IBM Research Report RJ 9840, May 1994.
Submitted to IEEE Computer Society Press, *Proc. Workshop on On-line Access to Digital Libraries*

Digital Library: Gross Structure and Requirements (Report from a Workshop)

Henry M. Gladney
IBM Almaden Research Center
San Jose, California 95120-6099
Internet: gladney@almaden.ibm.com

Nicholas J. Belkin
Rutgers University
New Brunswick, New Jersey
Internet: belkin@pisces.rutgers.edu

Zahid Ahmed
San Diego Supercomputer Center, Univ. of Calif.
La Jolla, California 92093-9784
Internet: ahmed@sdsc.edu

Edward A. Fox
Virginia Polytechnic Institute & State University
Blacksburg, Virgina 24601-0106
Internet: fox@fox.cs.vt.edu

Ron Ashany
National Science Foundation
Arlington, Virginia 22230
Internet: rashany@nsf.gov

Maria Zemankova
Mitre Corporation
McLean, Virginia 22102
Internet: mzemanko@mitre.org

Abstract: The overall workshop goals were: to define those digital library parameters which especially influence issues of access to, retrieval of, and interaction with information; to identify key problems which must be solved to make digital library service an effective reality; to identify a general structure or framework for integrating research and solutions; and to propose and encourage specific, high-priority research directions within such a framework.

We report the deliberations of a subgroup focusing on distributed data service taxonomy and digital library system requirements. After a prelude delimiting what we mean by *Digital Library* (**DL**), we suggest that the evolving *distributed resource manager* and *application enabler* concepts provide an organizing basis for offerings from which each data custodian and each end user can choose modules suiting his needs and preferences. We make a start towards identifying module classes and pointing out existing and needed requirements analyses.

The opinions, reflections, and ideas presented in this paper represent only the co-authors′ individual and collected thoughts, and should not be construed to represent views of their respective organizations.

CONTENTS

Introduction

Digital library (***DL***), also called *electronic library*, has suddenly changed from the relatively obscure concern of a few people in computer science and in library disciplines to a popular topic for many research groups[Fo93b,Fo93c], with the prospect of productive applications in the near future. Commercial, academic, and public interest are fueled by U.S. Government interest led by Vice President Gore, under the *National Information Infrastructure* label, and the national press, under the *Information Superhighway* slogan. Between November 1993 and February 1994, at least four topical conferences were announced for this area, which had seen no similar calls for papers before that.

The participants of a March 1994 workshop at the annual Conference on Artificial Intelligence for Applications (CAIA) in San Antonio agreed it worthwhile to document its deliberations, notwithstanding their tentative nature, as a starting point for similar discussions in other 1994 conferences. In addition to plenary sessions, the work group included meetings of subgroups directed at:

1. Digital Library Models, Frameworks, and System Requirements
2. Library Sciences and Automation
3. Information Retrieval, Organization, Navigation - Tools and Paradigms
4. Digital Library Specific Nomenclature, System Integration and Architecture Issues
5. Interfaces to Digital Libraries - Information Delivery and Presentation Issues
6. Role of Knowledge Representation Systems in Digital Library Interactions

This paper reports opinions shared in the first subgroup, drawing on elements of the plenary session, and including refinements generated as we prepared the manuscript.

Bounds of the Topical Area

The ***DL*** topical area currently lacks clear boundaries with related areas and internal taxonomy, at least in the sense of large group consensus about these factors. We cannot discuss system requirements without tackling these topics, and without straying into the other subgroup focal topics, particularly that of the fourth subgroup. Nor can we find a logical order for the discussion and report – an order that avoids forward references and assumptions. We therefore abandon any attempt to do so, and focus on what we mean by digital libraries, on a system taxonomy for distributed data services, and on system requirements in that order, trying for clarity rather than any form of completeness or precision. Hopefully what is presented is a useful start for iterative refinement.

One may take either an expansive or a narrow attitude about the boundaries of the topic at hand. Retrospection suggests an unspoken and unresolved difference of attitude among ***DL*** workshop participants about this. We give scant attention to borderline topics below; for instance, we treat electronic publishing as an outside topic even though a future library enterprise may well include publishing material derived from its own unique contents, as the U.S. Library of Congress is already considering doing. Some workshop participants tended to include topics at the periphery as part of the ***DL*** topic; others preferred to exclude peripheral topics whenever possible, especially topics already being tended by other interest groups. Such tension is natural because the technology potentially blurs organizational, human role, and task distinctions that evolved in a world of books and book-like materials. Our report deals only with aspects that distinguish ***DL*** services from other topics. This approach is not intended to imply a consensus among the authors, but rather is a practical way to control the report length and to focus attention on topics less thoroughly addressed elsewhere.

Interplay between Commercial and Academic Activity

To understand what ***DL*** technology is already available and what needs attention, it is helpful to remember that in this area commercial activity began more or less at the same time as academic activity. Commercial products already address some component classes identified later in the paper. The commercial presence is signaled by promotional material[1] which claims:

> "[X] offers the most advanced information management system available today. [It] is ... for large integrated collections of textual and image information and ... even an inexperienced researcher can use it effectively the first time.
>
> "[X] meets the growing need for document storage and retrieval through integration of optical storage, scanning and optical character recognition, powerful but inexpensive desktop computers efficiently networked, [etc.]. [X] technology is currently being used by:

- Kiosk systems for public information
- Public utilities to manage regulatory correspondence
- Law firms for litigation support and ... knowledge systems
- Newspapers for electronic archives
- Engineers for their technical documentation
- Pharmaceutical firms to support approval testing
- Law enforcement agencies
- Publishers of scholarly journals for CD-ROM distribution
- Universities for full-text books, multi-media manuscripts, periodicals, and catalogs
- ... ...

> "[X] features include:

- Intelligent search and relevance-ranked retrieval
- Heuristic, expert search assistance
- Text, data, image, and video-clip support
- Hypertext and access to external collections
- Access via fill-in-the-blank forms and [other common paradigms]
- Printing, sorting, access control, session logging, and [other utility functions]
- Call interfaces for embedded reuse in other applications
- Portability across many operating systems
- ... ... "

Inspection of any extant ***DL***, especially with a particular application and user class in mind, readily exposes functional shortfalls. As tempting as it might be to focus on a specific instance, e.g., publication of physics manuscripts[An91], this would not be enough. Since our intention is to be comprehensive and global we must, starting with this paper, encompass the requirements extant systems address and also enough of their architectures and data representations to avoid islands of automation. For this reason, and because of practical constraints and limited current knowledge, our report proposes broad taxonomy while avoiding detailed architecture and identifies essential qualities of requirements analyses by example and citation.

In support of this objective, we mention a few specific academic and archive library efforts which illustrate the range of applications that must be considered, without implying that the particular works cited span the range or are a comprehensive source for requirements analysis.

A Dream – Libraries for Scholars

We intuitively grasp what people like ourselves are looking for; we dream[Gl89] of efficiently solved frustrations:

---

1 We've simplified this quotation and made it anonymous because similar material occurs in other sources and because the current report neither can nor should evaluate projects or products.

> "My office overflows with paper, periodicals, and books; there is more at home. I know what is there, but still have difficulty finding things, even when I have a vivid recollection and terms of reference. From time to time, I reorganize the collection, but its organization always seems wrong for the problem at hand. Of course I could create an index, but building one of sufficient quality would take too long. Keeping a bibliography of the papers that I might cite is all I can afford.

> "Most of what is in my cabinets originated in a computer. Amended ways of exchanging information could provide access to the encoded sources, but it is unrealistic to hope for this on a large scale in the near future. In the meantime, conversion and indexing of the paper is the most costly step for me – not to be undertaken if it intrudes too much. Ideally this would be entirely a machine process, but for my own work there is an alternative. I inspect and mark things as they arrive. If a machine dialog were available for marking – a dialog that did not slow down reading much – I would use it instead of colored pens.

> "My need extends into my colleagues' offices. I remember a report shown by D. six months ago – it is about image technology, is formatted by Janus, and has a system schematic on the third page. I wait half a day until D. is accessible, and ask him. After 10 minutes search, he smiles ruefully, 'I know I had it here somewhere.' Later in the day, he brings it. In the meantime my train of thought is no longer fresh. I am embarrassed because D.'s effort has been larger than the inquiry merited.

