Cross-domain Resource Discovery: Integrated Discovery and use of Textual, Numeric, and Spatial Data: Annual Report: 1 October1999 – 30 September 2000

Ray R. Larson
(University of California, Berkeley)
Paul B. Watry
(University of Liverpool)

1 Introduction

The pursuit of knowledge by scholars, scientists, government agencies, and ordinary citizens requires that the seeker be familiar with the diverse information resources available. They must be able to identify those information resources that relate to the goals of their inquiry, and must have the knowledge and skills required to navigate those resources, once identified, and extract the salient data that are relevant to their inquiry. The widespread distribution of recorded knowledge across the emerging networked landscape is only the beginning of the problem. The reality is that the repositories of recorded knowledge are only a small part of an environment with a bewildering variety of search engines, metadata, and protocols of very different kinds and of varying degrees of completeness and incompatibility. The challenge is to not only to decide how to mix, match, and combine one or more search engines with one or more knowledge repositories for any given inquiry, but also to have detailed understanding of the endless complexities of largely incompatible metadata, transfer protocols, and so on. This report describes our progress in the 1 October 1999 – 30 September 2000 period on NSF/JISC award #IIS-9975164 in building an information access system that provides a new paradigm for information discovery and retrieval by exploiting the fundamental interconnections between diverse information resources including textual and bibliographic information, numerical databases, and geo-spatial information systems. This system is intended to provide an object-oriented architecture and framework for integrating knowledge and software capabilities for enhanced access to diverse distributed information resources.

1.1 Overview

This annual report discusses both the practical application of existing technology to the problems of cross-domain resource discovery (using the Cheshire II system), and also describes the design and basic systems architecture for our next-generation distributed object-oriented and for our next-generation information retrieval system (Cheshire III). For the first purpose we have been refining an making ready for production a next-generation information retrieval system based on international standards (Z39.50 and SGML) which is already being used for cross-domain searching in a number of applications within the Arts and Humanities Data Service (AHDS) (Specifically for the History Data Service hosted at the University of Essex) and the Higher Education Archives Hub (hosted at Machester Computing) in the UK. We are at work to include

additional data sources including the CURL (Consortium of University Research Libraries), the Online Archive of California (OAC) and the Making of America II (MOA2) database as principal repositories. The current Cheshire II system is being set up as a "turn-key" search environment for Digital Libraries based on SGML/XML and will serve as a model for developing efficient paradigms for information retrieval in a cross-domain, distributed environment.

The second purpose is being addressed in the on-going design, development, and evaluation of a new distributed information retrieval system architecture. This architecture includes both client-side systems to aid the user in exploiting distributed resources, object-oriented server-side functionality that supports protocols for efficient and effective retrieval in a internationally distributed multi-database environment. The aim for this design work is to produce a robust, fully operational system ("Cheshire III") within the three year period of the project, which will facilitate searching on the internet across collections of "original materials" (i.e., early printed books, records, archives, medieval and literary manuscripts, and museum objects), statistical databases, full-text, geo-spatial and multimedia data resources, and permit easy sharing of information across distributed, diverse collections.

This system is based on the previous work done with the Cheshire II system in UC Berkeley Digital Library Initiative project. However, as discussed below, the system is being completely redesigned and extended with additional capabilities within a new system architecture. The new extensions to this system will provide a platform and protocols to integrate databases with fundamentally different content and structure into a common retrieval, display, and analysis environment. These different database types, and some examples to be used in this project, include:

• Document databases which describe information about various topics ranging from news reports and library catalogue entries to full-text articles from academic journals including text, images and multimedia elements. (Higher Education Archives Hub, Oxford Text Archive, Performing Arts Data Service, California Sheet Music Project, CURL database, the Digital Archive of California and the Making of America II (MOA) database).

• Numeric statistical databases which assemble facts about a wide variety of social, economic, and natural phenomena (History Data Service, NESSTAR and UC Data).

• Geographic databases derived from geographic information systems, digitized maps, and other resource types which have assembled georeferenced view of the geographic features and boundaries including georeferenced information derived from place names (Archaeology Data Service, History Data Service, the UC Berkeley Digital Library database and the MOA database).

The work described in this report is also distributed, with the work on the new server architecture being developed at the University of California, Berkeley and the client-side implementations being developed at the University of Liverpool. The remainder of this report is organized as follows: Section 2 describes the Cheshire II system implementation and deployment; Section 3 describes the design and development of the Cheshire III system; Section 4 discusses research issues and plans in cross-domain resource discovery and the current design for implementing effective distributed searching using the Cheshire II system; Section 5 reports on the status of testbed

implementation; and Section 6 list new publications and public presentations associated with this project.

