unique digital library of documents supporting both of these fields, as well as navigation and access tools that support the particular methods of discourse and communication used in the fields.

We do not imagine that most archival material will ever be easily accessible by machine. However, there is no reason why collections like the *Foreign Relations* series could not be. In fact, the State Department has plans to put it on CD-ROM in the next year or so. Past volumes will have to be scanned in. Of course, nowadays most government memoranda exist in electronic form somewhere, but the need for secrecy makes it questionable whether scholars will be granted on-line access to it anytime soon. The laborious process of searching dusty files will be with us for quite a while, probably, although in the future the dust may be settling on floppy disks rather than paper.

However, the picture changes after a researcher has made a copy of an archival document. In the future, rather than copy it into thin air, the system should copy it into an information space as envisaged here, so that the next person who comes along can be directed to the copy instead of the original. In the long run, this will require government involvement as a file server for copied and catalogued electronic archives.

2 Indexing

A key problem area is the *index structure* of a collection of data. This is a structure that organizes the data along various dimensions. Some examples:

1. A classical card catalogue. It is now standard practice for these to be on-line. The only change is that from the card for a document it will be possible to access the document itself (for a growing population of on-line documents).
2. Document sectioning. For a given document, the system will store its overall structure in the traditional way, using a table of contents. It might also be possible to automate a traditional index. That is, if the index says "Clams, 205," then the user should be able to click and get to page 205.
3. A timeline. Especially for these historical documents, the user should be able to see a linear sequence of time points, each of which refers to a collection of documents for that time period.
4. A geographical map. The user sees a map of the world, with relevant areas labeled. By zooming in on an area, he or she can get references to documents relevant to that area.
5. Associative networks. This is a very general type of index. To take a specific example, consider a network of historical figures. The user will see a graphical representation of a piece of the network, with, say, Margaret Thatcher at the center. An arc labeled "husband" points to Denis Thatcher. An arc labeled "monarch" points to Queen Elizabeth, and so on. Zooming in on a node will lead you to documents about the person. Zooming in on an arc will lead you to documents about the two related persons and their relationship.
6. User-built indices. As users plow through and organize a document collection, they can build up new indices, presumably on the relationship-network model.
7. Hierarchical combinations of the indices. These indexing methods will gain in power by being combined. For example, zooming in on "France" could get you to a timeline, from which you can go to the fourteenth century, pick up the reigning monarch, and look at his family tree. Or different indexing structures can be mapped into different dimensions on the computer's display.

At any point, the user is located in a region of "information space" [40] called the *current context*. The computer tries to display the "shape" of the context and the documents associated with it. Contexts are defined by sets of *attributes* and their associated values. The associations can be as simple as keyword-value pairs, such as **<defense,present>**, where **present** means just that we're interested in documents containing the word **defense**. Other keywords can have a set of possible values, and a region can be denoted by giving a subset of the set: **president**$\in$ {**jefferson, madison**}. In historical domains, space and time are key attributes. The user might be interested in **time**$\in [1939, 1993]$, or in **region** = **Europe**.

The user gets to documents by navigating through this space, extending, contracting, and modifying the specification of the context. At any point in a session with the system, until the user has dived into a particular document, the screen will show the current context and the documents that are presumed relevant to it. Once a particular document has been chosen, the user will be able to read it, see images of the pages as they actually look, or follow hyperlinks out of it, to other documents, and to other regions of information space.

Let's postpone talking about the documents until Section 2.1, and stay in information space for a while. One way to display the current context is as a simple table:

Attribute	*Value one of:*
`Year`	*Range* `1939--1993`
`Place`	`Germany`
	`France`
`Topic`	`Borders`

However, this sort of display fails to take advantage of the bandwidth of a modern workstation. When dealing with numerical values like space and time, we can instead map these onto the spatial dimensions of the computer's display. Our domain is diplomatic history, so it's natural to organize data by geographical area and historical era. If the context encompasses a particular region, then we can display that region on the screen. The user indicates the region of interest either by mouse-drawing or typing. (If the focus is Europe, the user can mouse and click, or type `Europe` to a prompt.) He or she then moves to the time-period of interest, by typing, say, "1993," or using a timeline (see below). The display now shows a map of Europe with boundaries and place-names correct for 1993. See Figure 1.

Superimposed on the map are a bunch of blips. Each blip corresponds to an item in the collection (a book, document, or some organizational unit) that pertains to European diplomacy in 1993. The blips are localized geographically as far as possible. For example: blips corresponding to items having to do with the Channel Tunnel are clustered in the English channel area. The user can, of course, open a document by clicking on it, or select a set of documents graphically and see a list of them.

If the user chooses a time *interval* as part of the context, then an alternative view is of a chunk of space-time. Two dimensions on the screen correspond to geographical dimensions, and the third represents time. See Figure 2. To make the three dimensions look real, the user can rotate the figure in real time with a software "knob," getting a genuine sense of depth.

In such an array, we will use color to indicate the estimated relevance of various objects, and shape to code their type (book, treaty, researcher annotation, mailbox, etc.). Figure 2 can't really do justice to what we have in mind, since it is static and monochromatic.

Rotation for three-D effect is just one way to exploit change of the display over time. Another is to show waves of "relevance" moving out from an initial set of blips. Suppose that initially bright, warm colors, such as red, are associated with relevant documents, with less relevant ones colored blue. Upon the user's command, the display can change the color of documents that mention persons mentioned in the initially warm-colored set, and then propagate that to further documents, and so on. By watching this movie a few times, the user can get a feel for how the current array of documents and *dramatis personae* hang together. We can call this the "space-time interface" (STI) [17, 15]

Some archival holdings lack author, publisher, title; even their contents might be debatable. The STI allows them to be tied down within a uniform index-space nonetheless. (If an item's place or date are unclear, its blip might be associated with a space-time volume rather than a point.) Mapping each object to a colored point whose position in a spacetime

