II. Project Description

The last few years have seen an explosion of computer network connectivity. With the coming development of the proposed National Information Infrastructure there will be an explosion of explosions. The ordinary computer user in every kind of setting (business, academic, or home) will be able to access millions of information items and services, and to be an on-line provider to other people as well.

But access is useless if you can't find what you are looking for and manage what you find. The problem is not one of accessing but of searching, filtering and focusing attention; it is a problem faced by the end user of the system as well as the application writer. Today's "networked digital library" is full of fledgling services and applications with a bewildering variety of access paths, indexes and interfaces; Figure II-1 depicts the situation graphically. We haven't yet produced what

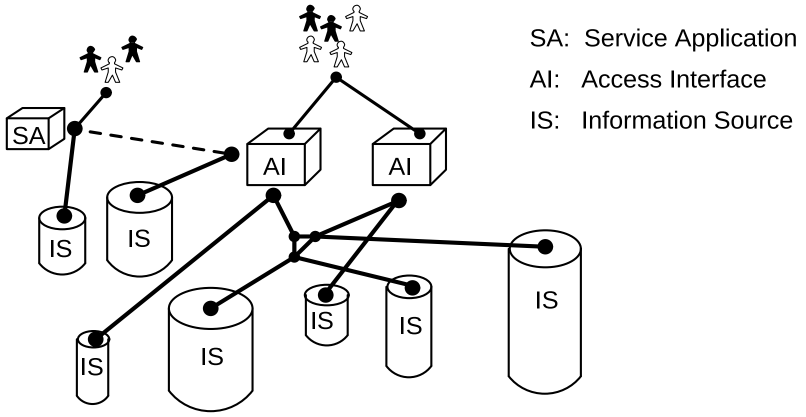

Figure II-1: Today's "digital library"

is needed from the human end: a coherent way of thinking about and managing the diversity of resources. This becomes even more pressing as the on-line world expands to encompass digital forms of image, audio and video data, along with the more traditional textual, hierarchical, and relational data.

This problem would be hard enough if it were one of designing a complex information space from the ground up. It is magnified by the fact that "the net" constitutes a widely distributed information space, populated by millions of distinct machines, managed by millions of people and organizations with differing backgrounds, interests, and legacy resources, with no central mechanism for control or coordination. It would be unrealistic to come up with an idealized architecture and expect people to throw away what they already have in order to adopt it. We need to embrace and enfold the diversity rather than hope it will disappear if only we could produce a "better" standard.

If we can achieve this, we will have much more to gain than simply having easier ways to access information. The networked world is not best thought of as a collection of separate information stores or databases, but as a vast (though chaotic) information space which can be navigated and manipulated in new ways. When we think of the *Digital Library* we go beyond the notions that limit the library of today, thinking instead of the library as encompassing this incipient world-wide information base, with its diverse forms of knowledge and information, and with its vast pool of users, who are themselves able to produce and organize information – the currently divided roles of publisher, librarian and library patron will give way to a more distributed collection of activities in which people can use their specialized skills to augment the library for others.

The goal of the Stanford Integrated Digital Library Project (SIDLP) is to develop a common framework that can support a world of diverse information sources, ever-changing services and diverse interfaces to them. The focus of our project is on theoretical development and the design of conceptual models, protocols, languages and servers, supported by a SIDLP Testbed on which we (and others) can build and test new kinds of services and tools.

There are more challenges associated with such a Digital Library than can be reasonably tackled in any single project. To focus our research effectively we have elected to highlight several aspects of the digital information space that are particularly crucial, and in which we are in a position to make a substantive contribution. Of these aspects, three – *massiveness, geographical distribution,* and *heterogeneity* – directly pose serious technical and sociological challenges that lie at the core of our research proposal. However, we would be remiss not to consider economic issues as well; to be a reality, the Digital Library must provide mechanisms for rewarding those who produce, collect and edit information, and for allocating costs appropriately to those who benefit from the services. In particular, it must provide mechanisms for safeguarding intellectual property, and for charging for information services. Thus, *economics* will be a significant additional strand in this proposal.

II.A The Information Bus and Services

The central goal of the Stanford Integrated Digital Library Project, one that permeates the entire project, is to define an *Information Bus* that ties together the materials, services, and users of information. The Information Bus consists of the basic concepts, language, and protocols which together suffice to describe all activities in the Digital Library.

A good way to explain the role of the Information Bus is by analogy with the developments in the software industry over the past decade. As the industry has matured, users have come to expect a large number of features as standard (such as a graphical user's interface, or GUI), making for increasingly complex applications. The response in industry has been to offload some of the burden from the application developer by standardizing some features (such as window management) and incorporating them into the operating system. More recently, in recognition that even this leaves much to the application writer, efforts have been put into allowing applications to communicate, whether to simply exchange data or to activate a procedure implemented remotely (we give specific examples in Section II.B.1). These allow heterogeneous applications to interoperate, a feature that is particularly important in a networked environment.

Similarly, we believe that internetworked digital libraries require that common functionality be standardized into a common software base, in our case the Information Bus. In fact, we already see the beginning of this trend with the emergence of common network protocols such as FTP, SGML, HTML, Gopher, and MIME. These protocols do facilitate the sharing of information, as long as one develops a suite of "translators" to convert between the diverse formats and protocols. This is illustrated in Figure II-2.

However, while clearly useful, such a structure still provides help at a very low-level as compared to the concerns of an information provider or consumer. The Information Bus aims to support higher level operations, such as querying about the availability of information or its format, and executing a transaction that includes exchange of information and (possibly) financial obligations.

One way to visualize what we propose is shown in Figure II-3. We caution that this is a *conceptual architecture*; that is, the components shown are logical rather than physical. Some of the components shown may run on the same computer, while others will be implemented as clients and servers distributed over a network.

As in the other figures, the Information Sources (IS) represent the underlying information resources to be made accessible. Above the ISs lies the Information Bus, the common set of

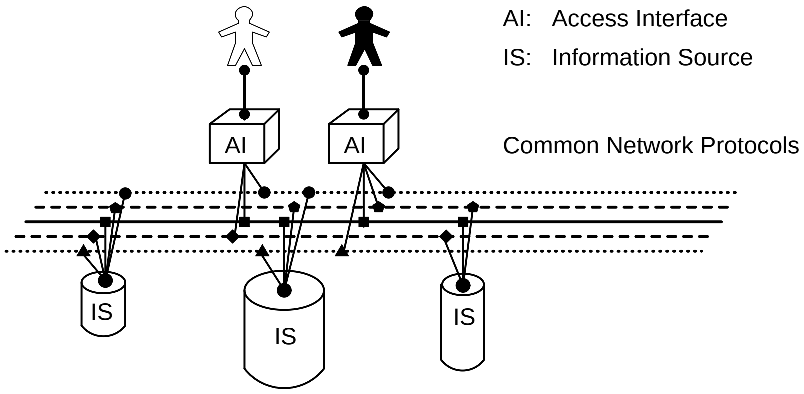

Figure II-2: Trend towards common network protocols

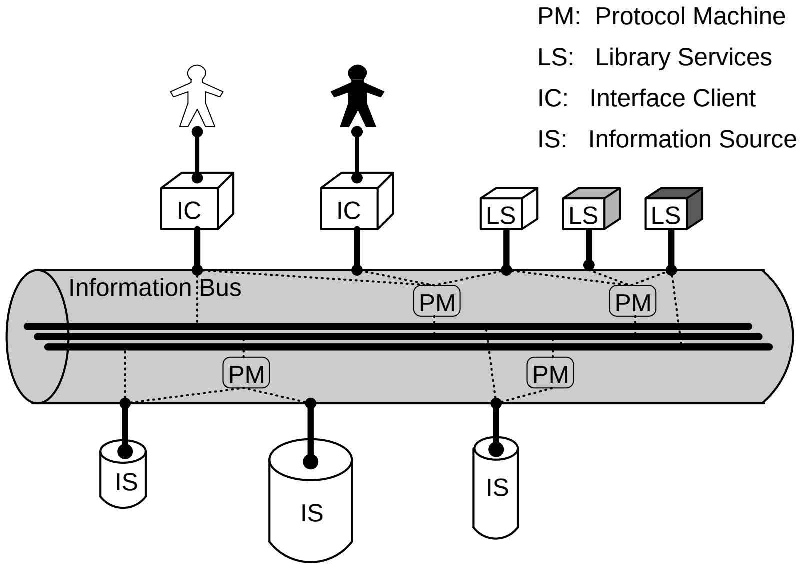

Figure II-3: The Information Bus

protocols (including an Application Programming Interface – API) and representation models for information items that allow the rest of the components to interoperate. The Information Bus is built on top of the existing protocols that were illustrated in Figure II-2. The Protocol Machines (PM) shown within the Information Bus embody the "intelligence" needed to operate the bus, as will be illustrated shortly.

End users access the Digital Library through the Interface Clients (IC). These client programs are components of the overall system, which allow users to see and manipulate the information items in the Digital Library. They make use of higher-level abstractions and visualizations (including both generic and domain-specific) to translate user needs into requests for information resources and activities on the Information Bus, and to translate results into a form that makes sense in the context of the user's task.

The final component of this logical architecture is a set of Library Services (LS), that include such things as:

- Finding Services: to locate either information sources and information resource items within them that are relevant to a user's particular interest.

- Translation Services: that may transform a document or other information resource from one representation into another.
- Meta-Information Services: that can describe the format of information resources, the meaning of various information fields, the use of protocols, and so on.
- Financial Services: for charging users for services and for paying providers. These could include a "bank" of the Digital Library.
- Copy Detection Services: for example, where information items can be registered and others can be sent for checking for similarity against previously registered ones.
- Publishing Services: that manage the process of creating an information resource, registering it, and disseminating it in a way that facilitates getting it to people to whom it will be most relevant.
- Authentication Services: that manage access and other information-manipulation rights to individuals and programs.
- Mediator Services: that combine and fuse information from a variety of other sources.
- Automated personal information services: end-user-programmable modules that actively find, filter, organize, mutate, and distribute information.

We envision a collection of such high-level services, all implemented to communicate uniformly through the Information Bus primitives, working in concert with one another. For example, a Personal Information Assistant (one of the services we propose to design) might specify the type of information required, the set of information sources to be queried, the desired presentation format, the timing and priority of the required service, the desired levels of security, payment terms, and other elements of the Information Bus. To execute this query, the personal information assistant (PIA) might require the services of a mediator for, say, Computer Science documents. To the PIA, the mediator appears as another information source that can return Computer Science documents of interest. In fact, when the mediator receives a query, it calls on a finding service to assist it. After that, the mediator might forward the query to what it considers to be the two best information sources for the type of request. Once it gets documents, the mediator checks for duplicates, perhaps not forwarding to the PIA documents that are too old. At the end of the transaction, the mediator sends a bill to the financial service that is representing the end-user (who was identified and authenticated at the beginning of the transaction, using an authentication service.)

As the Information Bus evolves, the types of operations it can perform will become higher level. In the example of the previous paragraph, each service explicitly called on other services. For example, the mediator directly called on an information finding service and a financial service. However, we plan to incorporate into the Information Bus high level, library-specific functions such as "obtain an item meeting some specification." The Protocol Machines (PMs) within the bus would then handle all the finding, billing, translation work. Thus, the Information Bus will hide many of the low level complications that arise from heterogeneity, distribution, and access and resource controls.

We believe that a sophisticated and library-specific Information Bus can greatly simplify the job of service providers and interface writers, and hence will be a key component of the Digital Library. We note that while the Information Bus can be viewed as a "standard" for interconnection, our interest is not to engage in the political activities that are typical of industrial standard setting. Rather, we aim to develop a conceptual model and architecture that will inform such commercial standard-setting efforts, which are sure to take place as the National Information Infrastructure takes shape.

The Digital Libraries vision we have sketched clearly requires a substantial research effort. In Section II.B we describe the fundamental problems that must be solved, both for the Information Bus and the key library services that must accompany it. Section II.C presents the experimental SIDLP Testbed we propose to implement to validate and evaluate our concepts, while the expected accomplishments of this effort are outlined in Section II.D. The tasks to be undertaken and the role of the participating institutions is discussed in Section III.

II.B. Research Problems

This section discusses the research problems that will be addressed by this project, and suggests some directions and approaches. Although many of the problems are interrelated, we break our discussion into five sections. The first (II.B.1) covers problems related to Information Bus modeling and infrastructure. The remaining sections cover issues from four different perspectives, that of the interface clients (II.B.2), that of information finding services (II.B.3), that of the economic (in a broad sense) problems of interest to information providers (II.B.4), and that of service creators using agents (II.B.5).