2 System Description

As noted above, there are two server-side systems being produced by this project:

1. The Cheshire II system that is built upon international standards and existing work in probabilistic information retrieval, and on the experience of the researchers in applying advanced retrieval methods to full-scale realistic databases. This system is being "hardened" and additional user tools are being developed so that this system can be easily deployed for providing access to SGML/XML collections.
2. The Cheshire III system that is a complete redesign, and indeed is an experimental system incorporating cutting-edge technologies.

The remainder of this section describes our progress in developing and implementing databases using the Cheshire II system. In it we also describe our progress on the client-side implementation. The following section (section 3) describes the current design and implementation status for the next-generation Cheshire III system.

2.1 Cheshire II development

The continuing development of the Cheshire II client/server system is based on a particular vision of how information access tools will develop, in particular, how they must respond to the requirements of a large population of users scattered around the globe who wish simultaneously to access the complete contents of thousands of archives, museums, and libraries, containing a mixture of text, images, digital maps, and sound recordings. Such a virtual library must be a network-based distributed system with local servers responsible for maintaining individual collections of digital documents, which will conform to a specific set of standards for documentation description, representation, and communications protocols. We believe, based on the current directions of research and adoption of standards by libraries, museums and other institutions, that a major portion of this emerging global virtual library will be based on SGML (Standard GeneralizedMarkup Language), and especially its XML subset, and the Z39.50 information retrieval protocol for resource discovery and cross-database searching. (We also assume that the forthcoming versions of the HTTP protocol will continue to provide document delivery and hypertext linking services, and that SQL3, when finalized, will provide the low-level retrieval and data manipulation semantics for relational and object-relational databases). The Cheshire II retrieval system, in supporting Z39.50 "Explain" semantics for navigating digital collections, allows users to locate and retrieve information about collections that are organized hierarchically and distributed across servers. It will enable coherent expressions of relationships among objects and collections, showing for any given collection superior, subordinate, related, and context collections. These are essential prerequisites for the development of cross-domain resources discovery tools, which will enable users to access diverse collections through a single interface. It specifically addresses the critical issue of "vocabulary control" by supporting probabilistic "best match" ranked searching (as discussed below) and support for "Entry Vocabulary Modules" (EVMs) that provide a mapping between a searcher's natural language and controlled vocabularies used in the description of digital objects and collections. It also allows users to "navigate" collections (the "drilling down approach") through distributed Z39.50 "explain" databases and through the use of SGML

as the primary database format, particularly for collection-level descriptions such as the EAD DTD. The system will follow the recommendations of the Third National Resource Discovery Workshop by providing fully distributed access to existing catalogues, and is designed to support cross-domain “clumps” to facilitate resource discovery. Finally, the proposed server anticipates the critical issue of displaying non-western character sets in its ability to handle UNICODE (in addition to the standard ASCII/ISO8859 character sets).

2.1.1 Cheshire II Development History

The development of the Cheshire system began in the early 1990s at the University of California, Berkeley, as a means of testing the use of “probabilistic information retrieval methods” upon MARC bibliographic data. It was found that these advanced retrieval methods developed at Berkeley were far more effective than traditional Boolean methods (or vector space model methods) in accessing records from a bibliographic database. Needless to say, the deployment of these “probabilistic” retrieval algorithms has very important economies particularly in the searching of databases or documents such as EAD which normally do not use a controlled vocabulary.

The second version of Cheshire, currently deployed at both the University of Liverpool and the University of California, Berkeley, was designed to extend the format of the server to include SGML-encoded data. Because SGML is increasingly becoming the markup language of choice for research institutions, it was critical to extend Cheshire’s capabilities to support the kinds of SGML metadata which is likely to be included in national bibliographies. These are: TEI (Text Encoding Initiative), EAD (Encoded Archival Description), DDI (for Social Science Data Services), CIMI (Consortium for the Interchange of Museum Information) records, as well as the SGML version of USMARC released by the Library of Congress (based on the USMARC DTD developed by Jerome McDonough for the Cheshire project).

The third version extends the use of SGML handling capabilities for these search indexes. This version was developed by Berkeley and Liverpool for the Arts and Humanities Data Service, enabling GRS-1 syntaxconversion for nested SGML data, component indexing and retrieval of SGML formatted documents, and automatic generation of Z39.50 Explain databases from system configuration files. The current version of the server is now able to include an element in an SGML record that is a reference to an external digital object (such as a file name, URL or URN) that contains full-text to be parsed and indexed, these can be local files or URL and URN referenced files anywhere on the internet. It also enhances the users’ ability to perform somewhat less directed searching provided by Boolean and probabilistic search capabilities that can be combined at the user’s direction. This version of Cheshire can display a number of data types ranging from full-text documents, structured bibliographic records, as well as complex hypertext and multimedia documents. At its current stage of development, Cheshire forms a bridge between the realms of purely bibliographic information and the rapidly expanding full-text and multimedia collections available online.