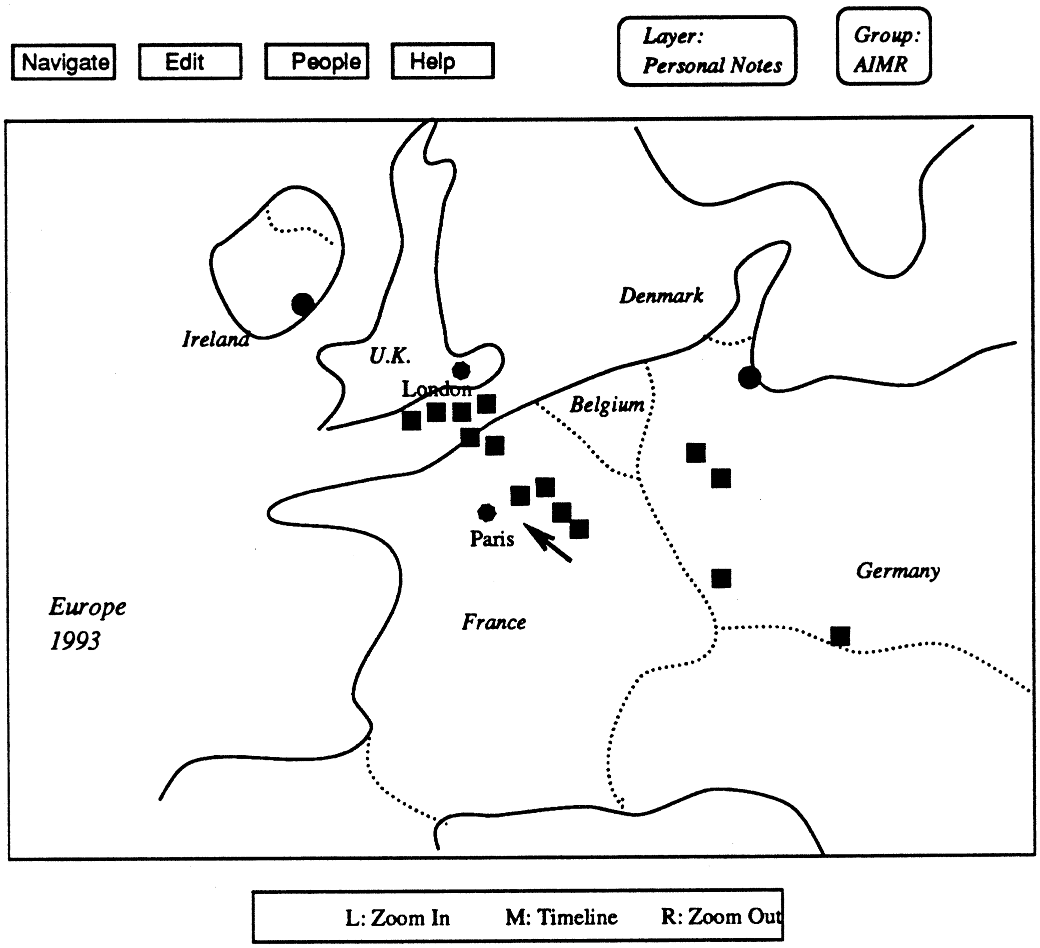

Figure 1: Map-Based Interface

volume is information-carrying yields a highly compact display. The STI might easily show the user several hundred points, and support browsing among them guided by some idea of their potential relevance; conventional indexing strategies based on card catalogs seem unlikely to yield anything like the same information-richness in a display. Date and time are a relatively unambiguous route of entry into a library's holdings.

In our figures, we have shown different shapes of blip, because there will be several different kinds of entity stored at point in information space, including message boxes, special associative subnetworks, or even *people*. As we discuss in Section 2.2, allowing students to find experts and working scholars to find each other is a goal our new library software should pursue. An expert who is willing to participate might be "filed" under the spacetime volumes that define his or herparticular interests, with keywords or feature-value pairs attached. Thus while exploring (say) the Weimar republic, some of the blips encountered might designate scholars who are interested in this period. Just finding their names might be useful; their names can be used as keys in a conventional search for material they've written. The system should also allow the user (by clicking) to find out more about

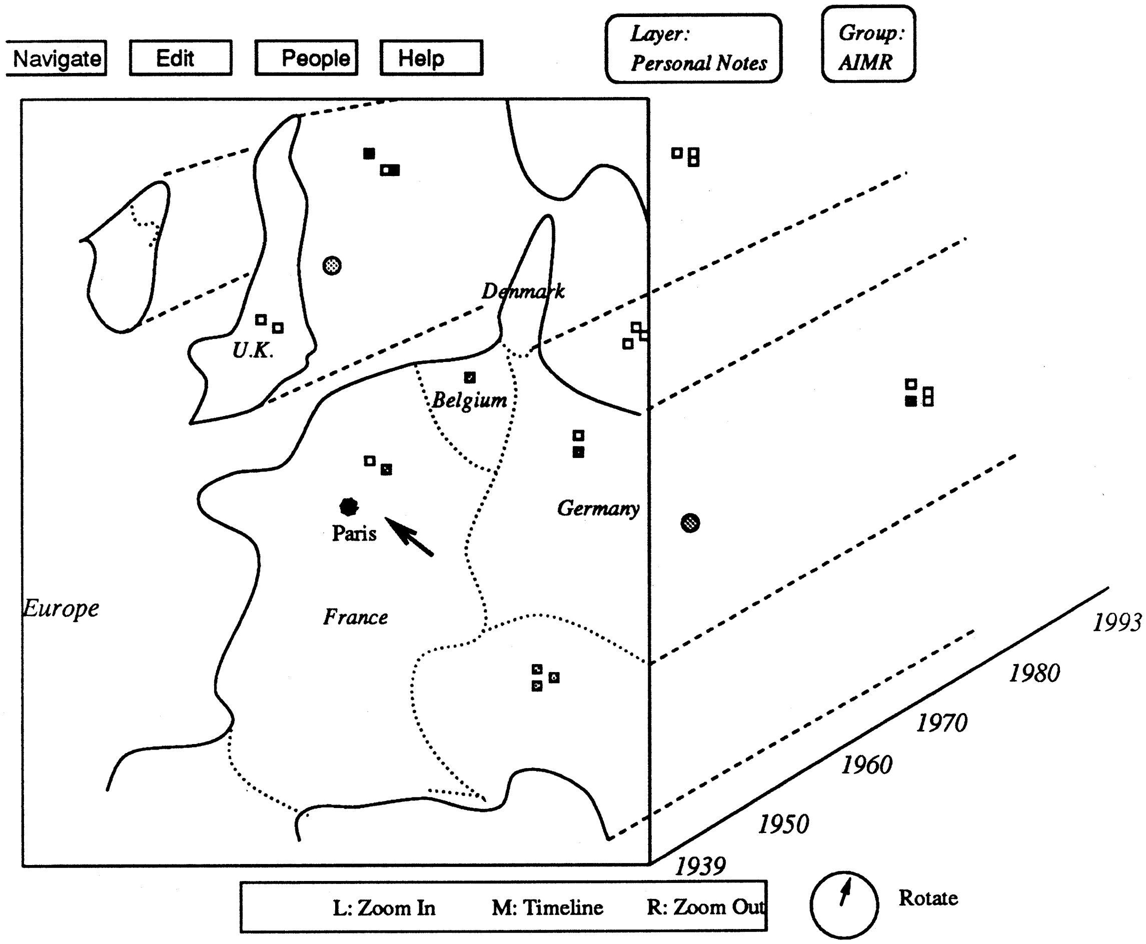

Figure 2: Space-Time Display

what they've done and about their current projects, and should allow email to be sent to them. The spacetime interface is a good place to leave scholarly queries as well: civic-minded users would mouse on query blips and, if appropriate, use the system to send email to the queriers.

Exploring exactly how the STI will work — and how well — is one of the crucial goals of this research.

What we have been discussing so far are *document-set displays,* where the focus is on the display of the structure of a document set. Another kind of interface is provided by the *index-structure display,* which allows the user to explore and manipulate the concepts used to organize the documents, while watching how the document set changes at a fairly coarse scale.

In the mockup displays of Figures 1 and 2, we have shown various blips representing documents and other objects of interest. Normally, the blips can appear only if enough of the context is filled in to make the display comprehensible. For example, if the user indicates `Topic=Trade`, then what are displayed are objects related to international trade

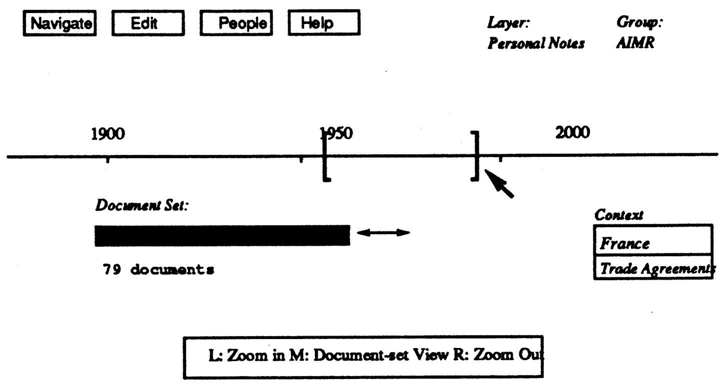

Figure 3: Selecting the Time Interval

agreements. If the user doesn't supply enough context, the document set can just be an unmanageable blob, even allowing for the increased clarity we expect from space-time indexing. Traditionally, users have coped with this frustrating situation by guessing new keywords to conjoin until the document set shrinks to a small but nonzero size [39]. A more interesting approach is to allow the user to choose a dimension, and watch the document-set size change as the context is altered along that dimension. For example, Figure 3 shows a display in which the temporal dimension can be varied while the rest of the context stays the same. The user can expand, contract, or shift the time interval of interest, then switch back to an altered "Document-set View" like the STI display of Figure 2 to see what the structure of the resulting document set is.