> "My need also extends to university libraries and other institutional collections. Some of my questions seek relationships between things in my personal library with things in other libraries. I cannot predict far in advance which university, company, or public library might hold the material of interest. I refuse to learn a new tool for every collection, and feel that such reluctance is reasonable."

Such problems motivated the RightPages™ experiment[St92]. The current paper is another small step towards realizing the dream.

What is a Digital Library?

What is a digital library (***DL***)? There are many buzz-words for related activities, including, but not limited to: *multimedia database*[Wo87], *information mining*, *information warehouse*, *information retrieval*, *on-line information repositories*, *electronic library*, *operational image applications*, *imaging*, *world wide web (WWW)*[Ni92,Ha94,pp.495-512], and *wide area information services (WAIS)*[Ha94,pp.476-493]. How many distinct activities does this list represent? What distinctions are essential, if any? What distinctions are more matters of marketplace focus than technical? Given some topical taxonomy, what requirements differ from topic to topic? Clearly there are too many topics in the list, with too much overlap of related activities, and entirely too much rediscovery of what is already known.

The NSF/ARPA/NASA *Digital Library Initiative, FY 1994*[Na93] states:

> "Information sources accessed via the Internet are ingredients of a digital library. Today, the network connects some information sources that are a mixture of publicly available (with or without charge) information and private information shared by collaborators. They include reference volumes, books, journals, newspapers, national phone directories, sound and voice recordings, images, video clips, scientific data (raw data streams from instruments and processed information), and private information services such as stock market reports and private newsletters. These information sources, when connected electronically through a network, represent important components of an emerging, universally accessible, digital library."

In a prior ***DL*** workshop report[Fo93,p.65], we find:

> "A digital library is a distributed technology environment which dramatically reduces barriers to the creation, dissemination, manipulation, storage, integration, and reuse of information by individuals and groups."

The former quotation, asserting inclusion of all network-accessible information combined with things not yet mentioned, defines ***DL*** expansively enough to stymie prioritized choices of action. The latter quotation attempts a narrower definition, but does not proceed far enough to distinguish ***DL***s from other data collections, i.e., does not teach how to recognize a ***DL*** when it is presented.

The relationship of ***DL***s to libraries of the conventional or traditional sort is clearly of some interest. In particular, we should inquire whether there are essential differences between these two services. If, as some people have suggested, there are few commonalities, we might ask why the word “libraries” was tacked onto “digital” at all. If, on the other hand, there are some differences but also significant commonalities, we should try to understand how such differences affect the transfer of experience (knowledge, practice, techniques, theory) gained in the context of conventional libraries to that of digital libraries. If we find that there are no essential differences between the two, then we need to think about why we need the term “digital” at all, and what it implies for changes in the way that things have been done in conventional libraries.

The function of conventional libraries has been described as four-fold: collection; organization and representation; access and retrieval; and analysis, synthesis, and dissemination of information. Collection includes techniques for understanding what information resources are of use to a client population and for cost-effective storage and preservation of such resources. Organization and representation have to do with classifying and indexing information resources in ways relevant to their potential users. Access considerations include design of physical space and organization of materials within such space to respond effectively to user needs and expectations. Information retrieval has been addressed, of course, in the design of systems specific to that task. Analysis, synthesis and dissemination functions include responding to reference questions, producing evaluative reviews, and devising community outreach programs. Librarians and information scientists have developed techniques, procedures, and systems for addressing each of these functions for many kinds of data and presentation.

It is hard to imagine that a ***DL*** would add many, if any, qualitatively new roles to these, nor does it seem likely that anything that people would call a ***DL*** would omit any of these roles. Although ***DL*** implementation may well depend on the local context and technologies chosen, there are commonly accepted constraints which must be maintained to please users. For instance, any of us can enter a conventional library in New York, in Stuttgart, or in Hong Kong and confidently expect to find what we want almost as quickly in the foreign library as we do in a familiar one. Somehow this property of conventional libraries must be achieved in digital library systems.

DL research has focused on automating activities carried out by librarians, such as automatic indexing and classifying and expert systems for reference desks. Conventional cataloging presumes people are involved and can only assign a few keywords; digital catalogs can support long keyword and key phrase (or even word sense) sets with weights, long user queries, ranked retrieval, etc. Information search via hypertext illustrates that indices can be implicit rather than explicit, giving users a seamless blend of primary and secondary works. Further, some current library activities may become irrelevant; for instance, circulation problems originating in a fixed number of copies of each work simply disappear. We might redefine and redesign library services to achieve the basic aims more effectively than is possible now. Thus ***DL*** involves not only automation of each traditional library activity and service, but also calls for redefinition of services, new groupings of services or replacements of groups of services with other solutions.

Whether one judges the differences between digital and conventional libraries to be intrinsically qualitative or merely quantitative, the quantitative factors are so immensely different that it will often be better to treat them as qualitative. For example, although photocopying led to problems for conventional libraries similar to problems anticipated for digital libraries, the ease and speed of digital copying and redistribution (what lawyers call *fungibility* of the commodity) suggest qualitatively different treatment. On the positive side, the potential ease of integration of information in different modes from mutually remote repositories creates new social possibilities; for instance, that the California death certificates are now being collected into a digital library enables a test of the conjecture that shipbuilding in 1940-5 led to premature deaths from asbestosis. So there are clearly new problems and new possibilities for which we should seek new answers, develop new techniques, and create new institutions or significantly modify old ones. But we see no reason to believe that the novel elements constitute the entire digital library service, and feel we should be careful to capture traditional values.

To some readers, these comments might appear to belabor the obvious. They were, however, not uniformly agreed among the workshop participants, and are probably still not fully accepted.

In view of the foregoing discussion and because the area is sufficiently new that consensus has not yet been reached about its content or the answer to "What is a digital library?", we are in fact free to assert how the term *digital library* should be construed. To prevail, such an assertion must be sufficiently close to what people expect, and must be repeated often and loudly. We therefore assert:

> A ***digital library*** is a machine readable representation of materials which might be found in a university library together with organizing information intended to help users find specific information. **A *digital library service* is an assemblage of digital computing, storage, and communications machinery together with the software needed to reproduce, emulate, and extend the services provided by conventional libraries based on paper and other material means of collecting, storing, cataloging, finding, and disseminating information**. A full service digital library must accomplish all essential services of traditional libraries and also exploit digital storage, searching, and communication.
>
> Public, private, professional, school, commercial, and other kinds of library emphasize different services, different kinds of information, and different service styles. While any digital library instance may thus offer only partial services, the technology suite from which library instances are assembled must permit assembly of a full service library. In addition, this suite must shield the user who wishes to draw on multiple libraries from inter-library differences which are irrelevant to him.
>
> What distinguishes a conventional library from a heap of things to read is organization provided by someone other than the authors of the collected materials. For a small, private collection this could be shelf organization; for a large collection it is typically a descriptive catalog[2] which is distinct from the collection, with at least one catalog record associated with each item held.
>
> Not every database is a library, but every library is a database[3]. What distinguishes a library from an arbitrary database are certain data integrity and security rules that constitute an implicit contract between custodians and users.

2 Here, a *catalog* is a set of records which might include the kinds of information found on traditional 3″×5″ cards. Organization can also be provided by citations, whose analog in a digital system are hypermedia links. Usually embedded citations are provided only by document authors, whereas catalog records are provided by librarians or other agents.

3 Here and throughout the report we use *database* when other authors might use *information collection* or *knowledge database*. We do this to avoid connoting any semantics because, as we see it, the lower layers of digital library software neither need to nor should exploit the content or format of the data stored and communicated.

A few circumstances and characteristics for which we expect ***DL***s to emulate conventional libraries holding books, pictures, and other material objects communicate the flavor intended:

- users are usually elsewhere than the information they want, and often want to correlate items from several sources;
- whoever wants to use a library must show permission to do so;
- different patrons are permitted different actions and to see different parts of each collection;
- the catalog and the collected items are used differently and not necessarily housed in the same place;
- to find specific information, each user must understand the catalog structure;
- the catalog may describe items in other libraries;
- documents contain cross references to other documents;
- documents are cataloged with text descriptors and also with conventional properties, such as author names;
- document identifiers are different from document names; a document may have several names, one for each context, e.g., “Tales of Hoffmann” in English, “Les contes d′Hoffmann” in French, and “Hoffmanns Erzählungen” in German;
- translations of a document may express essentially the same information, e.g., versions of classic literature in different languages;
- each stored item is valuable, often with part of its residual value owned by its authors or authors′ assignees;
- part of the value provided by a library is the provenance information it holds for each item;
- items are put into libraries because, while each is thought valuable for future reference, the specific individuals who will read it and the times when this will occur are not known.