II.B.1. Information Bus Infrastructure and Models

In order to provide the kinds of information services needed in a widely distributed information space, we need to create conceptual models that will serve as the basis for the protocols, service definitions, and interfaces that unify the different objects and formats in the distributed Digital Library. In particular, we need models that provide the necessary distinctions and properties for describing a variety of information objects (such as documents, images, indices, collections, etc.) in terms of a number of basic dimensions (such as document identity and versioning, authority, ownership, access rights, summarizing information, etc.).

II.B.1.1. Definition of the problem area

In order to illustrate the importance of these shared models, it is useful to draw an analogy with a simpler case, that of ordinary file systems. It was realized early in the history of computing that in order to handle the diversity of applications and storage implementations there had to be an intermediate level of abstraction for files and file collections, at which the concepts could be unified. Different interfaces could map from the concepts of users or specific applications programs onto these abstractions, with different "drivers" to implement them with a variety of underlying mechanisms. The uniformity of abstraction made it possible (in the ideal case) for the user or application developer to operate with a single conceptual model that covers in a unified way a collection of different application interfaces. A user or application needs to know whether a file is accessible or not, but not (in general) where it is stored on tracks fragmented across the disk. The systems software, in turn, maps objects and operations onto the details of disk tracks, network protocols, character set byte encodings, and the like.

At a higher level we want to have abstractions that deal with other aspects of information objects that cut across the specifics of different file system implementations, and that apply beyond static file stores to cover services such as searching, translation and visualization of information. For example a client accessing a document server might want to know the size or authenticity or cost of a document before attempting to transfer it, or a visualizing interface might want to provide a summary or "thumbnail" of a large item for the user to examine before deciding whether to retrieve it, or a search mechanism might want to get some kind of profile of an item's contents for use in matching a query, without full retrieval.

With networked information across the diversity of the Internet, the problem is compounded. The system needs to be able to provide access to information resources that are not under the control (physical, or administrative) of a party who puts together "the system." It is critical to deal gracefully with widely varied resources, rather than counting on the ability to legislate conformity to a single or small number of uniform implementations. None of the existing protocols and systems address the range of issues that are important in a *global* library. Concerns such as authenticity-of-information, cost, completeness-of-collection, and access time become immediate problems only when you look at things at a "library" scale, as opposed to a small or uniform collection.

When we say "none of the existing protocols and systems," we include Z39.50, GAIA, World-Wide Web, Dialog, OCLC, or any of the current commercial systems for workgroup or enterprise document management. Each of these systems deals with some aspects of information about information objects in its own way. Often, those who make information sources available know things about their repositories that they have no formal language for expressing.

We will also need to include meta-information abstractions for non-traditional digital media (audio, video, image,...), where the raw information content is not directly amenable to operations such as summarizing, browsing and searching. These are in an even more primitive stage than those for text documents and databases.

II.B.1.2. State of the art

There is already a rich set of ideas within the community to draw on. The opportunity at this point is to move upward from specialization to generality. The Information Bus will provide a uniform descriptive space that includes both the currently widespread and the newly emerging forms of digital media and will provide protocols for describing and communicating resource objects in this highly heterogeneous space. Since our goal is to integrate information across a wide spectrum, a number of different kinds of previous work are relevant. Overviews and references to a number of the systems mentioned below are included in [38, 27, 46, 10, 4]

Integrated information repositories A number of projects have set out to achieve the functionality of a general Information Bus within the architecture of a single server or distributed collection of dedicated servers. These include the MIT Semantic File System [17], Xerox PARC's System 33 [43, 41] and Tapestry [18], IBM's Melampus/RUFUS [30], and others. We can build on much of the detailed work they have done to provide uniform meta-information for documents and other information objects. The difference in our work is that we are not planning to build a unified repository, but rather to provide a networked infrastructure that makes it possible to carry out uniform information access and operations on the distributed information and network of machines that form the distributed Digital Library.

Network protocols Network protocols already in common use include base level networking protocols, such as IP/TCP, UUCP, X.25, etc. on which the others are built. We are not concerned with this level, and will use them (along with others that are emerging, such as TeleScript) in the standard way as a base on which to implement higher level protocols. On top of this level are a number of structure-specific protocols, such as mail standards (RFC822, MIME, SMTP, X.400 ...), file information and transfer protocols, (FTP, Gopher, Gopher+, HTTP, IAFA, ...), bulletin board and news protocols, (NNTP, Listserv, ...), protocols for directories of people, organizations, etc. (Whois, Finger, Ph, X.500,...). One of our goals is to generalize the shared underlying concepts that cut across these, and unify them in the information object descriptions that flow on the Information Bus. We would not expect to replace existing protocols completely, but rather to produce appropriate gateways and translators so that servers that use these formats can be integrated with the Digital Library collection of clients and services. Also, as protocols begin to bridge the gap between networking and interprocess communication we want to be able to apply the Information Bus abstractions directly to communication using system-specific protocols, such as OLE, DDE, AppleEvents, object-oriented standards (Corba, OMG, ...), etc.

Document structure standards A number of protocols deal with the structure of documents, both traditional and networked. Some deal with conceptual document structure (SGML, ODA, Xerox DSA...), others with printing and exchange formats (Postscript, PDF, RTF, ...), and others specifically with hypertext (HTML, Hytime, UR*, ...). Some of the concepts in these protocols are meta-information about the documents, and would be included in our work. Most of the detail, though, is concerned with visual formatting and domain-dependent content-structuring, which would not be our focus.

Database query and information retrieval protocols There are a number of languages and protocols for structured query of databases of various kinds, including relational (SQL) and bibliographic (Z39.50). In general, our concern is with protocols at a higher level of granularity – the unit is some kind of document (though generally not text) rather than a data element. But there is a spectrum of granularity in on-line materials, and the relationship between our protocols and conventional database protocols will be explored to develop appropriate mechanisms.

Communication structure protocols Many of the protocols mentioned above include a component that deals with the interplay of communication acts, or transactions, between client and server, as well as a component dealing with the content. Some higher level protocols try to separate out the communicative aspect, using logic-based protocols (KIF/KQML) or speech-act-based protocols (AOP, ActionWorkflow, ...). The Information Bus will include a high-level protocol for transaction/interaction, which will be based on a theoretical account of the structure of interactions.

Domain-specific protocols and standards A number of projects have set out to provide a unified structure of meta- information within a specific academic or scientific domain. This includes bibliographic categorization systems, medical information and nomenclature systems, newswire systems, software libraries, scientific collaboration testbeds, etc. Each of these has a significant body of detail related to the specific domain, along with more general protocols for supporting distributed access to diverse materials and meta-information about them. We will abstract and provide these more general services, using the existing systems as a basis for generalization, but will not attempt to integrate their specialized tools and environments into the Digital Library as part of this initial project.

II.B.1.3. Our approach to the problems

People who have worked on networking standards and protocols may initially find our goal of creating a uniform Information Bus to be overly ambitious. It has been hard to develop standards even for small subsets of this diversity, and the more things that need to be included, the more the complexity grows.

Our assertion is that by shifting the task to the larger and more general domain, we are forced to view it in a new way. There is an obvious analogy in the history of computing languages. Decades ago, frustration with the complexity of ever-expanding programs in machine language led to the development of concepts of higher-level programming languages: structured programming, modules and interfaces, object-oriented programming, etc. This enabled complexity to be partitioned into manageable modules and programmers could create and maintain much more complex programs.

In a similar sense, meta-information is a domain that requires new conceptual structuring. The language for expressing it will have two facets: a core language that serves as the structural "glue," and an extensible collection of *conceptual frames,* each dealing with one aspect of information, such as summarization, ownership, or access rights.

For example, consider the concept of *owning* a piece of information. Many existing file systems have some representation of the owner of a file, which may be separate from the creator or maintainer. But these differ from system to system, and do not address the kind of ownership that will become more central as we move beyond public-domain information. In general, information produced in the course of work in a company is owned by the company, regardless of the individual who created it or put it onto the system. Copyrighted material is owned by the copyright holder, but may have been licensed or transferred to other owners who provide it for a fee.

Looking more deeply, we find a complex bundle of different notions of ownership, relevant to different kinds of actions that people take (writing, putting on line, copyrighting, selling,...). We can work out a set of distinctions that keeps these appropriately separate, and is based on a clear semantics of who has done what (or can do what) with the information. These in turn can be formalized into a set of specific attributes that can be used for describing and operating on information items. This resulting structure is not a part of the core language, but forms a conceptual module of the Information Bus protocol.

Each of today's network standards can be seen as taking a cut through a bundle of conceptual domains, using its own structural core to express a subset of issues from the conceptual areas that the designers deemed most salient. By partitioning the problem into separable modules we provide the basis for working on parts that are most directly relevant and useful, and to take advantage of work done by others, just as programmers today take advantage of libraries of shared modules or objects in doing their programming.

In many cases there are already existing standards that might be adopted, simplified, or combined. In other cases there are specific organizations or informal communities of interest that have taken on the standardization task, and would be involved in a discourse about it. The goal here is not to legislate "the right" standard, but to create an environment in which there are social and commercial pressures towards consensus, and to provide mechanisms (in the spirit of widely distributed systems) that are able to cope with multiple overlapping standards and provide a uniform framework for services such as conversion, updating, translation, and storage. There are a number of interesting examples of this kind of activity already happening in the networking world, and experience suggests that whenever standards can provide obvious added value, people will willingly adopt them and contribute to their evolution. One of the uses of our testbed server will be to provide the information services that support this activity.

In developing the conceptual frames, we will make use of concepts from our own experience with database protocols [49, 19, 55, 5], agent protocols [47], knowledge representation languages [3], information access protocols [33], and network information protocols being developed by our commercial partners, EIT, Dialog, O'Reilly, WAIS and Xerox. Our goal is to find a viable tradeoff between complexity and functionality, through developing an understanding of what constraints, tradeoffs, and solutions are appropriate to networked information of the kind that will populate the collection we develop as the basis for our Digital Library.

II.B.1.4. Summary

The development of protocols will proceed from three main perspectives: the information user perspective, the information provider perspective, and the service-creator perspective. The thing that brings these together is a shared set of underlying concepts and representations for the different dimensions of description for information sources, objects, and actions. The development of this conceptual framework will cross the boundaries of the other project sections and will be an evolutionary process in which early versions are the basis for implementations of specific interfaces or services, and experience with applying them will lead to refining and extending the conceptual models.

II.B.2. Client Information Interfaces

One of the primary motivations for developing the architecture of the Information Bus is the desire to create more powerful user interfaces for operating in widely distributed information spaces. We will draw on current techniques for information presentation and visualization, and will extend and adapt them for the materials of the Digital Library.

II.B.2.1. Definition of the problem area

The production of successful interfaces will require work in three dimensions: Perceptual modalities, Conceptual spaces, and Interaction structures.

Perceptual modalities The human perceptual system is geared to dealing with large quantities of simultaneous information, but only when the information has a natural structure that maps onto experience. For example we can spot a familiar face in a crowd or pick out a shooting star in the sky without conscious effort. As we begin to build the distributed Digital Library, with its large and diverse collections of information, we want to enable searching and browsing the information space by tapping into these abilities. We need to provide interfaces that map from information characteristics onto spatial and visual properties that lead to effective human processing.

Conceptual spaces The accessibility of information depends not just on whether it can be identified and retrieved, but also on its integration into a broader context. We are interested in making use of this within a higher-level structuring that correlates information with its cognitive and operational relevance. To do this, we need to take not only the logical content of information but also the social/organizational structure as an explicit component of structuring the information space. As the pure volume of information on the net explodes, it becomes more critical to have explicit ways of asking "Where did this come from?" "For what purposes?" "With what assumptions about the audience?" "From whose point of view?" etc.

Interaction Structures In other areas of this project, such as work on information finding and agents (Sections II.B.3, II.B.5), we will be developing devices that can dynamically combine and integrate information. In our work on user interfaces, we will complement this with another capacity: the ability of people to deliberately create and share information structures that refer to and add value to other information. The Information Bus architecture will make it possible to build these information structures uniformly across the SIDLP Testbed.

II.B.2.2. State of the art

The emerging tools for network navigation clearly show some of the possibilities and are also limited in several ways:

- Many existing tools, such as WAIS, Gopher, FTP and their derivatives, require the user to understand the structure and much of the detail of the heterogeneous information sources. This is because they are fundamentally access-oriented, rather than content-oriented. For example, the first question the user must ask is "What Gopher site (or WAIS source, or database, or ...) should I access?" Cross-source catalogs such as Archie and Veronica attempt to overcome these limitations, but currently provide a minimum of structure (text search over global lists of file names or short phrases).
- Other tools, such as the World-Wide Web browsers provide a "seamless" space, but with a minimal ability to provide any kind of overview or global structure. The user is peering through a local window, jumping from place to place in the information space without being able to visualize it in larger terms.
- There have been many interesting experiments in visualization of large bodies of information, but these have not been actually linked into the net, and often suffer from the toy nature of their data.