Index-structure displays are not confined to variation along a numerical scale. Another possibility is to see a network of associations. As the cursor selects different nodes in the net (shown enclosed in ovals in Figure 4), the document set can change size. As before, the user can switch back to the document-set view to get a feeling for the structure of the result, by seeing a one-, two-, or three-dimensional array of the documents.

Although certain configurations will be standard, how the screen's dimensions are mapped to the context's dimensions is arbitrary, For instance, associative nets can be used instead of space and time. The screen can show a set of documents located at a table of attributes and values, with links to related tables and extended document sets. In Figure 5, the context table with the heavy boundary is the primary focus, and the document set shown there is derived by filling in default values for context elements not shown. Nearby nodes in the space of possible contexts have alternative key values and the documents that are thereby generated.

We plan to explore ways of generating neighboring contexts automatically, by studying methods such as:

- Keep track of the statistical relationships among symbols. There are a variety of ways of generating numerical estimates of the similarity among documents, including probabilistic information retrieval [32], connectionist back-propagation schemes (e.g., [38]), and "latent semantic indexing" ([22, 20]), in which discriminative dimensions are automatically imposed on a space of documents classified by keyword.

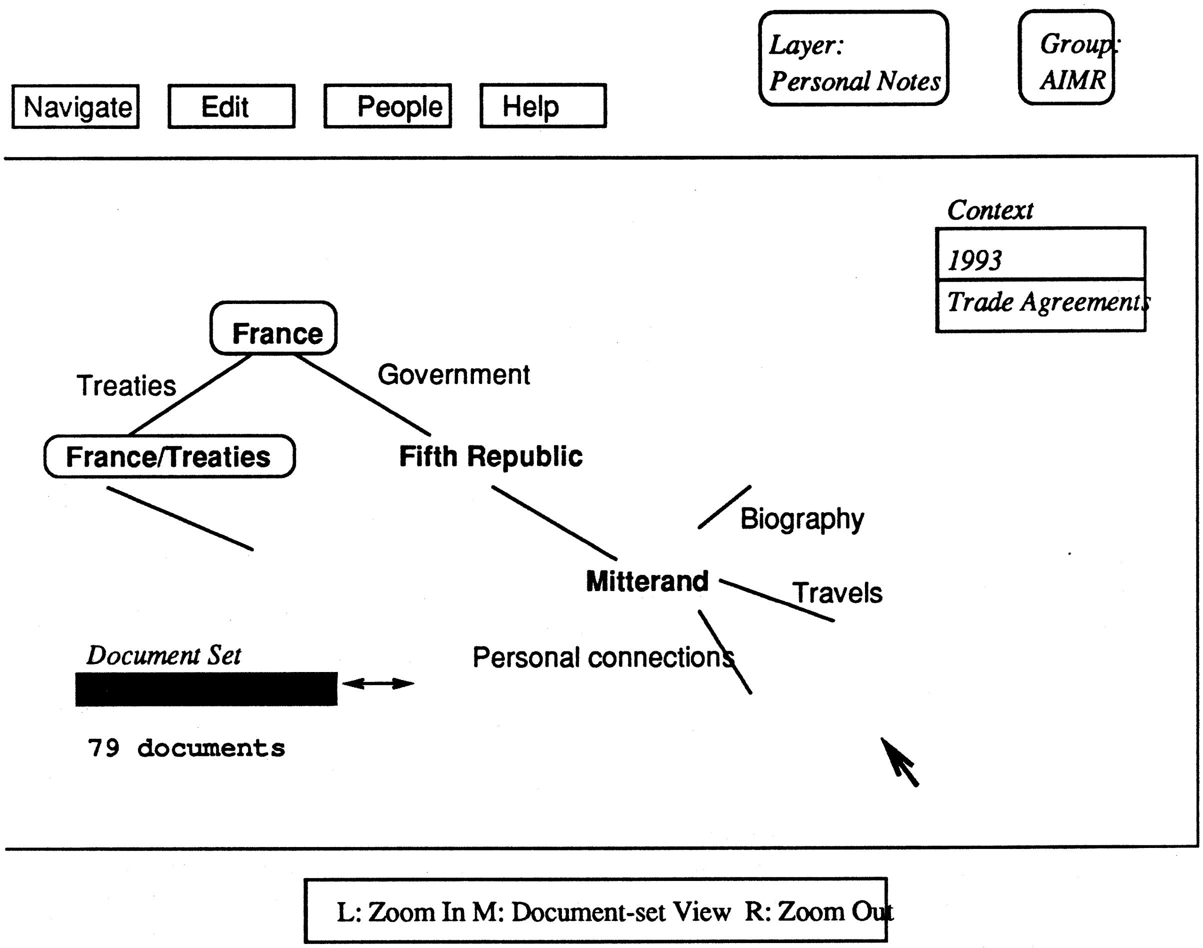

Figure 4: Associative Network Display

- Use an expert database (xdb) [16, 19, 18] to discriminate among different kinds of associations. For example, "Lenin" might be *diagnostic* for "Russian Revolution," because he is often mentioned in that context, and few others, so that documents relating to Lenin are automatically relevant to documents relating to the Russian Revolution.

The use of associative nets will address the seemingly trivial but crucial issue of *entity naming*. Many information-retrieval systems assume that indexes are made of records whose fields are character strings [39]. When creating a query, the user must be sure to type the same strings that were used as index terms. Sometimes glossaries of terms are supplied to make sure that the user stays within the special "controlled" subset of natural language, but most users are making guesses most of the time about how exactly to refer to entities. If you want to retrieve material about the Russian Revolution, do you use the string "Russian Revolution," or some combination of "revolution" and "Russia" (or "Russian"?). A particularly notorious case is the treatment of proper names, which can vary wildly from country to country, and from document to document.