The advantages of a digital library over a paper library are similar to those of any digital database over its paper counterpart: faster addition to the data collection, improved browse and search functionality, faster distribution from the points of creation and storage to the point of usage, better history tracking, finer granularity of control, and enhanced plasticity of its content. The benefits of improved control and plasticity are not only improved data quality, but also more freedom and reduced bureaucracy for individual users. Only a librarian may add to the collection of a conventional library, because of the discipline essential to create a quality catalog. In a digital library, cataloging discipline and search restrictions to authorized data can be automatically enforced. It can thus allow each patron and author a wider range of services than is practical with a conventional library.

Taxonomy for Digital Libraries

A “complete” library service will contain many components from which each installation selects a subset and each user draws on an even smaller set. We need a distributed computing infrastructure and a framework for such components. Part of such a framework is provided by the concepts *resource manager* and *application enabler*, which are well known to architects of distributed computing services[4]. Since they seem to be outside the experience of at least part of the digital library community we summarize them below. The concept of a resource manager will be seen to embrace notions from

4 We don′t know the provenance of these ideas, having learnt them from writings and presentations by product developers, but believe they come from DCE/DME deliberations[Ku91].

object-oriented computing and from client-server computing. A very similar notion has been implemented in such advanced information retrieval systems as CODER[Fo87,Fo87b]. Given suitable operating system and communication services, all distributed computing services can be built as a set of application programs, application enablers, and resource managers, with only the resource managers directly invoking operating system and communication services.

Resource Managers and Application Enablers

A *protected resource* is the combination of a persistent data collection and a set of programs which define its semantics. To define the data semantics with absolute integrity, this program set must constitute the only access path to the data; it is called a *resource manager*. Services such as authentication, file subsystems, network directory services, database management systems, and digital library components can all be constructed as resource managers.

We have in mind a network of mutually supportive resource managers, each providing a relatively specialized service. Each resource manager distributes itself for remote applications and accesses any needed sibling acting as a user, i.e., using the sibling's client interface. Whether sibling service is local or remote is transparent; exploiting propinquity is regarded as a network optimization issue.

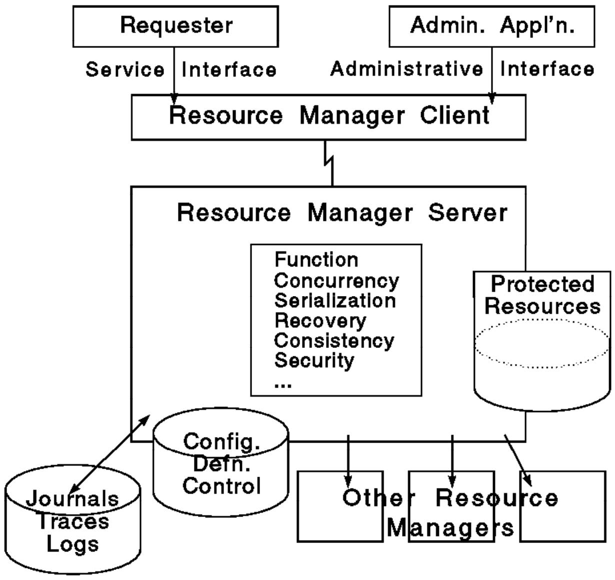

Figure 1. Resources and resource management in a computing service network: any encapsulated database can be contained in a set of resource managers; the storage subsystem of a digital library service is one example.

A resource manager is a service which combines state and processes and is accessible to multiple, concurrent clients (Figure 1). To qualify and be used as a resource manager, the program set and the data it manages (the protected resource) should satisfy the following criteria:

- The resource manager programs provide the only access path to the protected data, and therefore define and implement its semantics. (Practical systems always permit someone to bypass this proper access path, e.g., for data backup and recovery; alternative paths need to be protected by physical and administrative means if the data are to be safe.)

- Typically, the protected data are highly structured, possibly consisting of well-defined objects. Typically each protected resource consists of many such entities, called *items* below[5].
- The resource manager provides distributed access, doing so by having client and server portions connected by a private protocol. That the protocol is private distinguishes the resource manager approach from one based on open protocols, which must be standardized to be effective.
- To the extent consistent with achieving good performance and with practical aspects of software production and distribution, each resource manager avoids reproducing services that it can efficiently get from other resource managers. For instance, a library catalog manager could exploit a database manager, invoking it just as any other database manager client would.
- A resource manager often serves as an *access control enforcement function (AEF)* between a request initiator and a target, in the sense called for in international standards[Is88].
- As well as access control, a quality resource manager provides various data integrity protections, such as those called the ACID (Atomicity, Consistency, Integrity, Durability) properties[Gr93,p.6].

Thus each service instance encapsulates its own data within a cocoon—a form of object-oriented programming which is not necessarily bound to any particular programming language. There typically will be many instances of each kind of protected resource, with its associated resource manager defining the resource class, e.g., Network File Systems (NFS), DB2 databases, X.500 directories, X-windows services. Library content is a protected resource in the sense intended, and the library procedures defining access to the content constitute a resource manager. Resource managers are generic services.

Not all generic services need to provide network distribution or to hold data if they can do so indirectly by invoking resource managers. Services such as editors, filters, formatters, and other generic software constitute a class called *application enablers*. The purpose of such enablers is to make application programming easy and quick, or, optimally, avoidable entirely. Just as resource managers can be modularized by having each exploit other resource managers, application enablers can be cascaded. Figure 2 on page 9 suggests how applications, application enablers, and resource managers can be layered to exploit layered open communications and to hide irrelevant operating system and machine differences.

5 We use the label *item* here to avoid implying any particular properties, such as object properties associated with object-orientation.

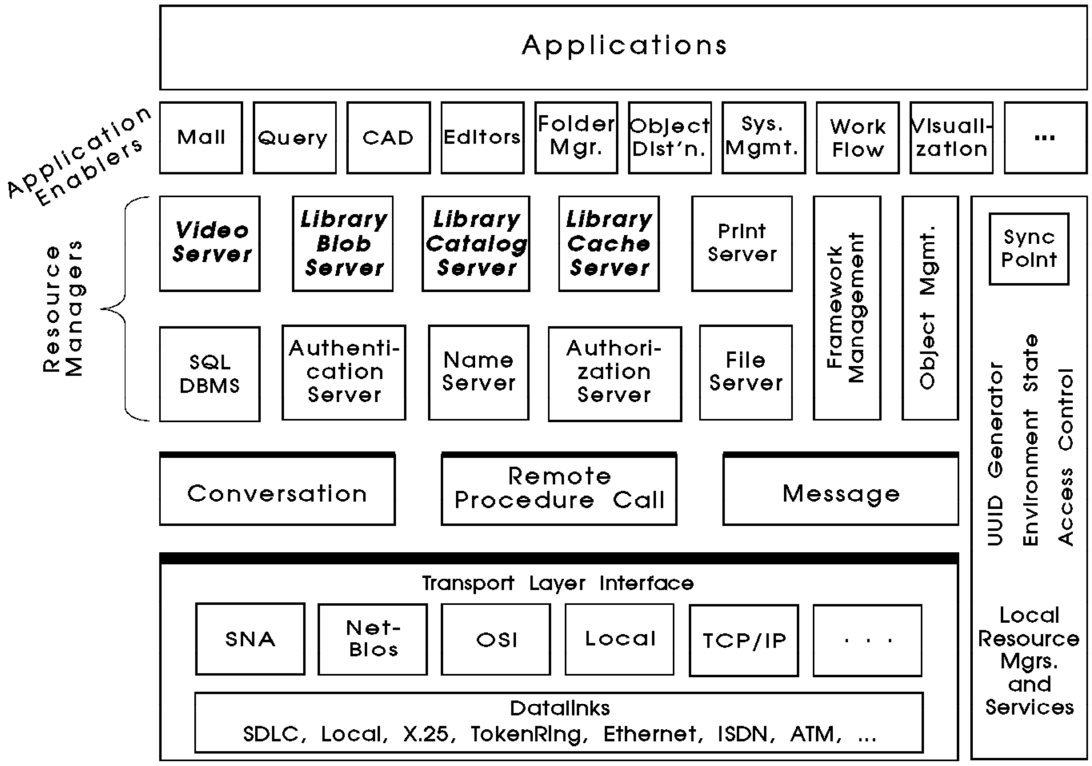

Figure 2. Distributed data service software = operating system software + communications software + resource managers + application enablers: four digital library components are depicted with bold face.