A number of dimensions can be brought to bear on making information spaces more visualizable. The first is the notion of space itself, mapping from the abstraction of information representation onto a two or three dimensional space that can be displayed and navigated using familiar spatial operations (move, zoom, etc.). Along with this, there are different kinds of topology and geometry that can give a sense of effectively manipulating a very large space, while using a limited display medium such as a fixed-size screen. This will be especially important in the world of ubiquitous computing, where displays range from wall-sized to wrist-watch size, and include aural as well as visual representations.

Previous work along these lines include the Information Visualizer project at Xerox PARC [6], which has developed interfaces such as the rooms browser, cone tree, perspective wall, and document lens. Unpublished work in the Project on People, Computers, and Design at Stanford has included the explorations of fish-eye views for networks, which has been developed elsewhere [14]. Work at Interval [50] has included the visualization of information spaces, such as incoming items in a news group, using a three-dimensional representation based on a flowing stream of objects in force fields.

In the area of conceptual spaces, there has been research on the structuring of hypertext [21, 37], but it has not yet been applied to the kind of structures that appear in the distributed heterogeneous digital library.

One of the most interesting phenomena that have emerged as the Internet evolves into a universal information web is the appearance of *value added services,* in which one party adds structure to information provided by others. The idea of allowing individuals to directly manipulate a shared digital space (without the aid of formal publishers) is quite old, and has been advocated for many years [11, 36]. On the current Internet, tools such as Archie, Veronica, Prospero and Gopher, represent the first attempts to allow users to structure information which is not owned by them [38, 46, 27]. to navigate through information from many other sites. Services such as mediators enable a user to ask for data without knowing on which of a number of systems it might reside. There will be an increasing traffic in value-added information, such as annotations, commentaries, filters, and collections of references to other items.

II.B.2.3. Our approach to the problems

In this project we will extend existing information interfaces and develop others, using the Information Bus architecture as a way of integrating multiple sources of data into a common interface. The framework provided by the SIDLP Testbed will offer an opportunity to create and test interfaces that make use of information about information in a general, uniform way. Through the common language of meta-information on the Information Bus, we can develop multiple interfaces for diverse implementations, based on a shared underlying conceptual structure and collection of conceptual frames.

Visualization and perceptual modalities

In an interface based on visualization, there are three basic components in presenting a field of information to the viewer.

- First, the raw information needs to be processed to create appropriate information characteristics (content categorization, summarizations, origin, date, availability, relevance, etc.).
- These are then mapped in a systematic way onto visual characteristics (size, labeling, 2D or 3D position, orientation, speed, color, etc.). This mapping may be dynamic (e.g., labeling might change with size or apparent distance).
- Finally the application hardware maps these characteristics onto the user's perceptual space. In doing this we want to go beyond current ideas of graphic interfaces. There is a need for interfaces that can map from information characteristics onto a range of human perceptual skills, so that people with different perceptual abilities and limitations (e.g., vision-disabled users) can access information. By starting with a uniform framework which separates visualization from the underlying content and services, we can provide multi-modality interfaces that provide for alternative instantiations of the underlying structures.

The first phase of the work will be to bring together in a common environment a collection of existing advanced visualization techniques, including those in our own groups and others in the literature referred to above.

The next phase, which will go along with the development of the Information Bus protocol, will be to determine the right abstractions for coupling these interfaces to our collection of sources, and to design the appropriate extractors for generating the necessary level of description from the actual information objects and directories.

The third phase will be to implement generic back end components to the visualization interfaces that can create the needed data structures dynamically, using the Information Bus as the access path for the information.

When these implementations are built, we will begin testing them with users of the information collection. By that time we will have developed the collection and begun to work with the user community. Based on this we will choose appropriate test sets of information sources and corresponding visualizations. This will, of course, be an iterative process in which experience leads to redesign of the interfaces and, when necessary, the Information Bus architecture that supports them.

By the last year of the project, we will be able to identify a stable collection of visualizers and integrate them into a common user interface framework, providing users with flexible visual access to browse and search the full collection of our SIDLP Testbed.

Designing conceptual spaces

The literature on hypertext [37], describes the problems that inevitably arise when a user has the power to move flexibly among items in a large information space. The user is often described as being "lost in hyperspace." When working with a particular piece of information or collection of pieces, he or she does not have a sufficient sense of context: where the information is from, what it is for, and how it fits in with what he or she has already seen and might want to see. Much of the discussion of hypertext uses the metaphor of physical space, with its motions, maps, and orientations. As we said in the previous section, this kind of concrete metaphor is an important step towards making the abstract space understandable and accessible to human beings, who have evolved and learned to work in physical spaces. But the mapping from abstract information space to any physical embodiment (even a non-Euclidean topological one) is not obvious.

The sense of physical place is based on some fundamental concepts:

Accessibility When you are "in" a place, you are by definition not in some other place. Places are created by the differential accessibility of things that are in and those that are out. There may be strong boundaries (like walls) or graded accessibility differences (like distance). Any notion of where you are has to be based on these differences. Any notion of navigation is grounded in the accessibility relationships of the space.

Boundaries In order to structure spaces usefully, we create boundaries. You are inside or outside the house, in a particular city, state, or nation, in my office or yours. By constructing boundaries that explicitly facilitate or hinder access, we give people the possibility to structure their materials in an effective way.

Hierarchies and granularity Boundaries can be nested, as with rooms, in a house, in a city, in a state,..., where the contents of one bounded area are included within the bounds of its immediate container, and so on transitively.

Point of view In some sense, where you are depends on who you are. To take a simple case, someone who knows the political customs of the community can be in one city or another, while a naive child simply sees the local street geography as gradations of distance. A bicyclist has a very different effective map of a campus than someone who drives. In creating spaces, and especially in creating maps of these spaces, it is critical to understand the different points of view that might be brought to the same overall territory.

These concepts all have straightforward interpretations in the physical world and the actions we do in it, and need to be mapped onto the quite different conceptual areas that are appropriate for information:

Content/relevance This dimension underlies traditional notions of information retrieval. The information space is in some way generated by the mutual relevance of information items, either through constructed boundaries (e.g., hierarchical category schemes) or graded distance relationships (word frequency, similarity, relevance ranking, etc.)

Ownership Most of the actual geography in today's networked information space is based on ownership (which in itself is a poorly understood concept, as discussed in the section on conceptual models). A file is on the Stanford server or on the Xerox server, in my directory or yours. Although there are relationships that cut across these boundaries, they form the ever-present core skeleton of the entire space.

Intended audience In publication, the boundaries of information are often characterizable in terms of intended audience. The "collection of materials published in *Scientific American*" may be describable in terms of ownership or content, but the real boundary is based on the editors' judgments as to what a particular set of readers will be interested in. It may, for example, include physics articles and advertisements for technically sophisticated stereo systems. The print world has a highly articulated system of publishers, genres and titles, which enables a reader to step into a bookstore or magazine stand with a clear sense of how to find "something I'll be interested in."

Availability One of the key boundaries for printed material is who can get access to it. The physical accessibility of physical works provides a natural set of structures ("Is it in our library?"). With networks, the lowest level of availability is becoming extremely homogeneous. If you have a URL (Uniform Resource Locator) you can get the item it refers to. But the physical accessibility of print materials has always been correlated with the other social and economic dimensions. Now that it is separated in the digital library we need to build it back in. As the Internet of today (with its primarily public-domain materials) gives way to the Digital Library of tomorrow (with its wealth of providers, costs, rewards, and motivations) this dimension will become more critical.

Our goal is to create a generally applicable sense of "infoplace," which can be used as a basis for browsing and accessing the diverse materials in the Digital Library. It is unlikely that a single spatial metaphor will be appropriate across the breadth of materials and uses in the Digital Library, but it is also likely that interfaces based on a few thoughtfully designed metaphors can give tremendous leverage.

Our initial project is to extend the existing world of networked hypertext browsers, such as Mosaic and Viola to give the user a sense of place. It should be possible for the user of a browser to look at the screen at any time and answer "Where am I? Where does this belong?" Today, one finds a wide variety of *ad hoc* textual and graphic conventions used for this purpose. By having a theoretical basis for constructing information spaces, and by providing for this kind of information in the Information Bus architecture, we can unify these and bring them into the basic interface. We will design explicit consistent visualizations for distinctions in the information space (ownership, accessibility, audience, point of view...) and build them into extended versions of the network browser software.

Both O'Reilly Associates and EIT have been working in this area, as it impinges on their areas of interest in commercial publishing and commercial transaction. They will work with us by providing access to the browsers and interfaces they are developing, so that we can suggest and make modifications that add a sense of place. By building on top of existing sophisticated interfaces, we will be able to move quickly to the new research issues, rather than having to build a base of interface facilities from scratch. In further development as the project goes on, we will integrate this work with the information visualization work described above, so that we can apply more sophisticated perceptual activities to viewing, traversing, and working in the information spaces. This will mean connecting up the conceptual place distinctions with the visualization mechanisms.

Interaction structures

One of the key features of the Digital Library that distinguishes it from traditional print repositories is the ability of each user to provide additional materials that extend, structure, or elaborate existing materials. One part of this is the ability to publish individual information items. But it is not useful to simply put something in the Digital Library unless it can be found by those to whom it would be relevant.

There are two approaches to making things findable: Giving the finder better search tools (as described in Section II.B.3) and giving the provider better ways to structure the information and its place in the information space. This includes not only making it accessible, but also providing additional contextual information that allows users to determine its relevance and add their own value to it.

We can identify several related kinds of structuring:

Collections Traditionally, the creation and management of a file directory has been under the control of the producers of the information stored within it. But with systems such as Gopher, Prospero and Essence, this is no longer true. Many sites now offer sophisticated directory structures, most of whose links are to information produced and maintained by others. There are even attempts to provide subject-oriented structures to cover the entire body of materials on the net. There are important commercial and intellectual potentials for people to perform this kind of activity. It can be argued, in fact, that it is the core intellectual role of publishers, whose selectivity and advertising is much more important in the long run than their role in the mechanics of print production.

Annotations and commentaries People have always annotated materials for their own use and for use by others they work with. The penciled notes in the margins of books and memos are being replaced by on-line annotations, which leave the original material intact while flexibly providing further information linked to it. Information display applications, such as Acrobat, use the metaphor of the "Postit," allowing people to stick visual reminders onto an information display, and have them available for later use. Groupware applications, such as Lotus Notes, provide sophisticated facilities for users to comment on information and share their comments selectively. Just as the networked nature of the Digital Library leads to activities of collecting, it leads to activities of annotating and commenting (including extended commentary dialogs among a group of reader/writers). There has been some exploration of authoring tools for annotations in the World-Wide Web, but they have been quite *ad hoc*. In order to do better, we need to ground the mechanisms in a theoretical understanding of the communicative structure: the points of view and speech acts that stand behind the posting of a comment in a particular information community.

Queries and Filters In traditional uses of information retrieval, a query is an evanescent object. It is entered, produces a result, and goes away, while the result set stays around for further use. In some systems, (such as Metaphor's DIS system [29]) though, the queries are first class objects. The capsules (as they are called in DIS) are created, stored, re-edited, and passed around to colleagues because they performed a useful information task that can be re-used over and over again. Information filters are essentially capsulated queries. They provide a re-usable search specification that can be reapplied periodically or continually for SDI (Selective

Dissemination of Information), as discussed in Section II.B.3.2. Our work will include the structuring and maintenance of queries, working with WAIS Inc., which has been a leader in the evolution of network query standards. Our goal in this will be to develop a theoretically grounded practical basis for integrating a broad range of different kinds of queries, and making queries be "first class citizens" that can be saved, reused, modified, shared, and visualized in an information space.

The previous section described plans to extend and modify existing World-Wide Web browsers, such as Mosaic and Viola to add a sense of place. We also will include mechanisms, representations, and visualizations for building collection and annotation structures. The "postit model" is admirable in its simplicity, but has many shortcomings. It doesn't deal with underlying information that can change, with sharing and filtering the annotations, or with providing more global structuring than "stick it wherever you want on the page."

In the first phase we will review and examine a variety of annotating mechanisms, including those mentioned above, others explicitly designed for group work (e.g., For Comment), and those used in other hypertext systems. In doing this we will be applying the basics of speech act theory and the language/action perspective [54], to distinguish and formalize the underlying operations and intentions.

Based on the theoretical framework that we can build from this analysis, we will develop a set of designs for potential mechanisms and interfaces, and explore them through the use of standard interface design methods, such as scenarios, mockups and prototypes.

On the basis of what is learned in the design studies, we will specify an information-structuring interface that can be applied, using the Information Bus structures, to the full spectrum of materials in our collection.

II.B.2.4. Summary

The Information Bus architecture will make it possible to develop interface prototypes that focus on a particular element of information structure and visualization, which can be applied across the diversity of materials in the collection. By applying new visualization and structuring techniques to realistic collections of materials and practical uses, we will be able to refine and develop them as well as devising new techniques.