2.1.2 Features of Cheshire II

The Cheshire II system includes the following features:

1. It supports SGML and XML as the primary database format of the underlying search engine. The system also provides support for full-text data linked to

SGML or XML metadata records. MARC format records for traditional online catalog databases are supported using MARC to SGML conversion software developed for the project.

2. It is a client/server application where the interfaces (clients) communicate with the search engine (server) using the Z39.50 v.3 Information Retrieval Protocol. The system also provides a general Z39.50 Gateway with support for mapping Z39.50 queries to local Cheshire databases and to relational databases
3. It includes a programmable graphical direct manipulation interface under X on Unix and Windows NT. There is also CGI interpreter version that combines client and server capabilities. These interfaces permit searches of the Cheshire II search engine as well as any other z39.50 compatible search engine on the network.
4. It permits users to enter natural language queries and these may be combined with Boolean logic for users who wish to use it.
5. It uses probabilistic ranking methods based on the Logistic Regression research carried out at Berkeley to match the user's initial query with documents in the database. In some databases it can provide two-stage searching where a set of “classification clusters”(Larson 1991) is first retrieved in decreasing order of probable relevance to the user's search statement. These clusters can then be used to provide feedback about the primary topical areas of the query, and retrieve documents within the topical area of the selected clusters. This aids the user in subject focusing and topic/treatment discrimination. Similar facilities are used in the Unfamiliar Metadata Vocabularies project at Berkeley for mapping users’ natural language expressions of topics to appropriate controlled vocabularies (http://sims.berkeley.edu/research/metadata).
6. It supports open-ended, exploratory browsing through following dynamically established linkages between records in the database, in order to retrieve materials related to those already found. These can be dynamically generated “hypersearches” that let users issue a Boolean query with a mouse click to find all items that share some field with a displayed record.
7. It uses the user's selection of relevant citations to refine the initial search statement and automatically construct new search statements for relevance feedback searching.
8. All of the client and server facilities can be adapted to specific applications using the Tcl scripting language. I
9. mage Content retrieval using BlobWorld
10. Support for the SDLIP (Simple Digital Library Interoperability Protocol) for search and as Z39.50 Gateway

2.1.3 Current Usage of Cheshire II

The Cheshire II system currently has a wide variety of ongoing implementations using WWW and Z3.50 implementations. Current usage of the Cheshire II system includes :

- Berkeley NSF/NASA/ARPA Digital Library
  - Includes support for full-text and page-level search.
  - Experimental Blob-World image search
- World Conservation Digital Library
- SunSite (UC Berkeley Science Libraries)

- University of Essex, HDS (part of AHDS)
- Oxford Text Archive (test only)
- California Sheet Music Project
- Cha-Cha (Berkeley Intranet Search Engine)
- Berkeley Metadata project cross-language demo
- Univ. of Virginia (test implementations)
- JISC data sets at MIMAS
- University of Liverpool Special Collections and Archives
- University of Warwick, Modern Records Centre
- Bodleian Library, Oxford
- The HE Archives Hub (Currently numbers 20 repositories, but to be extneded to include approximately 70 HE/FE repositories throughout the United Kingdom)
- DeMontfort University (MASTER project)
- University of London Library
- Online Archive of California
- CIAO, University of California
- University of Liverpool Museum and Art Gallery

3 Background and Design

The first year of this project has been largely concerned with the design and initial development of the next-generation Distributed Object Retrieval Architecture. This is the basis for our planned distributed system for cross-domain retrieval. In the initial proposal we expected to be using CORBA for distributed objects in the new system, but recent developments in Java have led us to choose instead the 'JavaSpaces' framework based on the LINDA system from Yale University. JavaSpaces will provide the ability to distribute the system and data in a much more effective way than is possible with CORBA. As noted in the original proposal, established standards have been followed in the on-going development of the Cheshire II system. While we have been designing and beginning development on Cheshire III we have continued to update the Cheshire II system and make it available for use as discussed in the preceding sections.

We see the architecture for the evolution of distributed information access systems as a highly extensible and dynamic system. In such a system both the data (digital objects instantiating information resources) and the programs that operate on that data (methods) to achieve the needs and desires of the users of the system for display and manipulation of the data (behaviours) will be implemented in a distributed object environment. The basic architecture is a three-tiered division of data and functionality. The tiers are:

1. The Client. The basic client for the distributed Cheshire system can be any JAVA-enabled WWW Browser. The primary data delivery format will be as XML (for initial versions), and the methods for manipulating and navigating within the data will be implemented as JAVA applets, delivered on demand to the browser.