What makes the modularization implicit in this model feasible today (it wasn't affordable until very recently) are the dramatic improvements in digital performance and costs already seen and yet to come. In addition, the *transport layer interface* depicted in the communication portion of Figure 2 allows the lower communication layers to choose efficient paths independently of how each resource manager calls communications (see Figure 1 on page 7). For example, in one extant implementation[Gl93] the transport layer detects the occasions when the client and server happen to be in the same machine and uses local operating system services for inter-process communications; for the library application, the performance is close enough to what would be achieved by combining the client and server into a single program.

Document Storage Subsystem and Document Servers

Document storage and access software can be realized in two layers above a base of file systems and database managers (Figure 3 on page 10). The lower one is a storage subsystem which stores and retrieves items to and from each library collection, updates and searches library catalog records, and limits who can manipulate which data—giving only services which are identical for all types of documents. Instances of the higher layer, which we call *document managers*, help applications or end users with their special kinds of documents and varied forms of presentation and manipulation.

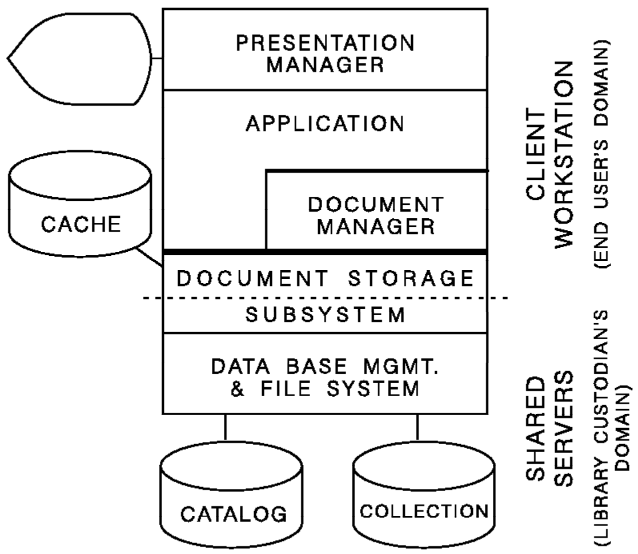

Figure 3. Partitioning of library service software: the document storage subsystem layer provides services needed by every kind of library and every data model; each of potentially many document managers implements a model such as hypertext or a service wanted by many enterprises. Application programs are workstation programs; the storage subsystem embeds needed inter-process communications.

The distinction between the document storage subsystem layer, which would be implemented as a set of resource managers, and document managers, which would be implemented as application enablers, deserves more careful articulation. Part of its reason is that we expect the storage subsystem layer to be difficult for most users to change or substitute, and document managers being made relatively accessible and malleable.

The storage subsystem limits its services to aspects which do not depend on the meaning or representation of items. With at most limited exceptions, items it delivers to requesting applications are faithful copies of items other applications stored[6]. The storage subsystem manages data placement and replication, implements custodial responsibilities for data security, and hides irrelevant network and other environmental dependencies to the extent possible. It is convenient to think of its application programming interface in three parts: a query interface, intended primarily for identifying items of interest to a browser, should allow whatever inquiries do not violate item owners′ confidentiality desires; a retrieval interface should deliver items with whatever timing and buffering is consistent with the data kind at hand and with the user′s responsiveness and cost objectives; and update interfaces for the library catalog and collection should enforce articulated policies for library data integrity and quality. Since searches for information may well depend on databases that are not part of what the librarian has chosen to include in the formal library catalog, the storage subsystem should also support queries which combine internal and external databases[7]. In summary, the document storage subsystem provides retention and catalog services and manages inter-machine communications, hiding them to the extent possible. Its implementation follows a client-server approach.

6 Partial document access is commonly wanted, and sometimes transformations which improve presentation without adding information are valuable.

7 This need is not commonly recognized, and is very difficult to satisfy as broadly as library patrons will want.

To provide enough flexibility for all possible applications, the document storage subsystem interface is likely to have many primitive operators, making it somewhat difficult to program for ad hoc applications. This can be overcome with document managers which implement broadly interesting information access methods. For example, we see Mosaic[Ha94,p.510] as a document manager. The storage subsystem attempts comprehensive coverage of functional requirements in its domain; good document managers would offer less flexibility and fewer options, but would be much easier to explain and understand.

More generally, document managers give those services which vary among access occasions because different document types need different presentation and manipulation and because users have different objectives and preferences. Document editing, transformation, combination, and presentation, and other complex manipulations are conveniently implemented as document managers. In the architecture suggested by Figure 3 on page 10, document managers execute in users′ workstations.

Thus, in a practical system, each of many document managers embodies a *document model*—the set of concepts that create the digital analog of some collection of papers or other physical objects, or some information network invented for a particular application, such as hypertext[Ha92], or some flow of documents through a series of routine steps. In contrast, the document storage subsystem layer avoids modelling. Typical document managers interpret scanned data to create catalog entries automatically, manage interrelationships among documents, facilitate the most common search methods, and help move information among workers. For instance:

- A folder manager might include a scanning service for memoranda, letters, contracts, and financial records; such a manager would extract names, addresses and dates to cross-index information received[Ma87] and associate each document with a folder.
- A second document manager might manage moving pictures; it would communicate with its users in terms of movies, reels, and frames and depend on a storage subsystem with video delivery channels.
- A third document manager might feature CAD and be applied to maintenance records of university buildings; it would generate and display building plans with a graphic editor and maintenance contracts with a customized text editor.
- A fourth document manager might model what is found in a university library—books and pamphlets with individually viewable pages, loose collections of papers in folders, manuscripts, video tapes, and so on.

Generic document managers for applications like geographic information systems, and enterprise-specific ones administering conventions and document quality standards, may evolve over time. While a good document manager would support most library services in its domain, the storage subsystem interface is exposed to allow other applications to bypass document managers.

Component Classes in Modular DL Toolkits

Even within a group of similar library applications (e.g., welfare case management, university geography department collections, digitized rare book collections), no two libraries have a sufficiently similar set of needs that a monolithic library software offering will satisfy many people. Individual users and individual custodians will have evolving views of what they want. We must define and design components which each enterprise, and to some extent, each user can select and combine for himself.

The software that creates ***DL***s will include at least the following module classes. Here we say "module classes" because each tabulated item in the list may be represented by several implementations to create different look and feel, or to provide different data transformations, or for different hardware and operating system platforms.

A document storage subsystem is a resource manager that creates a library abstraction implementing all the essential data storage, retrieval, protection, communication, and search functions as primitive operations which do not depend on the interpretation of the data handled. It is middleware that integrates more basic resource managers, such as ***database managers***, ***video servers***, ***cache managers***, and ***file managers***.

Network directory and security servers are intimately related resource managers which help users locate protected resources, limiting each user to what each resource and item owner permits[Ja92]. A ***name*** or ***directory service***[Is89,Za92] maps names to locations and other descriptors. ***Authentication***[La92] and ***authorization services***[Ab93] combine to control access to what is permitted.

A billing subsystem is a component or set of components which collects and validates subject and account information, rate information and resource consumptions, and submits these to a charge-back subsystem. We do not yet understand billing subsystems sufficiently to classify them as resource managers, application enablers, or combinations, or to say to what extent they can draw on other distributed computing subsystems.

A source selector and fuser is an application enabler which partitions a query among as many libraries as the query implies, calls on query managers to execute the queries, and assembles the results to hide those inter-library differences that the user considers irrelevant[Be94].