II.B.3 Information Finding Services

The problem of information finding in a Digital Library is one of matchmaking: On the one hand we have a user's query, and on the other we have a document, or set of documents, or parts of documents that are of interest to the user at this moment. The problem is a very challenging one due to three main factors:

- Scale: The needle in the haystack problem. There are huge numbers of documents, distributed over many sites.
- Heterogeneity: The tower of Babel problem. Each site represents documents in different ways.
- Query Expressiveness: The pidgin problem. The query language is not expressive enough to specify the user's interests.

In this section we do not address the third factor. We assume that Interface Clients (IC) translate the user's query into a relatively simple request for information, based on keywords or terms of interest. This request can either be a Boolean query (e.g., I need documents with terms "cat" and "dog") or a vector model query (e.g., I need documents with 2 or more of the terms "cat," "dog," or "bat"). The IC can post-process the retrieved documents to make sure they match the user's original query or intent.

After briefly reviewing the state of the art, we focus our discussion on three research topics:

1. The implementation of a network finding service,
2. Searches for candidate sources, and
3. Distributing the server.

This is not an exhaustive list of important research areas, only the ones we directly plan to address in this project.

II.B.3.1 Prior work

There is a substantial body of research on Information Retrieval [44, 45] that addresses languages and models for user queries, and their effectiveness. A variety of index structures have been studied, including inverted lists [53] and signature files [13]. Recently, there has been work on how to distribute such indexes to deal with very large collections, e.g., [49]. Techniques from the Database Systems area [53] may also be relevant for searching through structured data such as that found in catalogues and data dictionaries (see Section II.B.1).

Current network information services such as World-Wide Web (WWW) [2] and the Wide Area Information System (WAIS) [24] provide content-based indexes, but they typically span a single site or collection. In WAIS, there is a "database of databases" that describes the contents of document collections. However, these descriptions are manually produced, tend to be out of date, and are so general that it is easy to miss collections that may be of interest. There have also been some recent efforts to provide more powerful servers that can suggest sites of interest for a given query. In [19] we describe one such possibility and provide a more complete survey of work in this area. Work in the resource discovery field is also relevant [38], although it generally does not address content-based searches.

II.B.3.2 A network finding service

The goal of the information finding server we plan to implement is to guide users to relevant documents or to sites that contain information of interest. In addition, it will be a repository of meta-information that can describe the formats of information, the meaning of various components within documents, and the use of protocols. In particular, the server will receive the following types of requests:

1. Content-based documents queries. Based on a description of the contents of the desired document, either
   (a) Return matching documents, or
   (b) On a continuing basis, return newly created documents that match. This is known as selective dissemination of information (SDI), continuous queries, or alerts [48, 35, 42].

2. Content-based site queries. Similar to document queries, except that server returns the names of sites that may have matching documents. These may also be continuous SDI queries.

3. Service queries. The objective here is to find services (as opposed to documents). For instance, a user may be interested in locating a service capable of translating a document from one format into another.

4. Meta-information queries. The answers describe formats or protocols in use in the Digital Library.

We are especially interested in continuous SDI queries (in all categories above) because we feel it is an extremely powerful mechanism. Many of us already use it in our daily lives (e.g., magazine subscriptions, mailing lists) and it needs to be incorporated into a Digital Library. Furthermore, we have substantial experience in this area. Researchers at PARC were one of the first to study continuous queries, in the context of a successfully implemented electronic mail system called Tapestry [48]. Researchers at Stanford have more recently built an SDI server for NetNews articles and Computer Science Technical Reports [55]. Both of these systems are limited in many ways, but can form the basis for our research.

Beyond the implementation of SDI services, there are many additional challenges that need to be addressed before an information finding server can be implemented. Here we briefly illustrate some of them. In Section II.B.3.3 we discuss site queries in more detail; in Section II.B.3.4 we describe how the server can be distributed.

The server must be able to deal with heterogeneity. For example, the arriving queries may be expressed in different way, perhaps as a Boolean query, or as a vector of terms and weights. The terms may refer to desired word occurrences in the target, or to concepts in the target. The indexes available to the server may also differ. For instance, the server may have a conventional full text index for the documents in one source, and abstract and title index for a second, and English description of the topics covered in a third, a list of the most frequently occurring terms in a fourth, and no information on a fifth. The server must be able to evaluate a query against all these types of indexes and somehow combine the answers. For sources like the fifth one, it may distribute the query directly to them, so they may evaluate it.

Another critical problem to study is whether the server will be *pull* or *push* or both. A pull server is responsible for polling the potential information sources for index and meta-information; a push server assumes it can subscribe to the information. In either case, protocols have to be devised for determining which sources will be covered by the server, and for then interacting with them.

Even though our primary interest is not video nor audio documents, we still need to consider them in the design of the finding server. Specifically, the server should be extensible so that it can handle "user (or source) defined predicates." For example, a user may be searching for pictures that are "similar" to an existing one. The user provides a signature of the existing document, which is compared to those of candidate documents, using a "similarity" function.

II.B.3.3 Searching for candidate sources

Given the magnitude of the Digital Library of the future, we think that one of the primary functions of an information finding service is to locate *potential* sources of relevant information, as opposed to identifying the individual documents of interest. We would like to explore strategies that are "approximate," i.e., that may miss (with low probability) some sites of interest. We believe that such strategies can be more practical in a large scale, heterogeneous environment, while still giving users excellent guidance for exploring the available resources.

There are a variety of strategies for solving approximate queries. Here we will illustrate one that we have been recently studying [19]. (Others are surveyed in that report.) The basic idea is that each potential source extracts a "histogram" of its term or word occurrences. For example, a computer science library could report that the word "relational" occurs 12,345 times, the word "databases" 25,548 times, and so on. This information is orders of magnitude smaller than a full index since it does not report the identities of the documents that have the terms.

However, these histograms can still provide very useful information regarding what sites may have relevant documents. For example. say we are looking for documents with terms "relational" and "database" (a Boolean query). Assume we have the following information:

Library A:	occurrences of "relational":	10,000
	occurrences of "databases":	5,000
Library B:	occurrences of "relational":	10,000
	occurrences of "databases":	0
Library C:	occurrences of "relational":	50,000
	occurrences of "databases":	75,000

Clearly, Library B would not be a good place to search since we know no documents would match our query. We also know that both A and could have matching documents. We do not know for sure if they do, since the occurrences of "relational" and "databases" could be in different documents. However, it seems likely that they do, and as a matter of fact, it seems much more likely that Library C will have more matching documents since it has more occurrences of the desired terms. If we wanted to provide hints as to where to search, it would be reasonable to suggest Library C as the first place to search, and Library B as the second.

We have named this approximate search technique based on term frequencies as GlOSS. We have evaluated GlOSS (and several functions to weight the likelihood that a site matches a query), based on a trace of real user's queries and databases available at Stanford, and the results are very encouraging, showing that success rates of 90% are possible [19].

We propose to continue to investigate strategies such as GlOSS. In particular, GlOSS is currently limited to Boolean queries. However, we believe it can be extended to handle vector model queries, based on the same term frequencies. We also think GlOSS can be extended to handle continuous queries. In this case, GlOSS would yield a list of sites that may have new documents of interest (as opposed to the documents themselves). The list could be returned to the user, or alternatively, the information finding server could issue the queries directly on behalf of the user. Finally, we believe that another fruitful research direction is to add financial knowledge to GlOSS (see Section II.B.4). That is, a user may not only want to be guided to good sources, he may also wish to locate inexpensive sources. Thus, the weighting functions that GlOSS uses could incorporate cost information.

II.B.3.4 Distributing the server

Although we have referred to an information finding server, clearly, to be scalable, the server must be distributed. We plan to study how to efficiently distribute the server. To illustrate the issues and the types of solutions envisioned, consider the problem of providing an SDI continuous query service, where documents are generated at say 600 sources. One solution could be to have, say, 6 servers, and to partition the sources so that each server is responsible for 100 sources. Each source then notifies its assigned server of newly created documents. To subscribe, a user request must be posted at all six servers. The disadvantage is that each server must see *all* user requests.

Furthermore, if a server is down or unavailable, the user will miss (or get late) documents from 100 sources.

A different solution would be to form a grid of servers, say of 3 rows by 2 columns. Let us call the servers in the first row A_1 and A_2; those in the second B_1 and B_2, and those in the third C_1, C_2. When a source generates a new document, it notifies any one row of the grid. For example, it may notify B_1 and B_2. A user request for information is posted at one column, for example, at A_1, B_2, and C_1. This guarantees that the match is made in one server. Each server sees only one half of the user requests, but must now see one third of the documents (as opposed to one sixth with the previous scheme). This may provide a better balance, in cases where there are many user requests. Furthermore, it provides higher availability, since a user can post a request even if some of the servers are down. However, some documents may still be missed due to a failure.

We have done some very preliminary research in this area, and there are a large number of options for structuring the servers. Some offer much better availability than the simple grid, but may generate more network traffic. We plan to investigate the options in detail, and evaluate their applicability to our SIDLP Testbed system (Section II.C).

In addition to distributing the server itself, there is the more general problem of providing access to a Digital Library in a mobile, dynamic, distributed environment. The mobile stations may have users requesting information, or documents generated by a traveler. Researchers at PARC have developed a testbed for "ubiquitous computing" [51] that we plan to use to explore mobility and high availability issues. Specifically, we plan to study replication strategies for documents, so that not only they are highly available at all times, but are also never lost due to a disaster (e.g., a fire). Similarly, we propose to develop algorithms for *eventual* query processing [8]. The idea is that a search must not assume that all servers and sources are operational and connected at the same time. Instead, a search should propagate to sites as they become available, eventually visiting all the necessary ones and eventually sending the results to the interested user. Such eventual queries may be the only reasonable solution for large exhaustive searches.

II.B.3.5 Summary

Information finding is one of the central problems of a Digital Library. Users must be able to locate relevant information, even when sources are heterogeneous, distributed, and possibly disconnected. We will implement the SIDLP Testbed a distributed finding service that meets these requirements, and on it will evaluate a variety of strategies, including ones for continuous queries, approximate queries, and eventual queries. We will also incorporate the protocols for accessing information finding services into the Information Bus.

II.B.4 Information Provider Perspective

Digital Libraries are a concrete possibility today because of many technological advances in areas such as storage and processor technology, networks, database systems, scanning systems, and user interfaces. In many aspects, building a Digital Library today is just a matter of "doing it." However, there is a real danger that such a Digital Library will have relatively few documents of interest, not because the computers cannot handle more, but because information providers will not be willing to place their documents in the library [52].

The reason for this danger is that the electronic medium makes it much easier to illegally copy and distribute information. If an information provider gives a document to a customer, the customer can easily distribute it on a large mailing list or can post it on a bulletin board. The danger of illegal copies is not new, of course; however, it is much more time consuming to reproduce and distribute paper or videotape copies. Thus, to make *populated* Digital Libraries a reality, we need to develop technology for safeguarding intellectual property. In particular, there are three types of technology we plan to investigate:

1. For protecting intellectual property,

2. For detecting violations to intellectual property rights, and

3. For performing financial transactions.

Each of these topics in covered in subsections II.B.4.2 through II.B.4.4. In addition to these technology topics, we plan to study some of the social issues related to intellectual property and charging in a Digital Library. The focus will be on evaluating user patterns, actual costs, market demand and elasticity, using ACM's publications as a test case. This is discussed in Section II.B.4.5.

II.B.4.1 Prior work

Current technology does not strike a good balance between protecting the owners of intellectual property and giving access to those who need the information. At one extreme are the open sources on the Internet, where everything is free, but the ratio of valuable to useless documents tends to be low. At the other extreme are closed systems, such as the one that the IEEE currently uses to distribute is papers in CD-ROM. This a completely stand-alone system where users can look for specific articles, view them, and print them, but cannot move any data in electronic form out of the system, and cannot add any of his or her data. Clearly, we do not want the Digital Library of the future to be at either extreme. One would like to have access to all documents under a single integrated, open system, but at the same time, providers want to have economic incentives for offering their information.

Encryption is one well-studied way to protect digital information [40]. It can be useful for safely sending a document to a customer, but once the customer decrypts it, there is nothing to stop him from further distributing it. However, it can form the basis for some of the schemes we will discuss later. (See additional references later on in this section.)

Banking institutions worldwide have used electronic funds transfers for years (e.g., SWIFT Network). Some of this technology could be used for charging in a Digital Library; however, substantial research is needed to make it more secure than it currently is. Furthermore, current banking networks are closed; we need to investigate how they could be used in conjunction with a very open network.

Finally, the articles in [15, 23, 31] discuss a variety of topics related to intellectual property in a Digital Library.