2. The Application Tier Applications for search and manipulation of data are distributed between the client and network servers (including the repositories) to provide distributed functionality (and to provide new behaviours to clients on

demand from any compliant network server). The application tier or layer would both provide JAVA applets for execution on the client, as well as providing server-side methods invoked directly on objects in the repository either via direct invocations or indirectly via requests from other protocols (e.g. Z39.50 or Open Geo-spatial Datastore Interface (OGDI) for network access to heterogeneous geographic data held in multiple GIS formats and spatial reference systems). For example, a client browser might download an applet that can display MARC records, and invoke a server-side method to convert repository objects in XML to MARC format. We expect, for performance reasons, that many operations on stored objects will be server-side methods with primarily display functions on the client side.

3. The Repository Digital objects and metadata describing them will reside in the repositories tier or layer.

Repositories can be implemented in a variety of ways, ranging from conventional Relational, Object-Relational, or Object-Oriented database systems and Text retrieval engines, to metadata repositories referencing physical collections in libraries.

The following is derived from documents available on our WWW site as the basic design documents for the Cheshire III system.

3.1 Cheshire III Design: Introduction

As indicated in the preceding sections on Cheshire II, the Cheshire system is a client/server information retrieval system that brings modern information retrieval techniques to a wide array of data domains. Cheshire provides uniform document storage in the form of SGML/XML, supports probabilistic search, and supports Z39.50 interoperability with dozens of library information systems around the world.

Cheshire II is now several years old. Its program logic is coded entirely in C and most of its user interface is done in Tcl/Tk. Further development is being inhibited by a system complexity that has outgrown its original design and its dated software technologies. New technologies are now available that can dramatically reduce coding effort and enhance robustness, maintainability, and interoperability. This section of the annual report describes how we are trying to re-engineer Cheshire into a modern software system, with the hope of ensuring its future viability as a platform for information retrieval research.

The next section outlines the system objectives of a next generation system. Section 3.3 discusses the technologies available for meeting those objectives. Section 3.4 examines issues in migrating the existing system to the new design. Section 3.5 concludes the discussion of the current design.

3.2 Cheshire III: Design Objectives

To continue Cheshire's viability as a research and production platform, the system must appeal to users and developers alike. It must satisfy the information needs of users, and it must also make it easy for developers to modify and experiment with the system. We identify below seven sets of features that we see as desireable in a next- generation Cheshire.

- Distributed Queries. We want Cheshire to be able to satisfy a user's cross-domain information need. To do that, a Cheshire server needs to look at not only the database it maintains, but also other information sources reachable over the Internet. Each server needs access to meta information about other information sources, from which it can decide whether to query that particular source to satisfy a particular information need.
- Interoperability. To maximize its domain reach, Cheshire systems need to not only interoperate with each other, but with as many different types of systems as possible. Cheshire needs to support international standards such as Z39.50, emerging protocols such as SDLIP, and be ready to adopt future interfaces.
- Concurrency, Scalability and Robustness. A research system is most useful if the results of good research can be immediately deployed to serve a large number of users. Cheshire should be able to efficiently use machine resources to serve concurrent requests, grow linearly in throughput as more hardware resources are added, and offer reliable operations suitable for academic environments.
- Web-Based System Adminitration. The system should be easy to deploy and easy to administer. The administrator should be presented with a unified view of system operations and given easy means to customize them. A web-based administration tool will give administrators the most flexibility in accessing the system.
- Dynamic Databases. An administrator should be able to incrementally grow the database without interrupting user services. The database should be automatically indexed as it is grown.
- Simplified, Structured, Maintainable Code. A research platform exists to serve innovation. It is constantly evolving as new ideas are tested and old ideas discarded. Developers come and go. This process becomes more vibrant if the system is accessible to new developers and is amenable to change.
- Unprivileged Deployment. A research system is a toy for everyone. Its users will not always have privileged access to the operating system. Cheshire should not require such access for deployment.
- A focus on high performance, network of workstations style operations. A scalable, extensible platform for information retrieval research. Scalable performance allows us to explore more resource intensive IR techniques.

3.3 Cheshire III: New Technologies

A number of new software technologies exist today that can help achieve the objectives outlined in the previous section.