A search engine is a resource manager which accepts a query and returns item descriptors, but not usually copies of the items themselves. There may be primitive search engines that operate only on certain kinds of database (e.g., SQL relations, inverted text indices, ...) and also more complex search engines which partition queries among simpler query engines and combine results using joins and selections which are beyond the simpler engines. Part of a search engine might be an external ***query optimizer*** intended to overcome the performance problems inherent in multiple, separated library catalogs.

A filter service is a separately programmed bit-stream to bit-stream transformation that can be linked into a resource manager or an application enabler for functions like encryption, compression, and partial object access.

A link engine is an application enabler which interprets a hypermedia[Gr94] link and calls on resource managers to manipulate a copy of the item indicated, possibly by launching an separately-provided application program, and possibly invoking library services to provide integrity holds on the primary copy of the item.

A preview/thumbnailer is an application enabler which collects from one or more libraries a set of small data objects (“thumbnails”), or creates each such thumbnail from a library item, to present these to a user for selection. Each thumbnail remains bound to the associated item so that selecting it can be used to drive item retrieval or some other action.

A presenter or renderer is an application enabler which prepares an object for manipulation in a client machine, possibly assembling item parts from one or more libraries. Each tool is specific to an object class, i.e., is selected when the user asks to render an item of the class at hand. There is no clear boundary with a ***filter*** or a ***format converter*** which edits a set of files to create another set of files representing a subset of the same information, typically executing close to the data source or sink.

Data analysis tools, browsers, navigators, and authoring and editing services are application enablers.

Source/sink servers are resource managers such as scan servers, print servers, and fax servers.

Indexing tools, document analyzers, and other tools to recognize patterns and structure are application enablers for creating search indices automatically.

An actor or active agent is a persistent process which monitors database state and sends to principals filtered information about changes. Alternatively, an active agent is a network process which creates other active agent instances at remote locations, with each instance collecting filtered information from databases.

Notice that actors are neither resource managers nor application enablers. Resource managers and application enablers are essentially passive and cannot distinguish active agents from human users[8]. Triggers, a database feature in which an incoming message causes not only database changes and/or responses to its source, but also messages to a third party, share some of the characteristics of actors. Both triggers and actors are dependent on store-and-forward network services (e.g., electronic mail) because the message target may not be active when the message is ready for delivery.

Some of these components can and must be adopted from activities not labeled ***DL***. Such components certainly include those for naming, authentication, and many kinds of search engine.

Requirements Analysis: What Exists, What is Needed?

Any social unit (school, business, department, family, individual, ...) might create and manage its own library, and most individuals will want access to many libraries. All libraries should do certain things similarly—adhere to certain standards—so that people do not need to learn new methods for each library and so that information can be exchanged.

At the vague and general level found in requests for proposals, in the trade literature, and in business publications, there is broad consensus on what services the ***DL*** should have in 5-10 years. Sensible 2-year objectives are not equally obvious. For a few generic components, such as storage subsystems and document markup languages and interpreters, detailed needs analyses exist; each of these is characterized by hundreds of well-justified requirement statements. For most generic components suggested above, however, similarly comprehensive requirements analyses are not available in the generally accessible literature.

8 Distinctions between human clients and agents are important for principal authentication and resource authorization. The meaning and safety of “on behalf of”, ”speaks for“, and related issues are current research topics[Ab93,La92].

To prioritize the needs for academic, museum, and archive ***DL*** services we must consider a range of objectives which will differ among different institutions and for different kinds of collection. For some libraries, the premier objective will be improved accessibility to rare and valuable materials for scholars. For other libraries, it will be out-reach into troubled elementary school districts. We must start by characterizing each community we are trying to serve, what it wants and can handle, and the nature of the collection. A few existing projects suggest the range:

- In the TULIP project mounted by Elsevier in partnership with material science groups at several universities, the objective is rapid communication of recent research activity.
- In an IBM/Case Western Reserve University Project[Ba92], the library is intended to be a component of an "electronic learning environment", as part of a campus-wide information system.
- The Carnegie Mellon University Mercury project extends this by hoping to make information available "in specific disciplines ... as part of a national electronic library"[Ar92].
- In the Brown University Intermedia project[Ya88], an objective was a more narrowly conceived repository for interactive instructional materials.
- The Sequoia project[Ko93,St91] intends to create a quite specific tool – the data management component of a massive scientific investigation.
- The Library of Congress American Memory project[Cu92,Po92] intends broad public access to unique material.
- A Cornell University project under the auspices of the Commission on Preservation and Access[Ke93,We93] intends rescuing the content of fragile materials.
- Project Envision at VPI&SU[Fo93d] focuses on information retrieval and interface capabilities with structured and reusable objects, as well as on technologies for learning about computer science with materials from ACM, IEEE-CS, and others.
- A joint project of the Vatican Library, the Pontifical Catholic University of Rio de Janeiro, and IBM intends world-wide access to images of rare, historically significant materials such as a beautifully illustrated copy of Dante′s *Divine Comedy* and the four oldest surviving manuscripts of Virgil′s poems.

Such examples make it clear that part of what we need to do is compile requirements statements from a sufficiently large and diverse application set, in order to distinguish common from unique elements. Accomplishing this is a large enough task to warrant an orderly engineering approach, as illustrated[Gl90] for ***DL*** middleware; a helter-skelter approach, in which many individual, marginally-correlated projects decide requirements independently, is not enough. Orderly tabulations will reduce wasteful duplicated effort and will help us distinguish what is already available, what is merely a matter of technical development and/or deployment, and what deserves attention and research funding because it is both difficult and needed. Comprehensive requirements statements will provide a substratum which helps each project decide what it can reasonably expect to achieve for its clientele and what its contributions to ***DL*** technology should be.

Money for Information: Public and Commercial Policy

The issues of ownership of information, the benefits of rights to information, and whether, when, and how money should flow from information users towards information originators can neither be ignored nor be properly handled in this report. Browning[Br93] provides a commentary on some key issues:

> "... the Library of Congress is just beginning its own budgetary debate. It is now forbidden by law ... to charge more than the cost of reproducing documents, plus 10 percent. ... But building [CD-ROM] publications combining several media – marrying, say, Civil War photographs with letters ... from the ... vast collection requires heavy research, the costs of which the Library cannot alone recover. So the Library has asked for legislation overturning the 10-percent restriction, but the proposal has run into controversy.
>
> "The Information Industries Association, representing publishers, fears that the legislation will unfairly set up government-subsidized competition. The American Library Association, meanwhile, fears that the legislation will set libraries on a slippery slope that will lead to the elimination of free services. ...
>
> "If libraries do not charge for electronic books, not only can they not reap rewards commensurate with their own increasing importance, but [they] can also put publishers out of business .... If libraries do charge, that will disenfranchise people from information – a horrible thing. ... It is not really satisfactory either to cripple the technology so that ... texts cannot be stored, or to divvy information into two categories: the free (paid for by the taxpayer) and the commercial (paid for by the consumer)."

It is clear that these issues are very difficult, that they will be hotly debated for some years to come, and that librarians and technologists have no persuasive claim to a special voice in determining the outcome. It is equally clear that the resolution will be different in different political jurisdictions and for different classes of information, e.g., depending on whether copyrights are expired, whether the information is raw data or organized, summarized, and analyzed, and so on. What may not be obvious is that it is feasible (but possibly technically difficult) to implement ***DL*** services to support all possible policies, including situations in which different items in a collection are governed by different policies. So, for the moment, we content ourselves by asserting such feasibility and identifying as a requirement the ability to handle access to each stored item according to rules appropriate to it.

A nucleus for intellectual property rights management must be built into storage subsystem software in such a way that each data custodian can choose options to implement his institution′s policies. The tracking software needs to be able to determine reliably who to bill and that bills will be paid. For efficiency such software must be intimately related to authenticators, user registries, access control mechanisms, and electronic funds transfer mechanisms. The solution should have least the following characteristics:

- As much or as little interaction as wanted with each user to advise about charges;
- As much or as little administrator interaction as institutional objectives demand;
- Imperceptible data server responsiveness degradation; and
- Sufficient efficiency to collect 10¢ page charges.

The technical work to achieve this is much less than what will be needed to achieve industry conventions and deployment.