II.B.4.2 Protecting intellectual property

How can we ensure that a document is only seen and used by a person who is authorized (e.g., has paid) to see it? Let us illustrate the possibilities and the problems by suggesting two particular techniques.

The first technique is based on the notion of a *secure printer*. Such a printer is sealed and cannot be opened by its owner. It contains a public key encryption device, where the private key is unique to this printer and is only known to the printer itself. The printer's public key and the name of the owner are registered in a database provided by the trusted printer manufacturer. When the owner requests a document from an information vendor, the vendor first ensures the owner is authorized (e.g., has paid), then it fetches the public key of the printer from the registry, it encrypts the document using the public key and sends the result. When the owner receives the data, he can send it to the printer which can then decrypt and print the document. However, the electronic data cannot be used for anything else, as only this one printer can decrypt it. The data can be resent to the printer to create another paper copy, so the document can be reproduced in this way. However, illegally reproduced paper copies is a "previously unsolved problem" that this scheme does not address.

The main problem with this scheme is that it is too restrictive. It is more of an "electronic paper delivery system" than a Digital Library. Users cannot browse through documents before buying, and cannot use parts of the document in others, e.g., for quotes. Furthermore, it requires special purpose hardware. However, it may still be useful in conjunction with other schemes. For example, perhaps users can be allowed to browse through low-resolution copies of documents, or through documents that have key components missing. Once the user decides he wants to read the document, he can purchase the a high quality copy that can be delivered via the secure printer. The scheme can also be adapted for a "secure computer" instead of a printer.

The second technique we wish to illustrate is that of an *active document* (suggested in [20]). The idea is that an information vendor does not send out a documents; instead it sends out programs that can generate documents. When a user receives one of these programs, call it P, he can run it on his local machine. Embedded within P and its data structures is the encrypted document; as P runs, it displays the document. However, before or during display, P sends a message to the vendor, informing it that it is being run, and waits for a response. This way, the vendor can charge each time P runs, or can limit the number of times P runs.

This scheme also has its drawbacks. A user cannot read the document through his favorite viewer. The vendor must know the architecture of the user's machine in advance, to generate appropriate code. The user cannot see the document if the vendor's machine is unavailable on the network. Finally, the scheme is not bullet proof, since the user could run P in an software emulator of his machine that could record the characters of the document as they are displayed.

We are not arguing that the two illustrated schemes are the solution, or the only type of solution for protection, or that even protection is superior to detection of violations (next section). We are saying that for protection mechanisms, these two schemes provide good starting points, from which improvements and extensions may be developed. Our goal will be to study the spectrum of possibilities, and to carefully evaluate the strengths and weaknesses of each.

II.B.4.3 Detecting violations

Instead of focusing on protection of intellectual property, we may wish to assume most users are honest, allow them access to the documents, and focus on detecting those that violate the rules. Many software vendors have found this approach to be superior (protection mechanisms get in the way of honest users, and sales may actually decrease).

One possible direction is to incorporate a "watermark" into a document that identifies its origin [52]. For example, if we think of the documents as images, we may encode the watermark into a small number of random bits throughout the image. The users would be unaware of where the watermark bits were, but the information vendor that originally provided the document could extract them to determine who the document was sold to originally. If the document is possessed by a different person or organization, then a violation is detected. The main weakness of approaches such as these is that users may destroy the watermark by processing the document. For instance, passing the (image) document through a noise filter or lossy compression algorithm could easily change enough bits (without really altering the image) to destroy the watermark.

A second approach we plan to investigate is that of a *copy detection server* [1, 39]. The basic idea is as follows: When an author creates a new work, he or she registers it at the server. The server could also be the repository for a copyright recordation and registration system, as suggested in [25]. As documents are registered, they are broken into sentences. Each sentence is hashed and a pointer to it is stored in a large hash table.

Documents can be compared to existing documents in the repository, to check for plagiarism or other types of significant overlap. When a document is to be checked, it is also broken into sentences. For each sentence, we probe the hash table to see if that particular sentence has been seen before. If the document and a previously registered document share more than some threshold number of sentences, then a violation is flagged. The threshold can be set depending on the desired checks, smaller if we are looking for copied paragraphs, larger if we only want to check if documents share large portions. A human would then have to examine both documents to see if it was a legal violation.

The copy detection server can be used in a variety of ways. For example, a publisher is legally liable for publishing materials the author does not have copyright on; thus, it may wish to check if a soon-to-be-published document is actually an original document. Similarly, bulletin-board software may automatically check new postings in this fashion. An electronic mail gateway may also check the message that go through (checking for "transportation of stolen goods"). Program committee members may check if a submission overlaps too much with an author's previous paper. Lawyers may want to check subpoenaed documents to prove illegal behavior.

There are a large number of variations to the basic scheme that we plan to investigate. For instance, should the units be paragraphs instead of sentences? Or should we take into account order of the units (paragraphs or sentences), e.g., by hashing sequences of units? Is it feasible to only hash a fraction of the sentences of registered documents? This would make the hash table smaller, hopefully still making it very likely that we will catch major violations. If the hash table is relatively small, it can be cloned. Our mailer above could then perform its checks locally, instead of having to contact a remote copy detection server for each email. There are also other questions that need to be addressed. For example, how are sentences extracted from say Latex or Word documents? Can we extract them from Postscript documents, or from bit maps via OCR? Can these techniques be used for comparing programs? (We think so.) Can the copy detection server and the watermark idea be effectively combined?

II.B.4.4 Financial transactions

A Digital Library requires an electronic charging mechanism, so that information providers can be remunerated for their services. Although the notion of electronic cash has been suggested, e.g., [34], we believe that a simpler banking system would be more effective. Here customers and providers have accounts at *bank servers* on the network. Each account is backed by an actual cash deposit or by a credit line.

In current financial networks, typically the person whose account is to be debited (the purchaser) does not have access to the network. Hence, the purchaser must sign a check or credit card voucher to authorize the charge. (Sometimes the expiration date on a credit card is used as validation, but this is an extremely weak system!) In a full electronic system, however, the following scheme is more natural and secure: Purchaser P authorizes his bank PB to release up to X dollars to provider R (encryption is used for authentication). Then provider R asks its bank RB to collect Y dollars ($Y \leq X$) from P. Finally, PB and RB transfer the funds. (As in current financial systems, this last step is *not* a distributed atomic transaction. It is a pair of atomic steps at each institution, with possible compensation if things go wrong [5].)

Developing mechanisms for electronic financial transaction over an open network is one of the main foci of the CommerceNet Project (Section II.C), with which we are cooperating. (It is also the focus of the new Financial Services Consortium, with which we are establishing links.) Our main goal under this SIDLP Project will be to apply the technologies developed to the library application directly [28, 16, 56]. In particular, reading a document may not be as simple as paying a fee and getting the entire document. A user may first wish to browse a table of contents or sample sections, and then get the full document. Thus, we need to investigate "pay as you go" strategies. These need to be compared against "connect time" strategies where a customer pays an hourly fee to connect to a service, independent of what he reads.

Another important problem is that of reselling. Suppose I am the editor of a newsletter that provides excerpts and summaries of various articles. The classic solution would be for me to negotiate with my information providers a redistribution fee and then to charge my customers directly. However, in an electronic environment it may be possible for each copy of my newsletter to include direct payments to my providers. As a matter of fact, my newsletter could simply provide network pointers to the source documents, which my readers could then get and pay for directly [36].

We can also take this one step further and assume documents can have a positive or a negative value. If the value is positive, the reader pays; if it is negative, the reader gets paid to read. (Safeguards ensure a reader gets paid only once, even if he rereads the same document.) The negative value documents represent advertisements: if I want people to read about my new product, I should be willing to pay for this, and the readers of my advertisement should be encouraged to read it. If I produce my newsletter that has positive value, and add advertisement, I can reduce the positive value, making it less expensive for readers. The revenue I lose from my customers is made up by the advertisers. This is the way advertising works today; what we are proposing to do is to embed these mechanisms directly into a Digital Library.

Current news systems (like Unix NetNews [22]) do not have any cost mechanisms, and hence users are flooded with junk mail. It does not cost to advertise, so why no advertise all the time? There is no incentive to distribute high quality information, so why do it? We believe that a news system that incorporates the notion of positive or negative document value will cause many exciting services to blossom and will reduce the amount of junk messages (or will at least make it worthwhile reading the junk messages). Such a system can also charge for the basic delivery of messages, letting users pay for the fraction of the underlying network they actually use. Clearly, such a system should not replace existing free systems; we only suggest it as a complement.

The ideas we have presented in this subsection are clearly speculative. However, we believe they represent an exciting and critical aspect of Digital Libraries that has not been fully studied in the past, and needs to be carefully addressed. We intend to carry out this research, developing the right models and framework for discussing value and costs. In later years of the project, we plan to implement a simple news distribution system that incorporates the pricing policies we develop.

II.B.4.5 Customer impact

Beyond the technological problems we have discussed, there are clearly non-technical issues of equal importance that must be researched and resolved to make Digital Libraries a sustainable reality. Fortunately, one of our project partners, ACM, has a real-life environment with customers, materials, costs, and revenue at risk.

As part of our project, ACM will make available its materials in digital form, and will introduce new digital alternatives, such as a wholly electronic journal. At the same time, ACM and other participants of the SIDLP Project will develop and validate a business model for publishing in the Digital Library. This model will show the effects of decreasing print runs as electronic distribution increases, as well as the shifts between unit sales and access fees. ACM will also conduct evaluations of changing customer patterns and online information usage. These models and evaluations will also drive our study of property safeguarding and charging technologies, and will provide insights into the types of additional technologies and services that must be developed.

II.B.4.6 Summary

Digital Libraries require mechanisms for safeguarding the economic interests of the information providers, where a provider is not just a large publishing company, but any person who has valuable information. Our research will develop a variety of services (e.g., copy detection server, banking server) for the SIDLP Testbed, as well as models and protocols to be incorporated into the Information Bus. A second important goal will be to evaluate the changing business and social factors that impact the viability of the schemes we develop and of a Digital Library in general.

II.B.5 Agents

The term "agent" is encountered frequently today in connection with software automation (other terms used include "softbot" or "knobot," for software and knowledge robots). There is not yet general agreement on the role of software agents, or even on their very definition. Indeed, the work that takes place under this header is quite diverse, and we will not attempt to either justify or unify it. We will instead explain what we mean by the term and the role of "our" agents in the domain of Digital Libraries. We propose two activities: designing a general Information Bus agent engine, and implementing a specific agent (the personal information assistant). We discuss them separately in the two sections following the background review.

II.B.5.1 Background

In this section we discuss the general background to the agent work that we propose. Specific prior agent-related work in industry and academia will be discussed in the next two sections, when we discuss the particulars of our proposed research.

As was explained earlier, the role of the Information Bus protocol that we propose to design can be understood by analogy with the various efforts to facilitate inter-application communication (IAC) at the operating system level. To summarize the analogy, just as inter-application standards help glue together disparate applications by providing standard interfaces between them, our Information Bus will help glue together the elements of information space by standardizing the protocols essential to interactions with and among digital knowledge sources. Within industry, it has been a short step from the introduction of IAC standards to the creation of general scripting languages that support these standards; similarly, we can extend the above analogy to reach the notion of an "agent engine."

To briefly illuminate the idea of a scripting language, consider a facility such as an inter-application standard or a Graphical User Interface (GUI). There are two common modes of using such a facility. The less interesting one is for the provider to supply "hooks" which application writers may then use from within their own code to interface with the facility (this can be done either through so-called Application Program Interfaces (APIs) or through an event mechanism; the details are not important here). The more interesting approach is to create a general authoring tool, in which "meta-applications" can be written; these meta-applications simply call other applications as if they were subroutines. This is precisely the idea behind generic scripting languages, such as Summit's "Embedded Basic" for the PC which supports OLE-2, and Userland's "Frontier" and

Apple's "Applescript" for the Mac, which both support Apple Events. Of particular relevance are CNRI's "knobots" and the recently announced "TeleScript" by General Magic, which we discuss below.

II.B.5.2 The Information Bus agent engine

One function of the agent engine is to supply creators of information services with the analog of a scripting facility. The basic operations of the Information Bus, whatever they turn out to be, will all be included in the agent engine. A designer may specify a particular service in the language of the agent engine, using the primitives provided to execute commands specific to the Information Bus. Note that the Information Bus will have built-in primitives for communication among agents, effectively bringing a measure of homogeneity into the otherwise heterogeneous environment. As part of its role in scripting agents, the agent engine will capitalize on this uniformity.

The agent engine transcends what is usually meant by a scripting facility. An agent created by the agent engine will not be merely a script that runs once and quits. Rather, agents are persistent objects that exist for extended periods of time. For example, agents may be defined to function as (scripted) daemons, performing repeated actions, watching for special events, executing at certain times, and so on.