- Java Programming Language. Java has become the development language of choice for most new internet development. It includes a variety of features that make it desireable for development of the server. One of that major promises of Java is that once a developer writes Java code it should be instantly portable to almost every platform. Java is strongly typed and object oriented from the ground up, giving us

programs that are more robust, more maintainable, and more expandable. Networking is at the core of Java. Everything from naming, to its execution environment, to API design is tailored to networked computing. A rich set of reusable Java components is available to provide infrastructure support, and Java integration is available for nearly every other major language (to permit inclusion of “legacy” components). Java performance appears to be adequate to provide the systems glue, with performance critical components written in other languages. We also expect that fully optimized native compilation for Java will become available soon. Java 1.3 promises to provide significant improvements in client-side performance. Java tools are mostly free. Java programmers are easy to find. And as one of the most successful computing platforms of our era, users can expect continued expansions, upgrades, and community support.

- Java RMI Remote Method Invocation makes it extremely simple to implement client/server communications. A method on a remote object is called exactly the same as a local method. The network is nearly invisible.
- Java HotSpot Server (V2.0 available for Solaris and Linux in fall, 2000). High performance multithreaded Virtual Machine for server applications. This can be used in Cheshire for high performance concurrency.
- Berkeley DB 3.1.x Better concurrency support. Java API included.
- JavaServlet Pages. A Java architecture for generating dynamic content, to be delivered through standard web servers. We can use this to build a web interface for Cheshire. More and more thin clients such as PDA's will have web browsing capability. Cheshire may see radically different kinds of use in the future. For example, a book store patron may want to check on his PalmPilot to find out if an expensive book is available at a public library and if yes, immediately make a time limited reservation.
- DOM Compliant XML Parsers. Available with Java API's. Conform to the DOM standard for efficient (fewer passes) parsing of XML.
- Forte Integrated Development Environment. A Java IDE that includes an integrated debugger, GUI access to object tree, etc.
- Java Naming and Directory Interface Comes with a reference directory service. Cheshire servers connected to the Internet can discover each other through this service, allowing them to cooperate both in the local area and in the wide area.
- SDLIP A simple information retrieval protocol available with Java transport and CORBA and HTTP bindings. SDLIP is simple to understand and elegant in design. SDLIP is far simpler to implement than Z39.50 and may find more support from a wider array of information sources, particularly from the fast evolving web search engines. The simplicity of SDLIP can encourage experimentation in user interface, server architecture, and retrieval algorithms. SDLIP may reduce the barrier to entry for distributed library information systems the way the web reduced barriers to publishing. The web paradigm has been a move away from structure, away from careful selection and long range planning, and a move toward universal, low barrier access while powerful computers mine structure from information where its human makers were spared from the effort.

- JavaSpaces - high level coordination mechanism for distributed systems. Provides a light-weight publish/subscribe distributed programming model. 'Messages' in a java space should be small. Use direct, point-to-point mechanisms for bulk data transfers. Since we assume that Cheshire will have low volume transactions. In JavaSpaces distributed transactions, leasing, events come for 'free'. We intend to use JavaSpaces to govern distributed transactions for Search, Display and Update of the database(s). Using JavaSpaces we are adopting a single operational model for Cheshire that encompasses single node installations, uniformly administered clusters, as well as independently administered federations of servers. Some of the characteristics are:
    i. every operation is a distributed operation
    ii. an operation is applied over a set of collections
    iii. collections may be:
        1. **Single node** or cluster: can be partitions of other collections
        2. **Federation:** can be partitions or subsets of other collections. In other words, collections in a loosely coupled federation may have overlapping records.
        3. **Virtual Collection:** the external interface (or view) to collections. A VC may present only part of the underlying real collection in its interface. A VC may grow or shrink dynamically within the bounds of the real collection. A search only needs to be done over documents in VC, not all documents in the collection.

    This gives us a way to logically partition a collection across a number of machines for performance increase, but with built in redundancy in the case of node failures. When a node fails, its VC is simply distributed (logically) to other nodes in the cluster. Using this design Cheshire servers can be organized into server groups. A server group can be thought of as an administrative unit.

3.3.1 Designing for parallelism and scalability:

- One I/O worker per disk (conceptually). The philosophy here is that we don't want the OS to attempt more parallelism in software than actually exists in hardware. Because our database software is intelligent about concurrent accesses to disk, we already have this effect and don't need to do anything special here.
- One data worker for each data resource. Here, a data resource should not straddle multiple disks. The philosophy is that there is an optimal way to concurrently access each resource. And the data worker is responsible for scheduling this access.
- One compute worker for each CPU (conceptually). The OS can be thought to be exactly this kind of worker. The OS knows about its CPU resources and schedules them accordingly.
- One task worker for each task. A task may be protocol translation, search, display, update, etc. A distinction should be made between task and data workers in the same address space and those in difference address spaces. In general, data intensive tranfers should happen only

between data and task workers in the same address space. Task workers have direct read access to data. Writes are handled by background data workers.