Requirements for Document Storage Services

An analysis of the ***DL*** storage subsystem done by IBM Research[Gl90] identified several hundred specific needs—too many to tabulate here. However, several broadly applicable elements emerged, and are summarized below because they typify what needs to be worked out for each library component class identified above.

> **Distribution:** People are often distant from needed information, frequently in locations for which high speed links are not affordable. Such recipients of large items often want delayed delivery at times of day when communication tolls are low.

Performance: Updating a stored document is likely to be a rare event and not subject to stringent responsiveness objectives. In contrast, retrieval should be rapid, and search to identify which items are worth retrieving should be even more rapid. (Giving partial research results while a lengthy search is completed may help satisfy people.)

Large and small items: Item sizes range from about 10^3 bytes for ASCII notes to 10^7 bytes for high resolution pictures. Digital video and audio items are even bigger and must be delivered with controlled pacing.

Accessibility from all workstation platforms: Different people will have different kinds of personal computer because of history, function needed, or personal preference. Increasingly, individuals may use more than one machine type. Library service must be accessible from whatever workstation is chosen for other reasons.

Catalog service from all kinds of operating system platform: A large enterprise may have different kinds of database server in different locations, and should be able to provide compatible library catalog services from these database servers.

Support for all kinds of item storage: Custodians should be able to house items as economically as possible within their operational and policy constraints. They should be able to add capacity using the currently most effective storage medium hierarchy and attach this wherever needed to minimize communication costs and maximize responsiveness.

Low entry point, with growth to giant collections: Library service offerers want to start cheaply and to grow without disruption or breakage to large numbers of users (10^6 registered, with 10^4 active concurrently) and very large databases. There should be no system-imposed limit to collection sizes.

Low administration overhead: Installation and custodial responsibilities for a library should require only a small addition in time and training for data administrators. Installation and use of the workstation portion of library services should be easy given only “shrink wrap” materials, without help from specialists.

Joining libraries to other databases: People want easy use of library data in unanticipated ways, joining library catalogs to enterprise databases and combining data across agencies (e.g., toxic waste data with death certificates) and sometimes across administrative jurisdictions (interstate, county-to-state, ...). People want to do particular correlations on short notice and with low cost.

Application independence: The utility of stored data is hampered by anything that tailors it to one application or one usage paradigm in preference to alternatives. Cataloging documents is economical primarily under the presumption of future pertinence to multiple, unanticipated applications. An application-neutral interface to library services is required.

Open subsystem: Emerging workstation application packages—text, graphic, image, and audio editors, spreadsheets, CAD packages, and industry support packages such as those for hospitals and for doctors′ offices—are potential sources and sinks for large numbers of electronic documents. Document storage subsystems must provide application programming interfaces and exit points for enterprise tailoring.

Standard interfaces and protocols: The previous requirements imply a long term commitment to an application programming interface for library services, and to how the storage system exploits underlying communication services.

Application of Digital Library to the Archival Library Role

DL research attention has been more focused on recently generated materials than on managing the cultural heritage. We need to think about the possibilities and limitations for antique material. This includes recognizing the special virtues of paper as an archival medium (consider 15th century manuscripts), and also its limitations (recall the brittle books problem[Co86] – it is estimated that 11,000,000 titles are at risk).

Requirements for Catalogs

The radically new possibility for the ***DL*** is the storage and dissemination of collected items. In contrast, digital catalogs have been in practical use for some time, and there is a considerable body of experience and some standards in this area[Fo93e]. Instruction and standards are necessary since library cataloging has long been known to be difficult[Je53]:

> "The preparation of a catalog may seem a light task, to the inexperienced, and to those who are unacquainted with the requirements of the learned world, respecting such works. In truth, however, there is no species of literary labor so arduous and perplexing. The peculiarities of titles are, like the idiosyncrasies of authors, innumerable."

Digitally managed catalogs present many of the same problems as card catalogs and some new problems, and the possibility that automatic processes may improve some aspects. Catalog quality is limited not only by linguistic and technical factors, but also by difficult cost constraints exacerbated by ever larger acquisition rates. Catalog quality receives a great deal of attention from librarians, both in their graduate education and as a research issue. Librarians complain that computer scientists are not adequately involving them in ***DL*** deliberations; the criticism is merited.

In the workshop, we did not consider catalog structure, but recommend renewed attention to it, either by resurrecting prior needs analyses and re-examining them for current pertinence, or by constructing afresh something similar to what is available for the storage subsystem[Gl90].

Document Markup, Links, and Interchange Conventions

This topic is critical for documents produced specifically for the digital environment. As this has been realized for some years, markup, linking, and interchange have already received intensive attention, including standards activities and proposed industry conventions. We refer the reader to treatments of the Dexter model for hypertext[Gr94,Ha94a], of SGML and HyTime for standard document markup language[Go90], and to the trade literature for arguments about the merits of Microsoft OLE (Object Linking and Embedding) and Apple OpenDoc[Pi94]. There is considerable overlap among these tools, which are mostly promulgated for personal computers and office applications, between them and World Wide Web markup being popularized in the Internet, and probably between all these and further document markup languages that we have overlooked. In addition, the two standards for document interchange, ANSI Z39.50[Ly91] and ISO DFR[Is91], are mutually incompatible, and have unresolved relationships with the linking conventions.

The ***DL*** community should avoid further redundant activities. In the workshop, we did not consider the extent to which ***DL*** progress depends on the emergence of a limited number of document markup conventions or how the the ***DL*** community should participate, if at all.

Information Capture

By *information capture* we mean everything necessary to import each external information item into a ***DL***, including media conversion, creation of primary catalog records, document analysis to extract secondary indices, correlation with prior library contents, and conversion to a format suitable for navigation by following links and conforming to interchange standards, and whatever else is needed to make the ***DL*** convenient for the human client. Here, "convenient for the human client" is ill-defined; while we might aspire to some automatic generation of hypertext links, preparation of hypertext for instruction[La89] should probably be considered outside the ***DL*** topic (as well as being beyond what is automatically feasible in the near future).

Giant libraries can be achieved only with automatic means of capturing information. The NSF/ARPA/NASA initiative[Na93] calls for "an economically feasible capability to digitize massive corpora of extant and new information from heterogeneous and distributed sources". Information capture arguably presents the most difficult technical challenges of the ***DL*** enterprise and the largest costs of the deployed ***DL*** complex. A National Library of Medicine project[Th85] suggested that digital replacement of existing books is not economical, at least not yet; its objective was to make its resources available to practicing physicians, but it encountered an early obstacle in the cost of turning the pages of each book. So that dream is not yet realized[Si91].

Information capture from existing sources is itself a thriving field of inquiry, summarized by the ICDAR Conference Chair[Ya93]:

> "The large number of existing paper-based documents and the production of a multitude of new ones every year have raised the important issues of efficient handling, retrieval and storage of the information these documents contain. This has led to the emergence of new research domains dealing with the recognition by computers of the constituent elements of documents – including characters, symbols, text, lines, graphics, images, handwriting, signatures, etc. In addition, these new domains also deal with automatic analyses of the overall physical and logical structures of documents, with the ultimate objective of a high-level understanding of their semantic content. ... Automatic, intelligent process of documents is at the intersection of many fields of research, ... This second conference ... is part of a series of biennial conferences.
>
> "This year more than 260 papers were submitted. ... Of these 260 papers, 21 were selected for publication as long papers ... and 124 for publication as short papers [from] a total of 24 different countries."

Information Retrieval

Information retrieval (IR), being one of the earliest foci of computer science research, needs few words in this report. There are commercial products which create secondary indices from text bodies. Fuzzy indexing and search seem especially promising.

Morris has reviewed reseach on expert systems for information retrieval[Mo91] and concludes[Dr91]:

> "Current practice cousels expert system development only in narrow, well-defined, homogeneous domains; none of these attributes applies to on-line searching. ...
>
> "... Although they have developed beyond infancy, it seems that information retrieval systems will not reach adulthood before the next century. There is much to be done with regard to standardizing platforms and formats, in developing and integrating information retrieval and AI techniques, ... before consulting an information retrieval system is no more daunting and at least as rewarding as asking one′s personal researcher for 'Details about ...'"

Effective techniques for evaluating IR services and experiments are not yet available. Another topic that has not received sufficient attention, as far as we know, is combining queries over attribute databases with queries over (indices derived from) text bodies; we are addressing it[Be94]. Chaumier surveys recent introductions to the marketplace[Ch94].