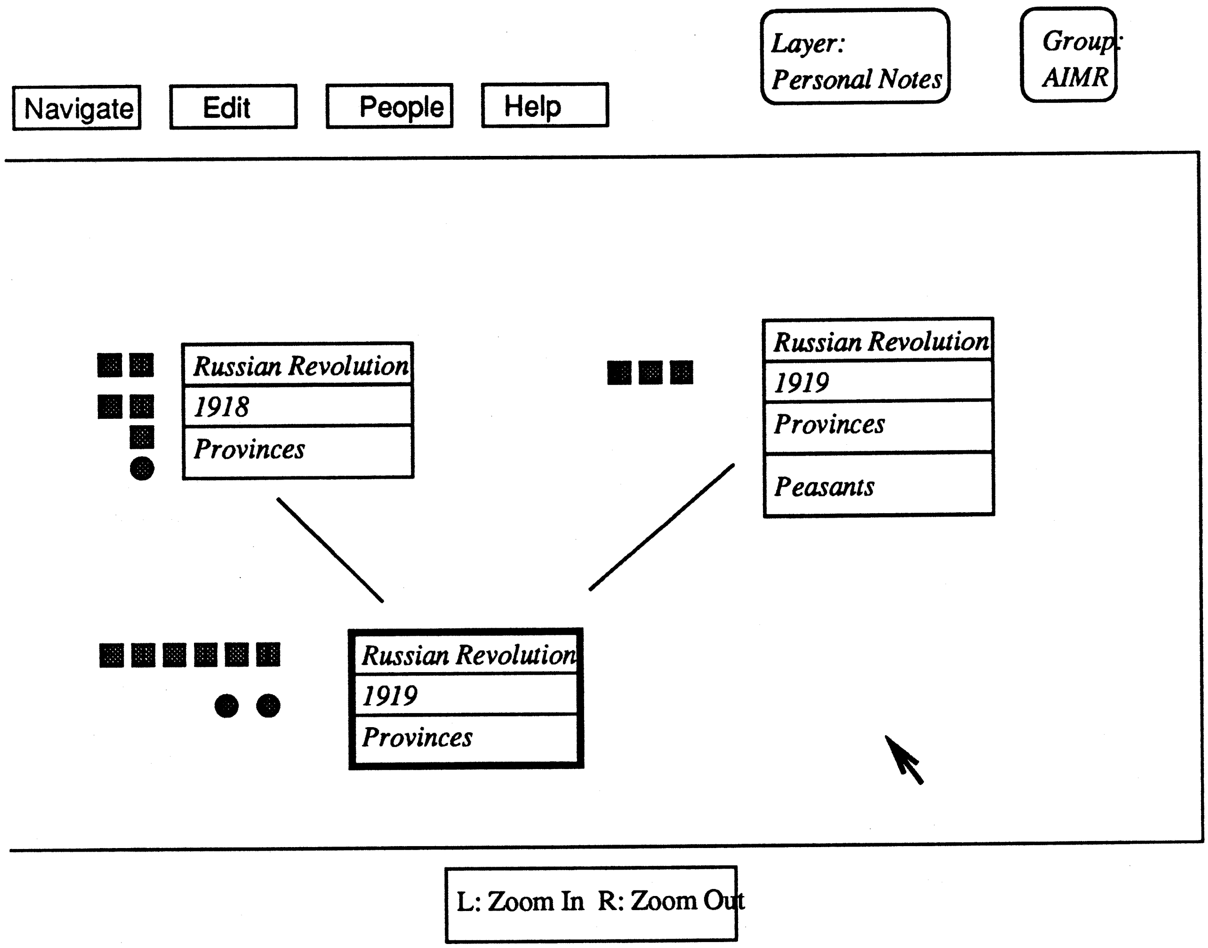

Figure 5: Neighboring Contexts

There are no easy solutions to this problem, but the first step is to separate entities from their names. Documents will be indexed by the entities they refer to, and the entities can have more than one name. The "name" relation will be just another arc label in the associative network for a domain. So the Stalin node could have two names, "Josef Vissarionovich Stalin" and "Ioseb Dzhugashvili." When the user types a string, the system attempts to figure out which node in the net is meant, and lets the user know when it fails to find a single candidate. But in many cases, the user will already be located in an index structure, and so can navigate to a name by following pointers as well as typing strings.

We will be bulding a large, distributed system, so management of entity pointers will be a crucial question. (See Section 3.2.) A major responsibility of the system managers is to find cases where the same entity in the world has two entity nodes in the system, and merge the two nodes. The system must take responsibility for finding candidates to merge, because the database will be too vast and responsibility for it too amorphous for a single archivist to scan it looking for duplicates. What the system must do is look for entities that have similar names and similar releationships to other entities. (E.g., if a government has two heads of state at the same time, that might cause an alarm to go off. Of course, the users will also notice such anomalies, and must be allowed to eliminate them locally without permission.)

When two entity nodes are merged, one will be marked primary, but it will point to the now-redundant node that it shadows. Documents indexed under the redundant node will still be found, because any search starting at the primary node will also retrieve documents indexed by the redundant one.

2.1 Documents and HyperDocuments

So far we have focused on how users navigate through the index for the documents. But eventually a user must bring up an actual document. Here the object we are looking at does not have to be designed; it is given, and in fact, to preserve the integrity of historical scholarship, must not be disturbed. However, we want researchers to be able to annotate documents the same way they make annotations elsewhere in the system. See Figure 6, which shows a document from [34], p. 2365. In this figure, there is a hyperlink, surrounded by an oval. Clicking here will take the browser to a document or note about the French Foreign Minister in 1950. There is also a personal note tacked to the document, which is not normally accessible to other browsers.

There is also a "Message" signal flashing in the upper-right-hand corner of the screen. That indicates that another user has left a software agent in some part of the present context (possibly this same document), and that agent is trying to contact you (or perhaps everyone who fits some profile). Clicking on that button will set up communication with the software agent, and ultimately to the person who left it. That's the topic of the next section.

2.2 Virtual Meeting Places and User Input

Often the most refined information about a subject can be found not on paper or in on-line records but in the minds of scholars in that area. Thus in addition to cooperative indexing facilities, the system will provide a mechanism for real-time communication between users. The purposes of this mechanism will be:

- To allow collaborative work between distant colleagues.
- To make the system easier for new users, by allowing them to take advantage of the knowledge of those with more experience.
- To enhance the system's usefulness as a tool for teaching and education.
- To enrich the ability of the users of the system to act as a scholarly community, by providing a social space for informal interaction and the transmission and expression of ideas that are not yet ready to be committed to permanent form.
- To supplement the indexing system in the database by allowing a user to direct other users to an entity simply by leading them to a virtual location, without needing to understand the details of the indexing scheme or to create new indices or pointers.
- To provide a natural mechanism for integrating software agents into the system, by treating each one as just another user.
- To bring together scholars and students with similar interests.