All of the technologies listed above are freely available. Source code is available for most of them.

3.3.2 Client Technologies

The project is a client/server system with the development of the client largely being written at the University of Liverpool and the server at the University of California, Berkeley. After some investigation, we decided not to adopt some of the original technologies incorporated in the original proposal, as follows:

Changes in Client Design:

Although the project proposal implied a Java based system for the client, we have subsequently decided to utilize the Mozilla framework being developed in an Open Source fashion by the Netscape Corporation. This has greatly increased the existing resources available for use in the creation of the client.

The client must be able to be used under any operating system, be that Windows, Linux, MacOS, as well as on as many hardware platforms as possible as well. Mozilla has been designed to fit this from the ground up, with the ability to be cross-platform of primary concern in the implementation strategy.

The client interface must be familiar to as many users as possible. As can be seen in the information technology marketplace every day, most new products have the same style of interface as those that have gone before in order to make the learning curve for consumers as brief as possible. By simply extending Mozilla to handle the Z39.50 protocol, this learning curve will be minimalised as most of the functionality of the client is already present in the original, and thus already familiar.

Mozilla is made up of small discrete components that fit together into a larger whole. As such, adding additional functions to it is just a matter of writing a component that works together with the other components already written. This reduces the development time as well as ensuring that when Mozilla is updated, the changes needed to keep the Z39.50 component synchronised will be minimised while still having access to all future advances in the Mozilla framework.

Mozilla implements XPI - the Cross Platform Installer. This is a means of having new components or User Interface modifications installed automatically by clicking a button on a web page or similar method. As in any system accessable via the Internet, clients will be run over network connections of different speeds. The startup time for a fully implemented java client run inside a conventional web browser over a typical modem connection from a home PC would be unbearably slow. By only requiring a once off install of the Z39.50 component and User Interface, this is minimised.

Finally Mozilla supports all of the current relevant standards. It supports the Document Object Model level 1 (DOM1), HTML version 4.0, ECMAscript (standardised javascript), Resource Discovery Framework (RDF), eXtensable Markup Language (XML),

Cascading Style Sheets (CSS) and so forth. Adherence to standards is essential to ensure that the product remains usable with different servers and data types.

To integrate Z39.50 into the existing code base of Mozilla will require careful planning which will require expert familiarity with the structure of the current code base. To ensure that future developments in the Mozilla codebase do not outdate the Z39.50 code, the standards used by Netscape must be adhered to strictly, and the new code added in the best place for it, not simply "bolted on". Since the code for Mozill ais currently more than 20Mb, this is not expected to be a trivial task. The first part of the project will therefore be a planning and familiarization phase to ensure optimal execution of the code within the Mozilla framework.

The construction of the client base will be done using the following phases:

- Z39.50 URL Phase. The client must be able to locate a server via a z39.50 URL. This will be used to retrieve a single document as well as search the server and display the results. The URL scheme must therefore be integrated into Mozilla's existing network code structure, reusing the existing functions and objects. (The URL scheme to be used is therefore not that proposed in RFC2056, as this does not incorporate user authentication, searching, or other Z39.50 commands.) The retrieved data may then be handled by the HTML, text, or other mime type handlers already in Mozilla. This work package will consist of preparing the draft specifications for the URL scheme, and start work on the development of same.
- Z39.50 Scripting Phase. The clieent must also be scriptable via Javascript and XUL. This requires the Z39.50 functions to be accessible via XPConnect (Mozilla's library to link Javascript and the underlying C++ code). As such it must be compatible with XPCOM (Mozilla's cross-platform component object model). A Z39.50 session object must be made available to the javascript components that can call CONNECT, DISCONNECT, SEARCH, SCAN, and so forth, to enable the client to intelligently process information available via the Z39.50 information retrieval protocol.
- Interface Building Phase. This phase will build upon the Z39.50 scripting phase, outlined above. Using these scripting capabilities and XUL, we will build and test a new interface that allows for improved accessibility to the target servers. The interface will therefore be easy to use, but have enhanced functionality. This interface will be accessible via Mozilla's automatic install procedures, so that the client may be seamlessly layered on top of Netscape 6.

Although the client and server are designed to work together in a seamless fashion, they must also interoperate with any other Z39.50 compliant system providing such systems support SCAN, EXPLAIN, SEARCH, etc. to a minimum version 3 standard.

3.3.3 Unused Technologies (and reasons)

After some investigation, we decided not to adopt the following technologies.