Many Libraries and Very Large Libraries

Problems of scale were not considered in the workshop, beyond recognition of their existence. Our treatment is limited in that it considers the topic ***Digital Library*** rather than the topic ***Digital Libraries***, i.e., almost ignores the problem of the user who needs to choose which libraries to search. For an entry into this topic, see Bowman[Bo93]. More recently, Sheldon[Sh94] considers the possibility of content labels to help control the size of query answer sets. Agrawal[Ag93] introduces database mining, which is discovery of what questions a database might help answer.

Scaling of the storage component is not a research problem. Storage management for very large collections is understood in principle; a document storage subsystem (Figure 3 on page 10) can knit together commercial database management and hierarchical file systems sufficient for the largest ***DL***s practical for the time being. We know how to handle 10^8 objects in a library, with a mixture of sizes from 10^3 to 10^7 bytes per object; what is practical will be determined by information capture capabilities and by costs. A fully compatible document storage subsystem has also been demonstrated[Gl93] within a single workstation. Such technology has not yet been applied to academic libraries, and performance challenges such as those which might arise in a multi-campus university with 10^5 students have not been identified, much less solved.

Other Module Classes

Every module class defined above needs to be considered along the lines we have started. For some, such as protection services and means for letting users proceed without knowing actual locations of information resources, requirements analysis is well developed[Ab93,Gl92,La92,Za92]. Others are not so well addressed.

Usability and Public Policy Factors

Recent American Memory field trials[Cu92,Po92] suggest that we should work to deflate extreme public optimism and regard with skepticism technology announcements bordering on hyperbole. For reasons which the field trial managers admit they do not understand fully, the users were not enthusiastic; while the reasons may have to do with choice of subject matter or other factors only marginally related to the technical ones that are our primary concern, it would be a tactical error to ignore such signals. Hard-headed requirements analysis may make disappointing field trials less likely.

Notwithstanding our enthusiasm for what digital library services promise, we feel that glib calls to replace conventional publication entirely must be regarded skeptically. Preserving the cultural heritage (e.g., in archive libraries with 500-year old manuscripts) has been better served by paper than digital means currently promise, and there is little funded work towards remedying this[Co86]. What ***DL*** for scholars will give is richness of online material. But we should not overlook essential qualities – preservation of information and promulgating its authenticity.

Digital access for scholars and other relatively well-to-do clients will not automatically help the population segments most in need of better information access; notwithstanding U.S. Government mandates against widening the gap, pertinent research problems are receiving relatively little attention. While the inherent questions about public policy are outside our scope, we have an implicit responsibility to manage the research agenda so that the choices are manifest. This can be done by identifying how needs differ for different information corpora and for different user communities. This can and should be done for each modular component class identified in the ***Component Classes in Modular DL Toolkits*** subsection.

Conclusions

The potential advantages of a digital library over a paper library are similar to those of any digital database over its paper counterpart: faster addition to the data collection with better quality control, improved search functionality and faster access to information found, but also more freedom and reduced bureaucracy for individual users. Achieving these advantages will depend not only on topics traditionally dealt with by computer scientists, but also on superb engineering for human usability.

The structuring concepts *resource manager* and *application enablers* (which are not new) provide part of a conceptual base for partitioning ***DL*** work into individually manageable projects, and also to create modular software components from which each service offerer and end user can select, mix, and match for his needs and preferences. Existing, well-known software illustrates the viability of the approach. Object-oriented concepts are compatible with this conceptual framework and can be used to enrich it.

We have made a start towards identifying resource manager and application enabler classes, but must admit doubt that the structures of some components (e.g., *billing subsystem*, *source selector and fuser*) are at all understood. Perhaps some distinctions differentiating components will forever remain fuzzy; nevertheless we feel requirements analysis cannot proceed without an orderly and complete partitioning which identifies what each requirement is for.

Although we have touched on information protection, and other authors have discussed it extensively, the technical requirements are incompletely understood. Deep-seated privacy concerns must be accommodated. These have not yet been thought through sufficiently, either from a technical or a public policy perspective.

A comprehensive ***DL*** requirements analysis is a multi-year effort for many people, but does not require nearly as much resource or time as will be expended if current ill-organized and redundant activities continue. By example from a ***DL*** portion in which an orderly requirements analysis is available[Gl90], we argue the feasibility and value of standard engineering approaches.

Acknowledgements

This report has been drawn from many sources, among which the other participants of the San Antonio workshop are prominent. We are also indebted to a dozen California colleagues who commented on draft manuscripts and identified sources of related technology.

Bibliography

Ab93 M. Abadi, M. Burrows, B. Lampson, and G. Plotkin, *A Calculus for Access Control in Distributed Systems*, ACM Trans. Prog. Lang. and Sys. *15(4)*, 706-734, (1993).

Ag93 R. Agrawal, T. Imielinski, and A. Swami, *Database Mining: A Performance Perspective*, IEEE Trans. on Knowledge and Data Engineering, Special Issue on Learning and Discovery in Knowledge-Based Databases, (Dec. 1993).

An91 *Report of the APS Task Force on Electronic Information Systems*, Bulletin of the American Physical Society, *36(4)*, 1119-1151, (1991).

Ar92 W.Y. Arms et al., *The Mercury Electronic Library and Library Information System II*, Mercury Technical Report Series, Carnegie Mellon University, (Feb. 1992).

Ba92 J. Barker and L. Kingman, periodic reports of the *Case Western Reserve University Library Collections Service Project*, available from CWRU, 10900 Euclid Ave., Cleveland, Ohio 44106, (1989-1993).

Be94 N.J. Belkin, P. Kantor, E.A. Fox, and J.A. Shaw, *Combining the evidence of multiple query representations for information retrieval*, Information Processing and Management, (in press).

Bo93 C.M. Bowman, P.B. Danzig, and M.F. Schwartz, *Research Problems for Scalable Internet Resource Discovery*, Univ. of Colorado Tech. Report *CU-CS-643-93*, (1993).

Br93 J. Browning, *Libraries without Walls for Books without Pages: What is the Role of Libraries in the Information Economy?*, Wired, premiere issue, (1993).

Ch94 J. Chaumier, *La nouvelle offre logicielle pour la recherche documentaire*, Documentaliste – Sciences de l'Information, *31(1)*, 3-8, (1994).

Co86 Anonymous, *Brittle Books: Reports of the Committee on Preservation and Access*, available from the Commission on Preservation and Access, 1400 16th Street, NW, Washington, DC 20036, (1986).

Cu92 J. Culshaw, *American Memory: Taking the Library of Congress to the Masses*, CD-ROM Librarian *7(9)*, 14-21, (Oct. 1992).

Dr91 H. Drenth, A. Morris, and G. Tseng, *Expert systems as information intermediaries*, Annual Review of Information Science and Technology, *26*, 133-154, (1991).

Fo87 E. Fox, *Development of the CODER System: A Testbed for Artificial Intelligence Methods in Information Retrieval*, IP*M 23(4)*, 341-366, (1987).

Fo87b E. Fox and R. France, *Architecture of an Expert System for Composite Document Analysis, Representation and Retrieval*, International Journal of Approximate Reasoning, *1(2)*, 151-175, (1987).

Fo93 E.A. Fox (ed.), *Source Book on Digital Libraries*, TR 93-35, Dept. of Computer Science, Virginia Tech, (1993); available using anonymous FTP to fox.cs.vt.edu and directory /pub/DigitalLibrary, or with gopher to the server on fox.cs.vt.edu, or with WWW to gopher://fox.cs.vt.edu/11/DL.

Fo93b E. Fox, *Digital Libraries*, IEEE Computer *26(11)*, 79-81, (Nov. 1993).

Fo93c E. Fox and L. Lunin, *Introduction and Overview to Perspectives on Digital Libraries*, Journal of the American Society for Information Science (JASIS) *44(8)*, 441-443, (Sept. 1993).

Fo93d E. Fox, D. Hix, L. Nowell, D. Brueni, W. Wake, L. Heath, and D. Rao, *Users, User Interfaces, and Objects: Envision, a Digital Library*, Journal of the American Society for Information Science (JASIS) *44(8)*, 480-491, (Sept. 1993).