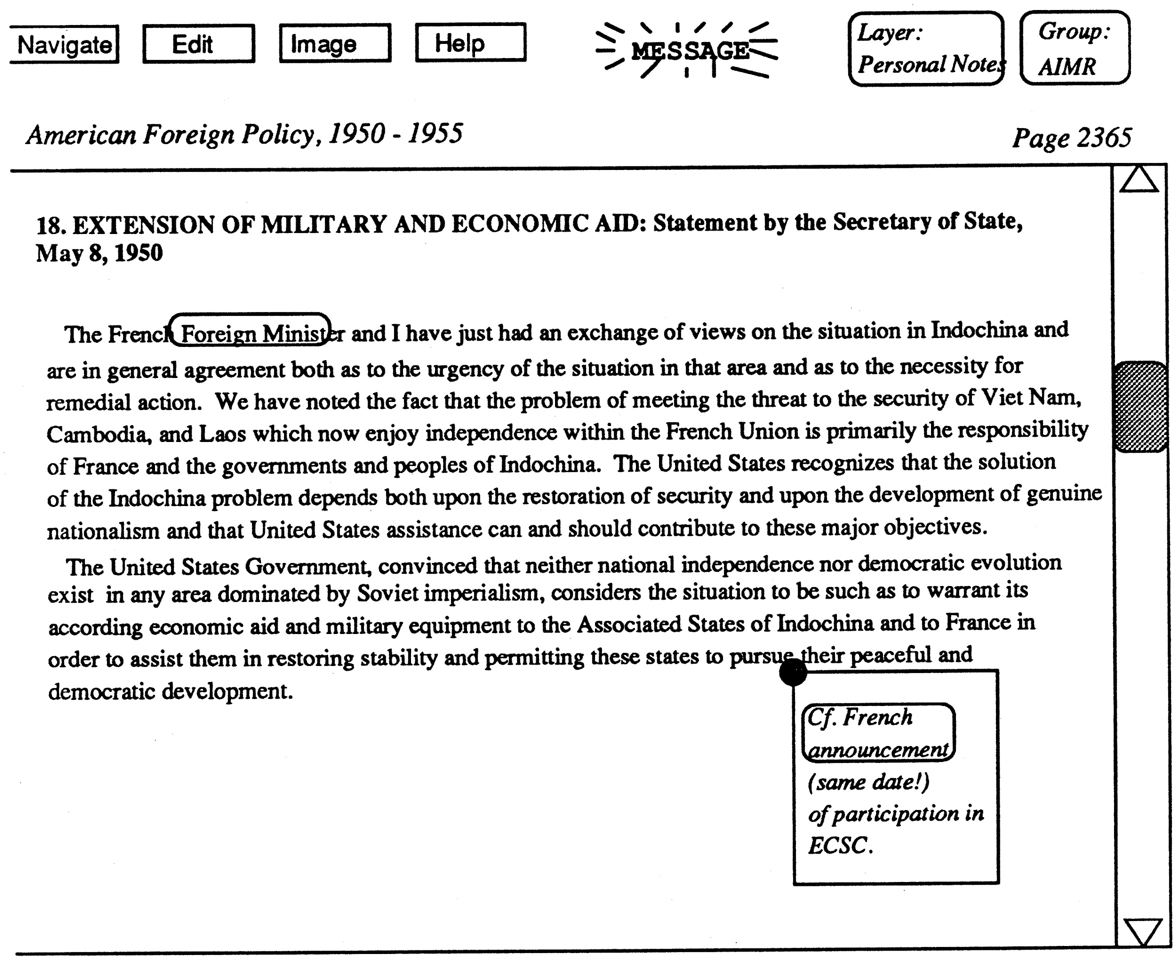

Figure 6: Typical Document

- To make it possible for the confused, the overwhelmed, and the lost to recover their bearings by asking passersby for directions.

To accomplish these goals it is not enough simply to provide a communications system parallel to the database system. Instead, we propose to integrate the communications system into the database at a fundamental level, so that both users and data are visible in the same conceptual space. The ultimate goal is to populate the virtual world of the system: from the individual user's perspective, the system will look more like an active scholarly community, made up of other users, intelligent software agents, and active databases, than the dead, empty library stacks modeled by traditional information systems.

Of course, to achieve this vision there are a number of social issues that must be addressed beyond the mere technical question of how to support communication among large numbers of users. For example: how will the system protect users' privacy against surveillance and unwanted intrusions? How can it help organize the interactions between users to prevent a "cocktail party" effect where users are inundated with irrelevant or unwanted conversation?

We propose to have users control their interactions with other users by making themselves selectively visible in two complementary ways. The first and simplest is for a user to leave a pointer to himself or herself in a document or index; following the pointer will give a connection to that user (if the user is available and currently willing to accept connections in this way.) This mechanism is conceptually no different from allowing users to leave their phone numbers in appropriate places. The second and more innovative method is to allow users to make themselves visible in connection with the documents they are using. Just as the separate rooms in a large physical library sort its patrons by their interests, the local regions of geographical indices or of associative networks can be used to create virtual spaces in which serendipitous encounters between researchers with overlapping interests are possible [24].

Virtual meeting places are merely one example of a broad range of new "electronic community" services that our system might provide. For example: we might associate an electronic journal with every book in the collection. The author or his publisher uses the journal to disseminate corrections, changes or responses to criticism; users may record their comments; librarians may add references to related materials. Each new library user might have a "tracking demon" created for him automatically; the user's demon is responsible for reminding him to return books, and forwarding recall requests; it also develops a model or profile of the user's interests, and notifies him when new library acquisitions might be of interest. There are many other possibilites along these lines.

The basic technology for providing "virtual meeting places" already exists, especially in the context of "MUDs," or multi-user dungeon games. See [3, 10, 11]. But it has never been combined with the idea of navigating through an index structure, and that will be our focus here.

As users move through an index structure, they will be encouraged to think in terms of visiting a set of places in an abstract space. One reason for this is to encourage the kinds of communication metaphors we have discussed. Another reason is to allow users to think of themselves as leaving a trail behind them, a trail that can be used to augment the index structure.

The technology for organizing document collections into hyperspace is not that novel. The main problem is labor, as it were. Just reading all the documents once is a tremendous undertaking, let alone indexing them. The trick is to get the users to do it, by providing such an attractive package of tools that they will become addicted to them. For example, we replace the photocopier with a scanner that provides images on demand, but also allows word searches (by doing OCR and noise-tolerant string matches). We allow marginal notes on documents, and control over who can see those notes. If while examining document A, the user asks for document B and then C, the system can build a relationship network among the documents, and prompt for a label for the relations involved [40].

Or suppose that a scholar becomes interested in the movements of the principals during a certain crucial one-month period of the Arab-Israeli peace negotiations. The way things are now, he or she would jot down notes on an index card, or perhaps a word processor. It would be much nicer to put those notes in the system, with hyperlinks to documents that provide evidence for those movements. For example, it has been proposed [42] that a shift in U.S. foreign policy in 1950 in favor of helping the French in Indochina was directly linked to a shift in French policy in Europe. The French agreed to join the nascent European Coal and Steel Community in return for American agreement to help them quell the rebellion in Vietnam, or so it has been claimed. One piece of evidence is formal diplomatic letters written on the same date in 1950 that announce these changes. (One is shown in Figure 6.) Copies of those letters, and other related memoranda, could be stored on-line and linked to the schedules of the diplomats involved.

This picture is pretty, but raises some serious questions of security and privacy. On one hand, users of the system will want the ability to make private notes. On the other hand, the system administrators can't allow just anyone to alter the contents and structure of system indices. Private notes are, of course, not a big problem, or at least not a new problem. Totally private information can be stored in the user's own workstation, and associated with documents as they are brought up. A more interesting issue is controlled sharing with collaborators, but various traditional solutions present themselves.

The question of allowing users to modify contents of the index is more interesting. Some users can't be trusted, and some can be trusted but lack competence. Hence publicly visible modification of the index will have to involve a partnership between scholars and archivists. Primary responsibility for alteration of the network will rest with the archivists, but the inspiration for new organizations will come from the scholars. Most scholars probably don't want to spend any more time on organizing the data than is directly profitable to them, so we have to find a way to get them to help without actually doing anything they wouldn't do anyway. We propose developing a system of *index strata* to accommodate this requirement. The base stratum is the official index structure. (It will involve all the spatiotemporal hierarchies we mentioned earlier; the stratum organization is orthogonal.) If a researcher or group of researchers wants to organize a collection of data in ways that extend the standard index, they can create their own stratum on top of the base, and change it anyway they like. For example, a group of advocates of the theory that all of twentieth-century history is governed by the activities of the Trilateral Commission might maintain their own timeline of historical events and how they relate to meetings of the Commission. A stratum would inherit the structure of the stratum below it, but any modification would be possible. One way to accomplish this is with the techniques of *assumption-based truth maintenance* that have been developed by AI researchers [31, 12, 36].