- Java Enterprise Edition Emphasizes support for enterprise computing needs such as reliable operations and security. Includes Enterprise Java Beans component infrastructure. EJB provides component support services such as life cycle management, persistance, and load balancing. Most of its power is not needed for Cheshire and EJB adds unnecessary complexity to the development

process, which may in the end inhibit experimentation in a research system rather than promote it.

- CORBA. CORBA is a distributed objects architecture used for building distributed systems. In the foreseeable future, SDLIP, Z39.50 and JavaSpaces already provide all the server-to-server interoperability Cheshire will need. In fact, SDLIP itself in a distributed information systems architecture that (in one form) sits on top of CORBA. For our purposes, SDLIP interoperability subsumes CORBA interoperability. At the present, there isn't a clear need to write distributed objects that sit below SDLIP. Even if that need arises in the future, it may be simpler to convert the object interfaces to JavaSpaces implementations.

3.4 Cheshire III: Migration Path

Having outlined a redesign of the Cheshire system using completely new technologies, some important questions now need to be answered. What is the right implementation strategy? Do we integrate the new technologies and new source code with the existing system one piece at a time, or do we build the system completely from scratch all at once?

The integration approach would be done from the top down. The top down approach says first we implement the glue that holds the system together by building interfaces to legacy components, then we replace those components one at a time. The opposite, bottoms up approach makes little sense for our situation; we would see none of the benefits of the new technology at the frontend until the entire system is complete.

The integration approach has the benefit of quickly realizing the benefits of modern server technologies in a deployable system. It also forces us to consider the full set of supporting interfaces as each new interface is implemented, reducing the chance that important functionalities become incompatible with the new interface.

The integration approach has two disadvantages. The first is that integrating legacy components at each step of the way forces us to deal with the old architecture, and will not fully benefit from the cleanness and consistency of a new design. The second is the enormous amount of programming effort that will be required to devise, implement and test these interfaces. Our opinion was that these two disadvantages far outweight the benefits of the integration approach. The existing Cheshire system is usable and there isn't an immediate need for a new server. An interface that doesn't work can always be redesigned. And such a redesigned interface will still be more self-consistent and would have required less implementation effort than an interface built on integration with legacy components at every step of the way.

Rebuilding the system from scratch does seem to be the more sensible approach. Due to the complexity of the Cheshire system, however, it will be difficult to sustain a happy development effort if we were to go on to build a massive system and see nothing working whatsoever for months. We have therefore decided that instead of trying to build all current Cheshire II functionality into Cheshire III from the beginning, to instead adopt an incremental approach where a minimally function system will be constructed and evolve into the full system as development continues.

3.5 Cheshire III Design: Conclusion

The above discussion is our proposed re-design of the Cheshire information retrieval system. While the design does not in itself achieve all of the systems objectives outlined in section 3.2, it does achieve the architectural objective of accommodating all seven of those system objectives. We enable distributed queries through a clean inter-server discovery and search interface. We style toward multiple interfaces and extensibility from the grounded up, with clean plug-ins for interoperability. We take advantage of Java and Java tools to write maintainable, concurrent, high performance code. Java web technologies enable a broad base of clients and allow Cheshire to be searched and administered from anywhere, without having to install software locally. The groundwork has been laid for a next generation platform for information retrieval.

4 Research Issues: Integrating Access to Resources Across Domains

The development and design of the Cheshire system outlined above has been driven in the belief that many of the recommendations of the National Resource Discovery Workshops (MODELS 3 and 4) can be catered for by a standards-based information retrieval system that will provide a bridge between existing online catalogue technology and databases and the explosively growing realm of network-based digital libraries with information resources including full-text, geo-spatial data, numerical data and multimedia. This section addresses the following research and development issues related to those models and our current work on implementation for them:

1) Management of Vocabulary Control in a Cross-Domain Context;
2) Distributed Access to Existing Metadata Resources;
3) Navigating Collections; and
4) Support for Cross-Domain Clumps to Facilitate Resource Discovery.

4.2 Management of Vocabulary Control in a Cross-Domain Context

This is a key issue in the integration of resources across domains, as brought forward in MODELS 4. The underlying problem, recognized for over a decade, is that the current generation of online catalogues in most libraries do not do a very good job of providing topical or subject access to the collections (Matthews, et. al., 1983), The common result of many subject searches (up to 50%) is search failure, or “zero results”. In a distributed environment, this is compounded by the lack of “vocabulary control” across domains, added to which is the tendency for users to use general wording or terms in subject queries, rather than specific ones. In its initial form, the Cheshire system was initially designed to overcome these difficulties, and provide users with the tools for formulate effective queries. In its current configuration, the server is able to map the searcher’s notion of a topic to the terms or subject headings actually used to describe that topic in the database. This is provided for by a variety of search and browsing capabilities,2 but the primary distinguishing feature is the support for probabilistic searching” on any indexed element of the database.