Although our agents are "knowledge based"—they embody knowledge about the Information Bus—no "intelligence" is built into the agent *engine*. In particular, our agent engine does not automatically build learning components into the agents it generates. Our position is that the utility of the various learning algorithms is yet to be proved, so we factor them out of the engine itself. However, our agents possess all the structure needed to implement any number of learning algorithms, and, as discussed below, the particular agent we intend to implement – the personal information assistant – does have a substantial learning component.

We should mention that we are well positioned to design the Information Bus agent engine, having introduced the notion of Agent Oriented Programming (AOP) in 1989 and having conducted research on the topic since then (see [47] for an initial description). Our work to date in this area consists of designing and implementing three agent languages of increasing complexity, testing these languages in limited domains (the most ambitious ones have involved network management and calendar coordination), and investigating general coordination mechanisms among agents (in a rather theoretical fashion). This new, highly complex application domain will require substantial modification and customization of the engine design; it will be our most ambitious application to date.

II.B.5.3 The Personal Information Assistant

The research directions discussed thus far are application-independent in nature: they address specific functionalities to be designed and implemented on top of the Information Bus. The research described in this section is complementary; we propose to design and prototype a specific application, a personal information assistant (PIA) to aid in the routine daily activities of an information space user. Because of our familiarity with the domain, we have decided to model the information needs of a typical computing professional; however, most of these needs will be seen to be common across professions. In particular, we have decided to concentrate on the following services:

- Information search and sharing (targeted, untargeted, and relevance-based).
- E-mail assistance (scripting and filtering).

- Mobile access (mobile terminals, cellular communication).

As we describe below, our emphasis is on tying together personal and impersonal information and making them accessible through ubiquitous "information terminals," some of which will be implemented on hand-held, radio-linked devices. Our plan is to implement the PIA as an Information Bus agent, using the agent engine discussed in section II.B.5.2. However, the implementation details are unimportant for understanding PIAs and their function. Note that since some of the functionalities of the PIA rely on results from other thrusts of this research proposal, the PIA will be implemented in an incremental and modular fashion.

Search

One job of the PIA is to support intelligent search of the information space, from major information sources such as newsgroups and commercial data sources to personal E-mail and other files. We divide search into targeted search, untargeted search, and an intermediate "relevance-based" search.

A targeted search is conducted when the user requests a specific piece of information, e.g., "references to reinforcement learning in AI." Because the body of information with which we are dealing is enormous, diverse, and dispersed, it is not straightforward to standardize this type of search. We will approach this problem in two ways. The first is to incorporate results of the other research thrusts of this proposal; in particular, we will give access, through the PIA, to the services described in Section II.B.3. The second way of coping with the difficulty of targeted search is to build an information-sharing component into the PIAs. Our thesis is twofold: information retrieval will depend in a large measure on the cooperative nature of the inhabitants of information space, and the most important databases will be those about other databases. A PIA will support cooperation by acting as a gateway for other PIAs into the part of its local knowledge that it is authorized to share. PIAs will also have explicit knowledge of which PIAs possess which type of information. For example, John's PIA might contain the information that Mary's PIA has access to an extensive bibliography on Machine Learning. Our PIAs will thus play a role in attaining the value-added information structure discussed in Section II.B.2.3.

Untargeted search is also known as "data mining." We envision the PIA as scanning a broad spectrum of articles (for example, the Dialog newspaper articles or 20 specified Internet newsgroups) on a daily basis and doing two things: collecting important statistics (such as the number of items and the most common topics), and presenting a specified number of articles deemed most interesting to the user. This is expected to be an adaptive process, in which the PIA continually improves its success rate. Current memory-based techniques (e.g., Maes' [32]) will be augmented by adding inter-user reinforcement: the PIAs of different users within a group will communicate, and if one PIA retrieves an article particularly pertinent to its user's peers, that information will be shared with the peers' PIAs.

Relevance-based search starts out targeted, but continues to follow relevant threads to uncover potentially useful information not anticipated by the user. The "tolerance" of the search determines how far the search may stray from the specified information. The primary problems to be addressed in relevance-based search are identifying an appropriate measure of relevance and determining a useful tolerance level.

E-mail assistance

As discussed in Section II.B.3, E-mail is an important mode of disseminating archival material (such as an SDI service or newsfeeds), but here we also include more personal material. Any user of E-mail is acutely aware of the overwhelming volume of E-mail traffic one is subjected to daily; it is not uncommon to return from a brief trip and find several hundreds of messages waiting. This traffic will clearly grow with the expansion of the information infrastructure. We envision two forms of assistance in dealing with e-mail. One is to have PIAs support mobile terminals accessible through a radio modem (see below), so that the user need not let too much mail accumulate. The other, and more important one, is to have the PIA automatically process as much E-mail as possible. This processing might include automatic replies, forwarding to a secretary, sorting by priority, and re-routing into different folders. The user should have an easy language in which to program the mail handling behavior of the PIA. Although we do not envision this as a main emphasis in this case, there is another opportunity here for adaptive behavior by the agent as it learns which of its choices more closely match its user's preferences.

Mobile access

To caricature things, the picture so far is of "everyone" having a PIA and, through it, being able to access "anything"; this component of PIAs aims to add the "everywhere" and "anytime" dimensions. Specifically, we aim to implement an interface to the PIA on a portable device, such as the HP100LX or the HP Omnibook, connected to the PIA through a radio modem. We do not consider this element of the proposal a significant *research* component, but we do believe radio access to online information is destined to play an important role, and we would be remiss not to include it in our project in order to gain realistic experience.

Prior PIA-related work

Previous related work can be divided into that related to specific components of the PIA, such as E-mail automation, and that related to its integrated nature. The first exists in both academia and industry; the latter mostly in industry.

Most modern commercial software is written in a way that encourages scripting. In particular, modern E-mail programs are usually enhanced with limited user-customization features. This is true, for example, of "CC:mail" in Lotus Corporation's Notes on the PC, and in QualComm's "Eudora" on the Mac. No commercial E-mail system to date aids the user by tracing "relevant" information (as we suggest above), and none attempts to automatically adapt to users' patterns. In academia, there have recently been a number of attempts to introduce such adaptation. Maes's work addresses adaptation in the contexts of E-mail handling and newsgroup scanning [32]. Both Maes [32] and Mitchell [9] attempt to automate aspects of meeting scheduling. Etzioni [12] uses so-called "softbots" to track user activities within the Unix OS and initiate help when appropriate; a first corresponding commercial product is Charles River Analytics' "Open Sesami!" (TM), which attempts to automate the user's routine Finder-level operations on the Mac.

The integrated nature of the PIA is similar to the functionality of Personal Digital Assistants (PDAs) such as the Newton, Zoomer, EO, and Simon. However, "integration" in current PDAs means the ability to switch easily between a number of different applications (such as memo, phone book, E-mail and calendar) and sometimes access one from another (for example the phone book from the calendar); this is not the sort of uniform window into the information space which we wish to pursue. Our PIAs come much closer to realizing Apple's own "Knowledge Navigator" vision and Microsoft's "Information at your Fingertips" slogan.

We conclude this section with two particularly relevant software agent notions. One is the "knobots" of the Corporation for National Research Initiatives (CNRI): knobots are pieces of code dispatched to a remote site to execute and search for information [26]. In a typical scenario, a knobot searching for information sends a clone of itself to a remote Digital Library; the clone executes the query remotely and returns the results to the originating knobot. At this stage, knobots are a concept with a prototype implementation, but they have no accompanying theory nor a general programming language. The second notion, General Magic's TeleScript language, comes close to our idea of an agent engine in that agents are written in a uniform scripting language. Although their functionality—locating the relevant databases, extracting the information and shipping it back to the querying site—is close to our agents', both knobots and TeleScript agents are designed to navigate the network; this contrasts with our framework in which stationary PIAs cooperate by accepting each other's requests for information. It remains to be seen whether the communication overhead of the PIA-to-PIA interaction in our approach outweighs the attendant security perils of the knobot/TeleScript approach.

Ultimately, the most important difference between our concept and these other ideas is perhaps that our agent engine will be written specifically for the Information Bus, supporting its basic primitives.

II.C. Testbed Facility

Section II.B described the research problems that will be addressed by this project. In order to validate and evaluate the mechanisms and ideas developed, we will implement a SIDLP Testbed facility that gives access to a variety of information sources. We will provide prototype software to users of the testbed and evaluate their use of it, in a cycle of testing and redesign. This section will describe the testbed in terms of its architecture, sources, and users, and give a schedule for its development.

II.C.1 Testbed Architecture

The architecture of the SIDLP Testbed will follow the logical architecture presented in Section II.A, where a collection of Interface Clients (IC), Services (S) and Information Sources (IS) are logically connected via an evolving information bus. As described there, the Information Bus is not a single locus of computing or communication, but a distributed architecture that enables point-to-point communication among the variety of clients, sources, and services, using common protocols.

The testbed will consist of:

- a collection of machines including our dedicated experimental machines, along with the hosts and servers for the information providers and users on the network.
- a collection of software including existing network and interface applications, plus software that we will produce.
- a collection of information sources, including existing sources provided by our participant companies, local sources, sources available on the Internet, and experimental sources we will produce (see Section II.C.2.)

II.C.1.1. Hardware

We are not basing our Digital Library on a special class of machines to be used by those who will access it. We want to provide the glue that allows standardly available personal computers and workstations to serve as library access points for both retrieving and publishing. We will do this in a way that works for existing machines and scales up to make use of higher capacity and resolution as it becomes available. Therefore, our architecture does not include specialized hardware. We will need two kinds of machines for the project:

- Workstations for experimenting with new interfaces (including capacities for real-time video and 3-D rendering) and for the general work of programming and development.
- Servers for providing experimental services, such as agents, search services, and translators that can mediate communications of a variety of kinds, as described in Section II.B.

The equipment requests are detailed in the budget (Section V), and are for UNIX-based workstations and servers. In addition to both servers and interfaces on UNIX systems we will support Macintosh-based interfaces. Some Macintosh machines are requested for placement at several user sites to facilitate access to the SIDLP Testbed (see Section V).

II.C.1.2. Software

The main product of our research will be the suite of software that implements the Information Bus protocols, services, and interfaces. In addition to running this software on our dedicated machines, we will provide it to users both for access to the Information Bus, and for integrating information onto the bus. We have not prepared a detailed list of software, and the choice of components to be developed will be a defining goal of the project, to be done during the first Project Phase (see Section II.C.4).

The different software modules will include:

- Interface clients
- Agent servers
- Search servers
- Other services (e.g., copy detection, commercial transactions)
- Translators (providing access to and from the Information Bus)
- Toolkits (which provide easy use of the protocols in building applications)

An initial suite of software will be released after the first year, with a first level of coordination and an emphasis on experimenting with a wide variety of functionality. A full suite will be released for testing after the third year, with an integrated structure and a focus on providing a coherent information environment.

II.C.2 Sources in the Collection

Since our research goal is to access to large and heterogeneous collections of documents, we will not focus on the capture of documents, and our resources will not go into document scanning or input. Instead, the testbed will provide access to existing large collections of documents, as well to emerging collections that are being generated as part of other research projects, including those of some of the participating industrial partners.

Specifically, the theme of our collections will be *computing and related literature*. It covers not just what may be of interest to a computer scientist, but also documents of interest to a computer professional, such as legal and business information about the computing field. The collections that will be included are:

The CNRI CS Technical Reports Library

A consortium of five U.S. universities, including Stanford, is currently developing an electronic library of Computer Science Technical Reports. The reports will be stored as images, and full text or Postscript versions will be available in some cases. Also, as part of the current NSF/ARPA/NASA Initiative in Digital Libraries, it is clear that other sources of documents will become available in the near future.

Dialog

Dialog is the one of the world's largest and most comprehensive data banks, with more than 400 databases (over 3 terabytes), including the complete text of stories from more than 2,500 newspapers. Dialog's databases cover an unequaled variety of topics, with particular emphasis on business, science and technology.

O'Reilly Associates book and journals

O'Reilly is one of the major publishers of computing literature, and has strongly advocated electronic publishing. They currently publish an on-line magazine, the *Global Network Navigator*, and will soon be initiating a series of on-line books about computing, which will become available as part of our collection for the testbed.

Public domain sources available on the Internet

These sources include the World-Wide Web (WWW), Archie, and the Wide Area Information System (WAIS). For a large but still partial listing of Internet sources, see [38], which is published by O'Reilly Associates. One of our partners is WAIS Inc., which is developing commercial WAIS software and databases on the Internet as well.

The Stanford University Folio System

FOLIO provides over 60 document bases, including INSPEC (abstracts of Computer Science, Electrical Engineering, and Physics papers), ABI (abstracts for business periodicals), LEXIS-NEXIS (full text news, legal and business transcriptions), EI/Compendex (Engineering and Technology Abstracts), the Oxford English Dictionary. Folio also provides access to the electronic catalogues of the Stanford Libraries, the Stanford Bookstore, as well as the catalogues of other universities worldwide. Folio includes a full-text computer science source on a networked CD-ROM called Computer Select, which will enable us to include the medium of on-line CD-ROM in our design.