The archivists would keep track of emerging strata and help make them more publicly usable, or even make them part of the base stratum if circumstances made that desirable. For example, scholars might develop an associative network describing an organization (e.g, the UN). For an organization, the map would include descriptions of the organization's scope of functions and activities; descriptions of the organization's structure; identification of staff and their roles and offices in the organization; the resources the organization uses to carry out its mission; relationships the organization has had with other organizations or persons; and its involvement with events in both its own history and that of the larger society of which it is a part. Once the scholars work all these details out, the archivists can take the structure over and make it a part of the base stratum for indexing materials connected to that organization.

The evolution of the library also creates problems with searching when new search techniques do not apply to earlier holdings. For example, a new relation "supersedes" could be created and applied when United Nations resolutions supersede previous resolutions. Any new U.N. documents added to the library would have such a relation if appropriate. When navigating though resolutions, scholars would be notified that a previous resolution existed, or that the current one has been made obsolete. During researching, the scholar would typically receive frequent notifications of this type and could easily assume that a document was not obsolete when no notification of obsolescence is provided. However, the notification might simply be absent because the document which made it obsolete was included prior to the creation of the relation. The library interface must help guard against this type of user error.

3 Document Representation and Storage

Most of the data currently exist as physical or microfilmed documents. These documents will have to be digitized and stored. There will be huge amounts of data, which mustn't depend on the integrity of a particular centralized file server. Hence we must store the data in *compressed, distributed* form.

3.1 Data Compression

We will need two different data-compression technologies: optical character recognition (OCR) for text, and image compression for illustrations. We focus on the second topic.

Some important theoretical advances have been made in image compression, by a group centered at Yale (including Vladimir Rokhlin and Rafi Coifman of the Computer Science and Math Departments). The theory, which we can call *adaptive waveform analysis,* is a tool kit for understanding large sets of measured data: sounds, images, seismic vibrations, atmospheric pressure maps, electrocardiograms, etc. The idea is to describe the data in terms of carefully selected collection of templates called *waveforms.* (Wavelets, wavelet-packets and Local Trigonometric Waveforms are used as our elementary "atoms" for feature extraction or "alphabet" for data storage.) This may be thought of as a generalization of Fourier transform theory, which replaces complex-exponential basis functions with bases from different families. The choice of family depends on the application at hand. When the right family is chosen, the compression scheme has the property that most of the signal is captured by a few of the terms in the expansion of the data. The remaining terms, which often just encode noise, can be discarded without losing much. This allows dramatic compression.

The following analogy may clarify how the process works. For one-dimensional data such as audio we should think of waveforms as musical notes having three characteristics, pitch (or oscillation number), duration, and intensity (or amplitude). An efficient description of a recorded piece of music can be given by writing a musical score. Adaptive waveform analysis provides a mathematical alphabet (of notes) and a method for transcribing a recorded signal into a most efficient notation.

This is achieved by matching the patterns in the recording to a library of waveforms and selecting the "shortest" possible transcription that preserves "most" of the data. As a result, the storage requirements are lowered substantially. Moreover, tasks involving extraction of features, elimination of noise, diagnostic analysis are considerably simplified and accelerated, because these processes can often be expressed as a manipulation of the compressed data representation.

For two-dimensional data such as images, or three-dimensional data such as video, we should think of the waveform as simple patterns or templates which can be used by superposition to resynthesise the data in a way somewhat analoguous to overlaying various simple transparencies to form a complex image. This "musical notation" for images and video is a powerful processing tool, enabling computational tasks which are otherwise impossible to achieve even on the most elaborate hardware. (For example, in areas of acoustic and electromagnetic scattering, or even in elaborate tasks as image rendering .)

To date algorithms developed at Yale have been adapted by the FBI and Scotland Yard as being the most efficient method for storing digitally their library of fingerprint cards, permitting a 20 fold saving in storage and transmission time. Other applications for medical diagnostic and medical image enahancing as well as feature extraction have also been developed. In a joint project with J. Berger at the Yale Music School, an old recording of Brahms, made by Thomas Edison, is being processed for enhancement as a "musical archeological dig". A big problem in this noisy recording involves finding all of the actual notes played by Brahms. As an application of this method various remarkable enhancement methods for echoplanar magnetic resonance fast imaging have been obtained.

3.2 The PASO Model for Distributed Databases

A digital library contains many data objects that are accessed by many different users. The data objects may be a text bibliographic reference, a text document, a compressed image, voice, video, or any other hypermedia. The underlying software that implements the library must therefore support many kinds of data that will undergo many different kinds of searching and update operations. The system should be distributed, highly accessible, and highly reliable.

We believe that these goals can best be achieved through the use of a model called Persistent, Associative, Shared Object (PASO) memory ([23, 21, 45]). The PASO model generalizes the notion of *Tuple Spaces* that are underlies the Linda coordination system [9]. A PASO memory stores a heterogeneous collection of data objects. It supports a wide range of search and update operations using the principle of associative matching. A *search template* is used to specify the general form of a desired object, and the system returns any object that matches the template. The search templates can be very general, such as a template that asks for any data object consisting of a text document with the phrase "United Nations" in the title, or very specific, like a template that searches for the current circulation record of a book identified by its Library of Congress catalog number and copy number.

There are two main programming models of communication in a multi-process system: message passing and shared address space. In a message-passing model, each process has a private memory and a set of communication channels through which it can exchange information with other processes. Being very close to the actual architecture of most multiprocessor systems, the message passing model is well-suited for writing programs that explicitly manage the various constraints and advantages of the communication architecture. Yet, just as in the case of assembly languages, message-passing programs are hard to write and reason about.

In the shared-address-space model, processes share a random-access memory. A location in the memory can be read from or written to in a single atomic operation, and all processors have the same mapping from addresses to locations. The model offers ease of programming and debugging as its main attractions. Truly sharable RAM does not yet exist, however. Current implementations simulate it by distributing virtual memory pages among local processor memories and running software and hardware protocols to maintain consistencies.

Hence, the model often hides from the programmer the potential cost of memory access, and the theoretical elegance leads to practical inefficiency.

The PASO model is a hybrid of the two approaches. In the sense that pass-by-value is a mechanism for passing arguments without reference to physical addresses, associativity — reference-by-value — is a mechanism for referring to objects without mentioning physical addresses. The PASO model offers the convenience of shared memory and some of the efficiency of message passing. It permits the programmer to distinguish local computations from potentially expensive communication/coordination actions.

An object in a PASO memory is a tuple of values drawn from ground sets of basic data types. The memory contains a heterogeneous collection of tuples, each of which has an arbitrary number of fields. The atomic operations on a PASO memory are **insert**, **read**, and **read&del**. The programmer is oblivious to the physical location of the objects and simply manipulates an abstract object space. A PASO memory is *associative* in the sense that tuples are accessed by pattern-matching. A **read** takes a tuple template specifying acceptable values for each field, and returns any one object matching that template. Both **read** and **read&del** have a blocking and a non-blocking version. For example, if one attempts to **read** an object that does not exist, a blocking **read** waits for a matching object to be inserted, while a non-blocking read returns an error message. Note that there is no **modify** operation; since an object is defined solely by the contents of its fields, modifying a field is logically equivalent to destroying the old object and creating a new one. There is no loss of generality, since a mutable distributed data structure can be built out of collections of immutable atomic objects.