Fo93e E. Fox, R. France, E. Sahle, A. Daoud, and B. Cline, *Development of a Modern OPAC: From REVTOLC to MARIAN*, Proc. 16th Annual Int. ACM SIGIR Conf. on R & D in Information Retrieval, SIGIR '93, 248-259, (Pittsburgh, June 1993).

Gl89 H.M. Gladney and P.E. Mantey, *Requirements Analysis for a Document Storage Subsystem*, IBM Research Report *RJ 7085*, (1989).

Gl90 H.M. Gladney and P.E. Mantey, *Integrated Records Management – A Statement of Requirements on the Library Subsystem*, IBM Research Report *RJ 7425*, (April 1990). Submitted for publication in Z. Ahmed and N. Belkin (ed.), *Proc. Workshop on On-line Access to Digital Libraries*, to be published by IEEE Computer Society Press, (Fall 1994).

Gl92 H.M. Gladney, *Access Control for Large Collections*, IBM Research Report *RJ 8946*, (August 1992). Submitted to ACM Trans. Information Systems.

Gl93 H.M. Gladney, *A Storage Subsystem for Image and Records Management*, IBM Systems Journal *32(3)*, 512-540, (1993). Part of what this paper describes has been realized in commercial products.

Go90 C.F. Goldfarb and S.R. Newcomb, *Hypermedia/Time-based Document Structuring Language (HyTime)*, ANSI Project X3.749-D, X3V1.8M/SD-7.

Gr93 J. Gray and A. Reuter, *Transaction Processing: Concepts and Techniques*, Morgan Kaufman Publishers, San Mateo, California, (1993).

Gr94 K. Grœnbaek and R.H. Trigg, *Design Issues for a Dexter-Based Hypermedia System*, Comm. ACM *37(2)*, 41-49, (1994).

Ha92 B.J. Haan, P. Kahn, V.A. Riley, J.H. Coombs, and N.K. Meyrowitz, *IRIS Hypermedia Services*, Comm. ACM *35(1)*, 36-51, (Jan. 1992).

Ha94 H. Hahn and R. Stout, *The Internet Complete Reference*, Osborne Magraw-Hill, Berkeley, California, (1994). Mosaic was written by M. Andreessen of the National Center for SuperComputer Applications (NCSA) at the University of Illinois.

Ha94a F.G. Halasz and M. Schwartz, *The Dexter Hypertext Reference Model*, Comm. ACM *37(2)*, 30-39, (1994). Extended version in *Proceedings of the Hypertext Workshop*, NIST Special Publication 500-178, 95-133, (March 1990).

Is88 International Organization for Standardization, *Open Systems Interconnection, Reference Model, Part 2: Security Architecture*, ISO 7498-2, Geneva, Switzerland, (1988).

Is89 International Telegraph and Telephone Consultative Committee (CCITT), *Open Systems Interconnection – The Directory*, ISO DIS 9594-1 to 9594-8, CCITT *X.500-X.521*, Switzerland, (1989).

Is91 International Standards Organization (ISO), *Information Technology – Text and Office Systems – Document Filing and Retrieval Draft International Standard*, ISO/IEC JTC 1/SC 18 *10166-1*, (June 28, 1991). (This draft standard has been ratified.)

Ja92 P. Janson, R. Molva, and S. Zatti, *Architectural Directions for Opening IBM Networks: the Case of OSI*, IBM Systems Journal *31(2)*, 313-335, (1992).

Je53 Charles Coffin Jewett, *Smithsonian Report on the Construction of Catalogues of Libraries*, (1853).

Ke93 A.R. Kenney and L.K. Personius, *A TestBed for Advancing the Role of Digital Technologies for Library Preservation and Access*, Final report by Cornell University to the Commission on Preservation and Access, Cornell University, (October 1993).

Ko93 P. Kochevar, Z. Ahmed, J. Shade, and C. Sharp, *Bridging the Gap Between Visualization and Data Management: A Simple Visualization Management System*, Proc. IEEE Visualization 1993 Conference, San Jose, CA, (October 1993).

Ku91 R. Kumar, *OSF's Distributed Computing Environment*, IBM AIXpert, 22-29, (Fall 1991).

La92 B. Lampson, M. Abadi, M. Burrows, and E. Wobber, *Authentication in Distributed Systems: Theory and*

Practice, ACM Trans. Computing Sys. *10(4)*, 265-308, (1992). Also in ACM Operating Systems Review *25(5)*, 165-182, (1991).

La89 G.P. Landow, *Hypertext in Literary Education, Criticism, and Scholarship*, Computers and the Humanities *23*, 173-198, (1989).

Ly91 C.A. Lynch, *The Z39.50 Information Retrieval Protocol: An Overview and Status Report*, Computer Communication Review *21(1)*, 58-70, (1991).

Ma87 T.W. Malone, K.R. Grant, F.A. Turbak, S.A. Brobst, and M.D. Cohen, *Intelligent Information Sharing Systems*, Comm. ACM *30(5)*, 390-402, (1987).

Mo91 A. Morris, *Expert systems for library and information services - a review*, Information Processing and Management *27(3)*, 713-724, (1991).

Na93 *Digital Library Initiative, FY 1994*, A joint initiative of the National Science Foundation, the Advanced Research Projects Agency, and the National Aeronautics and Space Administration, U.S. Government document *NSF 93-141*, (1993).

Ni92 G. Nickerson, *WorldWideWeb: Hypertext from CERN*, Computers in Libraries *12(11)*, 75-77, (1992).

Pi94 K. Piersol, *A Close-Up of OpenDoc*, Byte *19(4)*, 183-188, (March 1994).

Po92 J.A. Polly and E. Lyon, *Out of the Archives and into the Streets: American Memory in American Libraries*, Online *16(5)*, 51-57, (Sept. 1992).

Sh94 M.A. Sheldon, A. Duda, R. Weiss, J.W. O'Toole, and D.K. Gifford, *Content Routing for Distributed Information Servers*, Proc. Conf. Extending Database Technology, 109-122, (Cambridge, Mass., March 1994).

Sh90 T. Shelter, *Birth of the BLOB*, Byte *15(2)*, 221-226, (Feb. 1990).

Si91 M.C. Sievert, E.J. McKinin, E.D. Johnson, and J.A. Mitchell, *Retrieval from Full-Text Medical Literature: The Dream and the Reality*, Proc. 15th Symposium on Computer Applications in Medical Care, 348-352, (1991).

St91 M. Stonebraker and J. Dozier, *SEQUOIA 2000: Large Capacity Object Servers to Support Global Change Research*, available at s2k-ftp.CS.Berkeley.edu as file /pub/sequoia/tech-reports/s2k-91-01.ps.

St92 G.A. Story, L. O'Gorman, D. Fox, L.L Schaper, and H.V. Jagadish, *The RightPages Image-Based Electronic Library for Alerting and Browsing*, IEEE Computer *25(9)*, 17-26, (1992).

Th85 G.R. Thoma, S. Suthasinekul, F.L. Walker, J. Cookson, and M. Rashidian, *A Prototype System for the Electronic Storage and Retrieval of Document Images*, ACM Trans. Office Info. Sys. *3(3)*, 279-291, (1985).

We93 K. Webster, *Cornell Project Saves Documents, Books – and Makes Them Accessible*, Adv. Imaging, 42-46, (Sept. 1993).

Vi88 A. Vickery, H.M. Brooks, B.A. Robinson, and J. Stephens, *Expert system for referral*, Library and Information Research Report 66. London, The British Library. (1988).

Wo87 D. Woelk and W. Kim, *Multimedia Information Management in an Object-Oriented Database System*, Proc. 13th VLDB Conference, 319-329, Brighton (1987).

Ya93 K. Yamamoto (conference chair), Proc. 2nd Int. Conf. on Document Analysis and Recognition, Tsukuba Science City, Japan, (Oct. 1993).

Ya88 N. Yankelovich, B.J. Hahn, N.K. Meyrowitz, and S.M. Drucker, *Intermedia: The Concept and the Construction of a Seamless Information Environment*, Computer *21(1)*, 48-59, (1988).

Za92 S. Zatti, *Name Management and Directory Services*, IBM Research Report *RZ 2268*, (1992).