CommerceNet

Recently, ARPA has funded a Technology Reconversion Program project to develop electronic commerce on the Internet. One of our partners, Enterprise Integration Technology is the prime contractor, while Stanford University is a subcontractor. As part of this project, catalogues of various Bay Area computer-related businesses will be put on line, and facilities will be built for ordering products electronically. We will have access to the catalogues and libraries implemented under CommerceNet.

ACM Publications

The Association for Computing Machinery (ACM) publishes 15 journals, about 50 conference proceedings and 34 technical newsletters each year, and a variety of books, for a total of over 40,000 annual printed pages. By the end of the first project year, ACM plans to make available electronically new issues for all 15 journals. By the second year, new conference proceedings will be available, and newsletter for the third. ACM is also involved in digitizing old issues of its journals.

NASA Collections

Our partner, Interconnect Technologies Corporation is targeting a variety of NASA information sources. Most of these sources are public, the software provided by Interconnect will facilitate identification and access to these collections. Candidates include the COSMIC catalogue, NASA Office of Standards and Technology database STIS, NASA Global Change Master Directory and satellite images, Lunar and Planetary Institute Bulletin Board, NASA's MasPar repository, NASA Extragalactic Database, Space Physics Analysis Network, NASA's SBIR database. Interconnect will make available to us catalogues and indexes for collections such as these, as they develop them.

Local document and image sources

This will include all of the working materials for our own project, including the specifications, programs and documentation, and email archives of communications. Similar collections can be included for any of the users of the testbed. That is, in order to apply Information Bus services to their own collections, they do not need to store them in some central server, but can run the software we provide in their own local environment.

In summary, we are well supplied with many sources of useful digital information through our participating groups and the Internet. As we have stated, the challenge will *not* be in obtaining access to the significant collections; rather, the challenge faced will be to use the wealth of information in a coherent internetworked Digital Library environment.

II.C.3. The User Community

We plan to test and evaluate our system with three progressively larger user communities.

Project users Project users are the people associated with the project itself, at all of the different sites. We will make every effort to integrate the interfaces and servers that we are building into our own work, as we will be each others' most exigent clients in getting things up and running. This will begin immediately as we start to put together pieces of the overall software suite that will make up the Information Bus, and will continue throughout the project and beyond.

Local users Local users are in the same institutions as project members, so that support can be easily provided, but they will not have a specific interest in the project, so they will be evaluating the systems on the basis of actual usefulness, not future promise.

External users External users will be provided by project participants such as the ACM, NASA Libraries, and Dialog. They will provide a broader audience and we will do more systematic testing on their use of the facilities of the Information Bus. A major part of the planning during Year 3 (see Section II.C.4) will be in selecting an appropriate set of users from the computing community and preparing the materials (both the software and supportive materials such as documentation) that will make it possible for them to use our systems effectively.

To summarize, the users from the last two categories will be drawn from the following groups:

- Stanford University faculty, staff, and students. This represents a large community of over 15,000 individuals. Many of these have interests in computing, including individuals in the Engineering and Business Schools, scientists in the School of Humanities and Sciences, and so on.
- Scientists at the NASA Ames Research Center. The NASA Ames Library Branch will tie us to a sophisticated user community of over 2,000 government and contract scientists and engineers. Their research staff, especially in the Computational Systems Research branch both produce and use the computing literature that will be targeted by this project.
- Customers at our partners. Several of our partners, such as ACM, Dialog, O'Reilly and WAIS Inc. are interested in eventually providing testbed access to their customers. (Just Dialog itself has over 150,000 customers; ACM has 75,000 members.) This would be on an experimental basis, until we develop the technology to charge for use of the private data sources.
- Students and researchers at other universities, who will be given access to our testbed on an experimental basis, via the Internet. (See Section II.C.5.)

It is important to note that having computer professionals as our target users has advantages as well as disadvantages. Our expected users will be computer literate and relatively sophisticated. This means it will be easier to get them to use advanced, experimental library facilities. It also means that the developers of our testbed can directly use the testbed for their own research, obtaining feedback much faster than if services were provided to a widely dispersed community. Finally, in the short term we expect there to be more computing literature available electronically than for other fields, making it easier for us to perform significant experiments.

There is, however, a danger that by targeting a computer literate community, we will develop tools and interfaces that are hard for the general public to use and understand. We are making a conscious tradeoff in which the ability to accelerate early research is being traded off against the full generality that will come when more diverse audiences are addressed. We would expect other projects funded by the Digital Libraries Initiatives to choose complementary points in this tradeoff space, doing less fundamental research on the mechanisms and concentrating on working with other user communities.

We believe that for the kind of research we are proposing, the benefits of using a computer literate community far outweigh the dangers. We can draw an analogy to developments at Xerox PARC in the 70's, where many of today's key computer technologies were developed, such as the personal computer, windows, WYSIWYG editors, the Ethernet, and Smalltalk. One of the main reasons why this research was so successful in such a short period of time, is that the technologies were in constant use by their developers. Similarly, we argue that we are developing the basic Digital Library technologies that will be in common use in the year 2010 and beyond. By targeting today's computer literate community, we are building the base that will serve the library patron of the next century.

II.C.4. Testbed Development Schedule

The testbed will be developed in six phases, with a number of activities spanning across them, as depicted in the time line of Figure II-4.

Phase I: First six months: Getting started During the initial months we will be creating a shared background of understanding and expectations among the participants from the different institutions, setting up initial working groups and setting goals for the first round of implementation. This will be initiated by a two day group offsite meeting at the beginning. A second offsite meeting at the end of the period will be used for reaching agreement on the next phase activities.

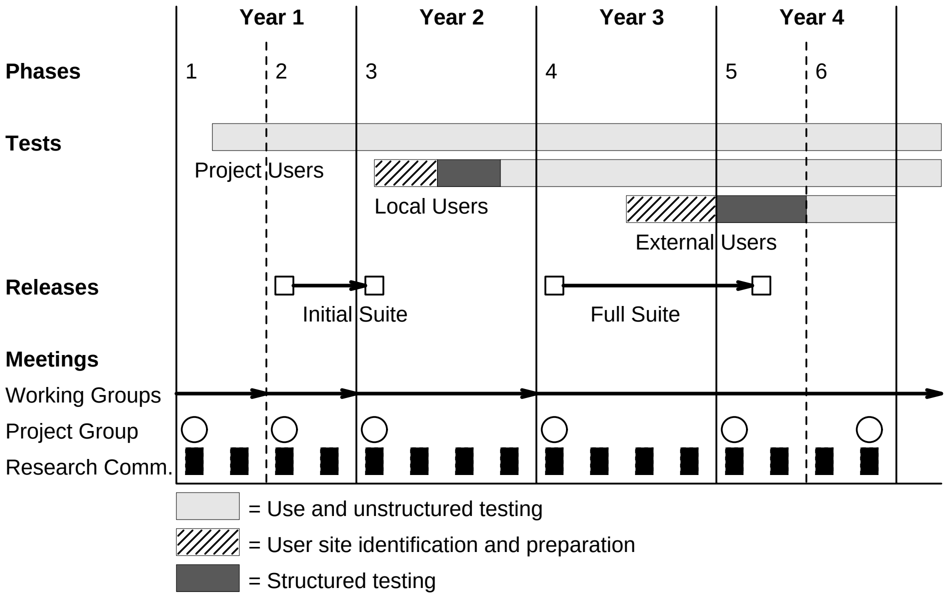

Figure II-4: Timeline for the Stanford Integrated Digital Library Project

Phase 2: Second six months: First implementations During the second six months we will implement an initial suite of information sources and software within the framework of the Information Bus. These will be coordinated, but not fully integrated. Some of them will be foundations for others to come later (as with the basic protocols) and others will be experiments with specific interfaces and services. They will include things such as some regularized forms of access to Folio, the CNRI TR Library, AOL, WWW, WAIS, etc. At the end of phase 2 we will have a coordinated release that we will make available to the project participants and that we will be preparing for testing by local users.

Phase 3: Second year: Testing round 1 and further development During the second year we will do our first round of systematic testing. During the first few months (marked by the diagonal stripes in the timetable) we will identify a suitable set of test users and provide the materials and training needed to let them test those aspects of the system that are ready. During the remaining months we will respond to their experience in an iterative design cycle, in which updates and new releases will be produced as necessary. In conjunction with this, we will be laying the foundations and identifying the needed pieces for the full suite of Information Bus software needed for the second cycle of testing.

Phase 4: Third year: Full scale development and test preparation During the third year we will take the results of testing from Phase 3 (and ongoing use by the same users) and develop the full suite of software for the Information Bus. This will include services such as an information finding server with searching for candidate sources, a copy detection service, an initial set of financial transactions, a personal information assistant, etc. We will also add information sources that become available by that time frame, for instance, the CommerceNet catalogues and on-line books from O'Reilly. We will begin with a group offsite meeting at which we will determine the basic directions and design decisions, and then working groups will do the development throughout the year. During the second half of the year, we will be preparing for the broader testing, identifying user communities and providing materials and training as needed.

Phase 5: Fourth year, first half: Testing round 2 During the first half of the fourth year we will do systematic testing with the external user community, and continue development in response to their experience.

Phase 6: Last six months: Evaluation and planning During this last period we will evaluate the results of the tests, and incorporate them into plans for the continuation of the project. This will include identifying future research areas and looking into possibilities for commercialization or other dissemination of what we have produced. We are not guaranteeing external sites that the software will be supported after the end of the fourth year, but are open to that possibility if appropriate funding and mechanisms are available. In any case, we expect to continue using it in our own work and for those at our own local sites who want to continue with it. We have scheduled a final group offsite meeting in the middle of this last phase, so that we can review what has happened and can initiate plans that working groups can carry out in the last few months.

II.C.4.1 Development, testing and evaluation

Our testing will be of two different styles, which we have called "structured" and "unstructured."

Unstructured testing will go on throughout the duration of the project, as we develop software, use it ourselves, and test it with others. We will develop a set of practices and tools to use this testing to the greatest benefit, including:

- *Version, documentation and release management* tools, of the kind developed in work on software engineering, to coordinate the development and distribution of a large body of interlinked software.
- *Use measurement* tools. These include:
  - *Logging modules* that are built into the application programs for generating logs of activity
  - *Analysis tools* that can be used to process the logs to produce quantitative and statistical analyses of their uses.

We expect to thoroughly "instrument" our software for interfaces and services, to provide usage information that can help identify and analyze problems and areas for development.

- *Developer feedback* tools, which make it easy for users to report problems and enter into a dialog with developers about what is happening and what can be done. We can make use of the network information interfaces we are developing (e.g., specialized forms in Mosaic) to facilitate this interaction.

Structured testing will include several aspects:

- *Careful choice of test users*, to identify potential interactions with currently used tools, past experience and computer knowledge. We will attempt to avoid situations where special circumstances of these kinds would bias the results, and to aim for breadth and variety in our test population.

- *Planned training and debriefing activities* so that we can better isolate the issues of the software itself from the situational problems of dissemination and support.

- *Planned measurements*, which build on the instrumentation mentioned in the previous section. While unstructured logging will provide a large body of data that can be explored, the planned logging and measuring will be targeted at specific questions that we formulate as part of the development of the Information Bus software suite.

- *Structured user observations, interviews and surveys*, using instruments such as the Questionnaire for User Interface Satisfaction [7]. Some of these will be based on ongoing use, and others on yearly demonstrations/evaluation sessions. For each of these we will assemble a panel of users (computer scientists and professionals) for a one-day session, in which they use services from the testbed. In these sessions we will use a variety of HCI observation techniques, such as paired thinking-aloud protocols and video recording for later analysis.

II.C.5 Testbed Management

The overall management of the testbed will be the responsibility of the Executive Committee of the project, with the advice of the Advisory Board (see Section III.C for a description of the levels and membership of management groups). Operational management will be carried out by a full-time Testbed Manager, to be hired as part of the Stanford core.

The Manager will negotiate with the bodies of potential users to set up the tests, provide training and documentation, and carry out the evaluation procedures, such as surveys and interviews. Decisions about the user population and specific test configurations will be made by the manager with the consent of the Executive Committee, within a general set of guidelines to be formulated by the Advisory Board.

Since our testbed will provide access to some commercial sources, external users may have to be charged for access to those sources, and negotiations on this will be approved by the Executive Committee.

II.D. Expected Accomplishments

In general terms, our research will develop concepts and technology for an integrated, economically feasible, scalable, and usable digital library, built out of a collection of heterogeneous repositories of digital information. In particular, we expect to accomplish the following:

- Develop models for library documents and services that can facilitate interoperation and sharing across heterogeneous repositories.