A PASO memory is *shared* in the sense that any object can be accessed by any participating process regardless of the physical node hosting the process. A PASO memory is *persistent* in the sense that the lives and deaths of objects are independent of the lives and deaths of their creating processes.

Shared memories that qualify informally as PASOs have been used as coordination languages in a variety of parallel programming systems, e.g., in the context of C [9], Scheme [26], Prolog [8], distributed object-oriented systems [30], Modula-2 [7], program visualization systems [37], math libraries [13], and as part of other coordination mechanisms [1, 28]. They have proven to be an effective basis for concurrent programming in many multi-processing environments [9]. The fact that "informal PASO memories" are a pragmatic success makes them good candidates for formal, algorithmic, and theoretical research that aims at improving them.

The PASO model can support quite general distributed data structures. For example, a hypertext can be implemented as a collection of primary data objects, one for each page of the hypertext. Fixed links between documents are easily implemented by using pointer fields, and intertext links can be displayed through graphical interfaces similar to those currently being developed as part of the Linda project at Yale. More general searching is also possible. A user can select any substring from the hypertext and use this string as a search template to find any other object that contains it. This type of search is very powerful but potentially inefficient. The distributed nature of the PASO system allows for parallelization of the search process, and an important focus of research is on how to best utilize the distributed computing power to make such general queries run fast.

The Yale PASO system is intended to be platform-independent and to run on a wide variety of machines and operating systems. The first prototype of the system, currently being implemented by graduate students in the Department of Computer Science, is built on top of the Isis system ([4, 25]). Isis provides fault-tolerant communication functions based on the TCP/IP protocol suite. Isis can operate in a heterogeneous environment of workstations running different operating systems. Consequently, a PASO system running on top of Isis is effectively platform-independent. Although Isis provides some long-haul capabilities for handling wide-area networks, it is primarily a tool for local-area networks. One of our immediate research goals is to expand the PASO implementation so that it can run on a wide area network.

The PASO model offers flexible treatment for security problems such as defining different classes of users in the system. One can easily impose a hierarchical structure on PASO objects, possibly via other PASO objects. Navigating this structure will involve obtaining some kind of handle, most likely from a PASO object, that will carry the permission information. A similar approach is used Open Tuple Spaces ([27]).

The PASO system utilizes dynamic run-time data replication both to guarantee fault tolerance and to give improved efficiency during associative search. *Fault tolerance* means that the abstract object space must remain unchanged even though a series of faults occur. Ideally, the fault model would incorporate any kind of failure. In the face of *Byzantine* failures, however, in which errant processes actively lie, cheat and steal, even the most basic concurrent computation is provably impossible. Lynch [29] surveys many impossibility results of this type.

The main kind of fault we address are *fail-stop* errors, in which a processor simply crashes and stops communicating. If a processor crashes, all objects stored at the processor become inaccessible or unrecoverable, and all communication links to and from the processor disappear. A processor failure can therefore destroy data and partition the communication network. If a machine restarts, we regard it as a completely new process rather than a recovered old process. This is motivated by a desire to store many objects in fast RAM memory, which will be wiped clean in a machine recovery. Use of reliable communication primitives ([4, 25]) eliminates basic communication faults such as loss or corruption of messages. Like previous researchers, we exclude Byzantine failures from the fault model. Doing so makes our problem theoretically tractable and in practice Byzantine errors are far less common than the fail-stop and communication errors we do include.

The degree to which the memory is fault-tolerant is parameterized by a value λ. The memory will be guaranteed correct as long as no more than λ faults occur simultaneously. The value of λ can be chosen by the user, or it can change automatically in response to increasing or decreasing frequency of detected errors.

Data redundancy can also improve efficiency. If many processors are reading the same object, then by replicating copies of that object among several processors, the overhead per processor is reduced as is the communication cost. Very general search queries can be performed in parallel, reducing response time. Data objects can migrate through the network to sites from which there are being heavily accessed or updated. Having local rather than remote copies reduces communication cost and response time. The foundation of PASO memory is highly adaptive algorithms that automatically regulate the distribution of data objects over the network, responding on-line to changing patterns of usage.

There is a considerable literature on the use of static replication schemes to improve efficiency (See [14]). State-of-the-art research now focuses on adaptive, run-time algorithms that respond effectively to changing access patterns [5], [6],[41], [2], [35],[46],[44]. Less is known about adapting well to changes in network configurations because of failures, but the problems are closely related. Our thesis is that these theoretical techniques for adaptive replication can be used to efficiently manage the redundant data needed for fault tolerance. Furthermore, as the required degree of fault-tolerance is reduced, an implementation of PASO memory using data replication can actually become more efficient than any that now exist.

Because we will be storing compressed images in the PASO memory, we will be investigating algorithms for retrieving compressed objects associatively. In many cases, adapative waveform techniques support efficient algorithms that operate directly on the compressed form. However, it is not clear that anything of the sort will work with text. This is an open research question.

4 Where We're Headed

A project is already underway at Yale to digitize a corpus of 10,000 documents. To some extent this effort is focused on preservation rather than access, but it makes sense to direct some of the resources of that project to digitizing of historical and diplomatic documents.

As we incorporate more and more documents, and erect an indexing superstructure on top of them, we'll make the system available to various groups, both at Yale and at remote sites. To access the system, it will be necessary for a client to run the X window system, and be able to receive high-volume TCP/IP packets to control what's displayed. We anticipate that Yale's campus-wide network will support this kind of interaction in the next couple of years. We also anticipate that the Internet and NREN will bring enough bandwidth to allow remote access.

There is no reason why the basic functionality of our system could not be duplicated elsewhere, but in the early years it's probably too big a distraction to try to set up such "mirror" sites.

Currently we are operating on a shoestring budget, and have not yet gotten adequate funding for this project. If and when we get it, we plan a three-year timetable in which we develop underlying database technology at the same time as indexing superstructures.

Evaluation Metrics: It's difficult to evaluate digital libraries, for the obvious reason that they are interactive systems. The mere existence of a pile of features doesn't mean that the system will actually be used.

What we need to do is to interview prospective users, find out their expectations, and then follow up after they've had a chance to use the system. If users report that they'll never go back to pencil, paper, and photocopier, then we'll be happy. To make comparisons meaningful, we will develop a standard questionnaire to structure the "before" and "after" interviews. These results can be reported in appropriate forums for scholars in the area of human-computer interaction.

Of course, there are more objective measurements of what we will have accomplished. How many volumes have been digitized? How fast can a link to a document be followed? How many documents are actually referenced? How many users stay with the system for more than a month?

Another set of measurements concerns the way in which the system evolves. We are especially interested in measuring the volume of information stored in the system by users compared to the volume of information stored by archvists.

This effort will be a collaboration between the Yale Library, the Yale Computer Science Department, and the Yale History Department, with various other groups and private companies involved to a lesser degree.

Acknowledgements: This project involves the work of several reearchers in Yale's Computer Science, Mathematics, and History departments, and at the LabTek Corporation. Portions of this paper were contributed by Jim Aspnes, Rafi Coifman, Tom Griest, and Gaddis Smith.

References

[1] G. Agha and C. Callsen. Actorspaces: An open distributed programming paradigm. In *Proc. 4th ACM SIPLAN Symp. on Principles and Practice of Parallel Programming, San Diego*, May 1993.

[2] Y. Bartal, A. Fiat, and Y. Rabani. Competitive algorithms for distributed data management. In *Proc. ACM Symp. on Theory of Computing*, pages 39–50, 1992.