- Develop scalable Information Bus protocols for accessing and manipulating documents, and for calling on a variety of library services.
- Implement a Human Computer interface that can clearly present to users the wide variety of information resources available over the network, and can let users easily specify their information needs and interests.
- Design and implement scalable information finding services appropriate for a very large, distributed, heterogeneous, and sometimes disconnected digital library.
- Develop and test mechanisms for safeguarding intellectual property rights, and for charging for services and information access.
- Design and implement an agent engine that can assist in the rapid prototyping of library service agents.
- Develop and demonstrate a personal-information-assistant agent that can seamlessly provide access to personal as well as shared information of all types, and can capture and manage common day to day information manipulation tasks.

The schedule for these expected accomplishments is closely correlated to the schedule for our Testbed Facility (Section II.C.4), since it is our clear goal to implement and demonstrate each of the mechanisms and concepts as we go along. Here we summarize where we expect to be at the end of each year:

1. **Year 1:** At the end of this year, we expect to have developed a simple Information Bus model and its associated translators. Our testbed will provide access to an initial suite of software and information sources, mainly ones that are today easily available over the Internet, or those that our partner companies already have. We also expect to have completed fundamental research in each of our thrusts, for instance, developing the architecture for the agent engine, key algorithms for copy detection, and strategies for information finding.
2. **Year 2:** We will have completed an evaluation of the initial release of software and sources by the end of this year. A more complete architecture for the Information Bus will be developed at that time, allowing a richer set of operations and services in our system. The testbed will also provide access to new collections, as they become available, e.g., ACM publications and CommerceNet catalogues. We also expect to have some simple prototypes of the new ideas under research, for example, a human computer interface that can allow non-expert users to perform powerful searches over multiple document repositories, or a personal information assistant.
3. **Year 3:** We will continue to fill the "pipeline," performing more basic research, generating new concepts and mechanisms, implementing and demonstrating them, and evaluating them. By the end of this year we will have developed the full suite of software for the Information Bus. More complete and powerful services will be included, such as an agent engine, an eventual finding service, a financial service, and so on. In addition, results of customer evaluations to be performed by ACM will be available and will be used to refine our financial and access models.
4. **Year 4:** By the end of the final year we expect to provide access, via our testbed, to the most diverse and comprehensive collection of computing literature. We will have completed

a six month course of systematic testing of all aspects of the Information Bus, and will have analyzed and evaluated the results. In addition to these experimental results, the testbed will provide a real environment for testing and evaluating a wide range of library services, including those developed in earlier years of our project, as well as those of other researchers who will have access to the testbed.

With the type of project we are proposing, it is not easy to have precise metrics to quantify progress and success. For example, if our goal were to digitize a body of documents, our metric could be the number of documents scanned, and we could set specific targets by year. However, since our "deliverables" are concepts, services and algorithms, our measures of success cannot be as precise.

In spite of these difficulties, in Section II.C.4 we described some candidate metrics and procedures for evaluating our testbed. Beyond this, success of our project will be evaluated by answering the following questions:

- Can our testbed system provide access to a significant number of heterogeneous information sources?
- Do the services we provide (e.g., information finding) meet their stated goals?
- Are the end-user interfaces we provide easy to use and give users the information they want?
- Can we demonstrate via experiments or analysis that our services and techniques scale to larger systems?
- Can we show that the algorithms and mechanisms we develop are superior, in terms of performance, reliability, or some other standard metric, to ones in existence at the beginning of the project?
- Do we have a significant number of people using our testbed system in a realistic way?

At the end of each project year, we will prepare a report that answers the above questions, providing documentation to support our answers. We believe that, given the nature of our project, this is the most realistic and effective way to see if our goals have been met.

II.E. Summary

To summarize, we propose to design the Information Bus; this involves investigation of conceptual issues as well as definition of data structures, protocols, and services. We will pursue a bottom-up strategy, designing, implementing, and experimenting with a number of library tools and services on the SIDLP Testbed. We believe that this research effort will contribute fundamental concepts and mechanisms that will make the *Integrated* Digital Library a reality.

In Section III we list the major tasks to be undertaken as part of this project, and the participants that will carry out the work.

References

[1] C. Anderson. Robocops: Stewart and Feder's mechanized misconduct search. *Nature*, 350(6318):454–455, April 1991.

[2] T. Berners-Lee, Robert Cailliau, J.F. Groff, and B. Pollermann. World-wide web: The information universe. *Electronic Networking: Research, Applications and Policy*, 1(2), 1992.

[3] D. G. Bobrow and T. Winograd. An overview of KRL, a knowledge representation language. *Cognitive Science*, 1(1):3–46, 1977.

[4] U. Bormann and C. Bormann. Standards for open document processing: current state and future developments. *Computer Networks and ISDN Systems*, 21(3), May 1991.

[5] Y. Breitbart, H. Garcia-Molina, and A. Silberschatz. Overview of multidatabase transaction management. *VLDB Journal*, 1(2):181–239, October 1992.

[6] S. K. Card, G. G. Robertson, and J. D. Mackinlay. The information visualizer, an information workspace. In *Proceedings of CHI'91*. ACM Press, April 1991.

[7] J. Chin, V. Diehl, and K. Norman. Development of an instrument measuring user satisfaction of the human-computer interface. In *Proceedings of CHI'88*, pages 213–218. ACM Press, 1988.

[8] A. Demers et al. Epidemic algorithms for replicated database maintenance. *Operating Systems Review*, 22(1):8–32, January 1988.

[9] L. Dent, J. Boticario, J. McDermott, T. Mitchell, and D. Zabowski. A personal learning apprentice. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, Cambridge, MA, 1992. MIT Press.

[10] R. S. Dick and E. B. Steen, editors. *The Computer-based patient record.* National Academy Press, Washington, DC, 1991.

[11] D. Engelbart. The augmented knowledge workshop. In A. Goldberg, editor, *A History of Personal Workstations.* Addison-Wesley, 1988.

[12] O. Etzioni, N. Lesh, and R. Segal. Building softbots for UNIX (preliminary report). Technical Report 93-09-01, University of Washington, 1993.

[13] C. Faloutsos. Access methods for text. *Computing Surveys*, 17(1), 1985.

[14] G. W. Furnas. Generalized fisheye views. In *Proceedings of CHI'86.* ACM Press, April 1986.

[15] J. R. Garrett and J. S. Alen. Toward a copyright management system for digital libraries. Technical report, Copyright Clearance Center, 1991.

[16] M. Getz. Electronic publishing: An economic view. *Serials Review*, 18(1–2):25–31, 1992.

[17] D. K. Gifford, P. Jouvelot, M. A. Sheldon, and J. W. O'Toole, Jr. Semantic file systems. *Operating Systems Review*, 25(5), 1991.

[18] D. Goldberg, B. M. Nichols, D. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12), December 1992.

[19] L. Gravano, H. Garcia-Molina, and A. Tomasic. The efficacy of gloss for the text database discovery problem. Technical Report STAN-CS-TN-93-002, Stanford University, December 1993. May be obtained via Anonymous FTP from db.stanford.edu in directory /pub/gravano/1993/stan.cs.tn.93.002.ps.

[20] G. N. Griswold. A method for protecting copyright on networks. In *Joint Harvard MIT Workshop on Technology Strategies for Protecting Intellectual Property in the Networked Multimedia Environment*, April 1993.

[21] R. E. Horn. *Mapping Hypertext: Analysis, Linkage, and Display of Knowledge for the Next Generation of On-Line Text and Graphics.* The Lexington Institute, 1989.

[22] M. Horton. *Standard for Interchange of USENET Messages.* Unix documentation.

[23] M. B. Jensen. Making copyright work in electronic publishing models. *Serials Review*, 18(1–2):62–66, 1992.

[24] B. Kahle. Wide area information servers: an executive information system for unstructured files. *Electronic Networking: Research, Applications and Policy*, 2(1):59–68, Spring 1992.

[25] R. E. Kahn. Deposit, registration and recordation in an electronic copyright management system. Technical report, Corporation for National Research Initiatives, Reston, Virginia, August 1992.

[26] R. E. Kahn and V. G. Cerf. The digital library project volume 1: The world of knowbots. Technical report, Corporation for National Research Initiatives, March 1988.

[27] E. Krol. *The Whole Internet: User's guide and catalog.* O'Reilly and Associates, Inc., Sebastopol, CA, 1992.

[28] M. Lesk. Pricing electronic information. *Serials Review*, 18(1–2):32–37, 1992.

[29] D. E. Liddle. What makes a desktop different (user interfaces). *Multimedia Review*, 1(4), Winter 1990.

[30] A. Luniewski, P. Schwarz, K. Shoens, J. Stamos, and J. Thomas. Information organization using Rufus. *SIGMOD Record*, 22(2), June 1993.

[31] P. A. Lyons. Knowledge-based systems and copyright. *Serials Review*, 18(1–2):88–91, 1992.

[32] P. Maes and R. Kozierok. Learning interface agents. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, Cambridge, MA, 1993. MIT Press.

[33] L. Masinter and R. Rao. Generic archival information access. Available through the WorldWideWeb via <http://pubweb.parc.xerox.com/hypertext/dlbx/gaia.html>, December 1993.

[34] G. Medvinsky and B. C. Neuman. Netcash: A design for practical electronic currency on the Internet. In *First ACM Conference on Computer and Communications Security*, November 1993.

[35] L.G. Mondschein. SDI use and productivity in the corporate research environment. *Special Libraries*, 81(4):265–279, Fall 1990.

[36] T. H. Nelson. *Literary Machines 90.1.* Mindful Press, Sausalito, CA, 1980.

[37] J. Nielsen. *HyperText and HyperMedia.* Academic Press, Inc., Boston, 1990.

[38] K. Obraczka, P. B. Danzig, and S.-H. Li. Internet resource discovery services. *IEEE Computer*, 26(9), September 1993.

[39] A. Parker and J. O. Hamblen. Computer algorithms for plagiarism detection. *IEEE Trasnactions on Education*, 32(2):94–99, May 1989.

[40] G.J. Popek and C.S. Kline. Encryption and secure computer networks. *ACM Computing Surveys*, 11(4):331–356, December 1979.

[41] S. Putz. Design and implementation of the system 33 document service. Technical Report P93-00112, Xerox PARC, 1993.

[42] T.B. Rajashekar. Improving SDI systems: Implication of new retrieval models. *Library Science with a Slant to Documentation*, 25(1):44–62, March 1988.

[43] R. Rao, D. M. Russell, and J. D. Mackinlay. System components for embedded information retrieval from multiple disparate information sources. In *Proceedings of the ACM Symposium on User Interface Software and Technology.* ACM Press, November 1993.

[44] G. Salton. *Introduction to Modern Information Retrieval.* McGraw-Hill, New York, 1983.

[45] G. Salton. *Automatic Text Processing.* Addison-Wesley, Reading, MA, 1989.

[46] M. F. Schwartz, A. Emtage, B. Kahle, and B. C. Neuman. A comparison of Internet resource discovery approaches. *Computing Systems*, 5(4), 1992.

[47] Y. Shoham. Agent-oriented programming. *Artificial Intelligence*, 60(1):51–92, March 1993.

[48] D. Terry, D. Goldberg, D. Nichols, and B. Oki. Continuous queries over append-only databases. *SIGMOD Record (Proceedings SIGMOD Conference, San Diego)*, 21(2):321–330, June 1992.

[49] A. Tomasic and H. Garcia-Molina. Caching and database scaling in distributed shared-nothing information retrieval systems. In *ACM SIGMOD Conference*, pages 129–138, Washington, DC, May 1993.

[50] Brygg Ullmer. Physically-modelled environments for dynamic presentations of abstract information. Technical report, Interval Research Corporation, 1994. In Preparation.

[51] M. Weiser. Some computer science issues in ubiquitous computing. *Communications of the ACM*, 36(7):74–84, July 1993.

[52] D. Wheeler. Computer networks are said to offer new opportunities for plagarists. *The Chronicle of Higher Education*, pages 17, 19, June 1993.

[53] G. Wiederhold. *File Organization for Database Design*. McGraw-Hill, New York, 1987.

[54] T. Winograd. A language/action perspective on the design of cooperative work. *Human-Computer Interaction*, 3(1):3–30, 1987-88.

[55] T. W. Yan and H. Garcia-Molina. Index structures for information filtering under the vector space model. In *Proceedings 1994 International Conference on Data Engineering*, Houston, Texas, February 1994. May be obtained via Anonymous FTP from db.stanford.edu in directory /pub/yan/1993/sdi-vector-model.ps.

[56] H.W. Zais. Economic models to evaluate SDI pricing performance. In *Proceedings of the International Symposium on Technology for Selective Dissemination of Information*, pages 35–40, San Marino, September 1976.