[3] R. Bartle. Interactive multi-user computer games. Technical report, MUSE Ltd., 1990. Available via FTP from parcftp.xerox.com in pub/MOO/papers/mudreport.*.

[4] K. Birman, R. Cooper, T. Joseph, K. Marzullo, M. Makpanguo, K.Kane, F. Schmuck, and M. Wood. The ISIS system manual, version 2.1. Systems User Manual, Sept. 1990.

[5] D. Black, A. Gupta, and W. Weber. Competitive management of distributed shared memory. In *Proceedings, Spring Compcon 1989*, pages 184–190. IEEE Computer Society, San Francisco, CA., 1989.

[6] D. L. Black and D. D. Sleator. Competitive algorithms for replication and migration problems. Technical Report CMU-CS-89-201, Department of Computer Science, Carnegie-Mellon University, 1989.

[7] L. Borrman, M. Herdieckerhoff, and A. Klein. Tuple space integrated into Modula-2, implementation of the Linda concept on a hierarchical multiprocessor. In Jesshope and Reinartz, editors, *Proc. CONPAR '88*. Cambridge Univ. Press, 1988.

[8] A. Brogi and P. Ciancarini. The concurrent language Shared Prolog. *ACM Trans. on Programming Languages and Systems*, 13(1):99–123, 1991.

[9] N. Carriero and D. Gelernter. Linda in context. *Commun. ACM*, 32(4):444–458, April 1989.

[10] P. Curtis. Mudding: Social phenomena in text-based virtual realities. In *Proc. 1992 Conf. on the Directions and Implications of Advanced Computing*, 1992. Available as Xerox PARC technical report CSL-92-4.

[11] P. Curtis and D. Nichols. Muds grow up: Social virtual reality in the real world. In *Proc. Third International Conference on Cyberspace*, 1993.

[12] J. deKleer. Three papers on assumption-based truth maintenance. *Artificial Intelligence*, 28:127–224, 1986.

[13] C. C. Douglas. A tupleware approach to domain decomposition methods. *Applied Numerical Mathematics*, 8:353–373, 1991.

[14] L. W. Dowdy and D. V. Foster. Comparative models of the file assignment problem. *ACM Computing Surveys*, 14(2):287–313, 1982.

[15] M. Factor, D. Gelernter, C. Kolb, P. Miller, and D. Sittig. Real-time performance, parallelism and program visualization in medical monitoring. *IEEE Computer*, March 1991.

[16] S. J. Fertig and D. H. Gelernter. Machine musing and expert databases. In L. Kerschberg, editor, *Expert Database Systems*, pages 605–620. Benjamin Cummings, 1989. Originally published in Proc. Second Int. Conf. Expert Databases, April 1988.

[17] S. J. Fertig and D. H. Gelernter. Experiments with parallel software architectures for information filtering: Trellis and FGP. In *Proc. of the Bellcore Workshop on High-Performance Information Filtering*, Nov. 1991.

[18] S. J. Fertig and D. H. Gelernter. FGP: A software architecture for acquiring knowledge from cases. In *Proc. of the International Joint Conference on Artificial Intelligence*, Aug. 1991.

[19] S. J. Fertig and D. H. Gelernter. FGP: A software architecture for the intelligent retrieval of information from unstructured data. In *Proc. of RIAO 91, Vol. 3: Intelligent Text and Image Handling*, 1991.

[20] P. W. Foltz and S. T. Dumais. Personalized information delivery: an analysis of information filtering methods. *Comm. ACM*, 35(12):51–60, 1992.

[21] E. Freeman, D. Gelernter, J. Westbrook, and L. Zuck. A fault tolerant paso for lans (extended abstract). Technical Report YALEU/DCS/TR-1012, Yale University, Feb. 1994.

[22] G. Furnas, S. Deerwester, S. Dumais, T. Landauer, R. Harshman, L. Streeter, and K. Lochbaum. Information retreival using a singular value decomposition model of latent semantic structure. In *Proc. 11th International Conf. on Res. and Deve. in IR*, pages 465–480. ACM/SIGIR, 1988.

[23] D. Gelernter, J. Westbrook, and L. Zuck. Towards an efficient fault tolerant PASO memory system. Technical Report YALEU/DCS/TR-1000, Yale University, Dec. 1993.

[24] D. H. Gelernter. *Mirror Worlds.* Oxford University Press, 1991.

[25] I. Isis Distributed Systems. The Isis distributed toolkit, version 3.0. Systems Reference Manual, 1992.

[26] S. Jagannathan. TS/Scheme: Distributed data structures in Lisp. In *Proc. 2nd Workshop on Parallel Lisp: Languages, Applications and Systems.* Springer-Verlag LNCS, Oct 1992. Also published as: NEC Research Institute Tech Report: 93-042-3-0050-1.

[27] D. L. Kaminsky. *Adaptive Parallelism with Piranha.* PhD thesis, Department of Computer Science, Yale University, 1994.

[28] B. Liskov. Position paper. The panel discussion at OLDA2, Vancouver, October 18 1992.

[29] N. A. Lynch. A hundred impossibility proofs for distributed computing. In *Proceedings of the 9th ACM Symp. on Principle of Distributed Systems*, pages 1–28, 1989.

[30] S. Matsuoka and S. Kawai. Using tuple space communication in distributed object-oriented languages. In *Proc. OOPSLA '88*, pages 276–284, Nov 1988.

[31] D. McDermott. Contexts and data dependencies: a synthesis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-5(3):237–246, 1983.

[32] C. T. Meadow, editor. *Text Information Retrieval Systems.* Academic Press, Inc., 1992.

[33] U. D. of State. *Foreign relations of the United States.* Supt. of Docs., U.S. G.P.O., 1900–.

[34] U. D. of State. *American Foreign Policy, 1950–1955.* U.S. Government Printing Office, 1957.

[35] G. Pfister and et al. The IBM research parallel processor prototype: Introduction and architecture. In *Proc. International Conf. on Parallel Processing*, pages 764–771. IEEE Computer Society, 1985.

[36] rew McDermott. A general framework for reason maintenance. *Artificial Intelligence*, 50(3):289–329, 1991.

[37] G. Roman, K. Cox, C. D. Wilcox, and J. Plun. Pavane: a system for declarative visualization of concurrent computations. *J. Visual Languages and Computing*, 3:161–193, 1992.

[38] D. E. Rose and R. K. Belew. *Toward a Direct-Manipulation Interface for Conceptual Information Retrieval Systems*, pages 39–54. Greenwood Press, 1991.

[39] G. Salton and M. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, 1983.

[40] B. R. Schatz. Building an electronic community system. *J. of Management Info. Sys.*, 8(3):87–107, 1992.

[41] C. Scheurich and M. Dubois. Dynamic page migration in multiprocessors with distributed global memory. *IEEE Transactions on Computers*, 38(8):1154–1163, August 1989.

[42] G. Smith. *Dean Acheson.* Cooper Square Publishers, Inc., 1972.

[43] D. J. Waters. The organizational phase of project open book. Technical report, Commission on Preservation and Access, 1992.

[44] J. Westbrook. Randomized algorithms for multiprocessor page migration. *SIAM J. Comput.* To appear.

[45] J. Westbrook and L. Zuck. Adaptive algorithms for paso memory. Technical Report YALEU/DCS/TR-1013, Yale University, Feb. 1994.

[46] A. Wilson. Hierarchical cache/bus architecture for shared memory multiprocessors. In *Proc. 14th International Symp. on Computer Architecture*, pages 244–252. ACM SIGARCH/IEEE Computer Society, 1987.