This enables the use of a natural language queries to retrieve the most relevant entries in one or more databases, even though there may be no exact Boolean matches. The

results set of a probabilistic search is ranked in order of estimated relevance to the user's query. The search engine also supports relevance feedback, as well as automatically generated hypertext links that will allow the user to follow dynamically established linkages between associated records. Support for probabilistic retrieval is critical to the success of a cross-domain server, insofar as it allows users to make effective queries even when there is no controlled vocabulary. In the case of the Social Science documents, for example, a user can make a successful "probabilistic search" on a given subject, where a traditional Boolean search would fail. The deployment of these algorithms is only a preliminary steps in managing vocabulary control. There are, naturally, many research issues to be addressed in finding the optimal mappings of user and document vocabulary to the controlled vocabularies used in descriptive metadata this is discussed further under "Search Support for Unfamiliar Metadata Vocabularies" below.

4.3 Distributed Access to Existing Metadata Resources

4.3.1 Data Mining: Dublin Core Metadata

One approach to semantic interoperability of distributed systems is to use a standardized set of metadata, such as the Dublin Core, for the description and retrieval of electronic resources from disparate data. For example, the Arts and Humanities Data Service (AHDS) currently operates on a model in which extended Dublin Core elements are used as the means of retrieving information from five different service providers. In practice, this has proved to be an inefficient way of effectively leveraging data from these services, since the data providers often interpret Dublin Core elements differently and also because Dublin Core elements only comprise a small part of the complex, rich data resources which could be available as a means of search and retrieval. To take one example, a full text search of the existing AHDS gateway (http://prospero.ahds.ac.uk:8080/ahds live) for the term 'England' in the Dublin Core element set for the Oxford Text Archive produces only one 'hit'; whereas a Cheshire version of the TEI-header information of the same service (http://sherlock.berkeley.edu/OTA/) produces 258 'hits', ranked in order of relevance.

This is why many of the fundamental retrieval algorithms being developed as part of the Cheshire project, described below, are based on the premise that front-end prototyping will involve entire information resources, not simply restricted subsets based on Dublin Core metadata. These will provide a much richer platform for the development of retrieval strategies among large and complex data sets, and the inclusion of the Arts and Humanities Data Service (AHDS) in this project will bring particular expertise in this area, since this service currently is a practical implementation of the Dublin Core and the service providers are already experienced in a production environment in the use of Dublin Core for resource discovery (Miller & Greenstein, 1997).

4.3.2 Search Support for Unfamiliar Metadata Vocabularies

The next step beyond simple shared category lists like Dublin Core will be to provide support for enhanced retrieval of unfamiliar metadata (as distinct from Dublin Core Metadata), extending the findings of the DARPA-funded research project on Search Support for Unfamiliar Metadata Vocabularies (http://sims.berkeley.edu/research/metadata). This research, based on work from the Cheshire research projects, is attempting to go substantially beyond the state-of-the-art in developing systems that can construct linkages between natural language

expressions of topical information and controlled vocabularies automatically. Today most such systems depend on the expensive human crafting of links within and between vocabularies.

For the purposes of this project we propose continued development of the Cheshire client and application layer middleware to provide sets of “entry vocabulary modules” based on the controlled vocabularies of our testbed databases. These “EVMs”, will accepts natural language expressions of user’s queries and will generate a ranked lists of controlled vocabulary headings most likely to be useful for that search. This will have three uses:

1. as a prompt when searching an unfamiliar vocabulary

2. as computer-aided or automatic indexing of data resources using existing controlled vocabularies

3. to extend searches, using derived information of found records as a basis for finding similar records in another database

When used in conjunction with the existing Cheshire algorithms for probabilistic indexing and retrieval, these EVMs provide descriptive surrogates can be used to match user or document terminology to corresponding controlled vocabulary terms.

4.4 Navigating Collections (the “Drilling Down Approach”)

One of the primary considerations brought forward in the discussion of search models during the MODELS workshop 4 is the use of Z39.50 to support a “drilling down” approach, which would permit users to “drill down” between generic and domain-specific descriptive information. The difficulties of this in the context of Z39.50 are cited in the MODELS 4 recommendation for further work [item 2.3].

In designing the second and subsequent versions of the Cheshire system, we faced the question of how to provide a search engine that could support a navigational record schema, that could be used on both simple text and complex structured records, and also support complex multimedia documents and databases. In answer to this, it was decided to adopt SGML as the fundamental data storage type for the Z39.50 client/server. Virtually all data manipulation for the database has been generalized as processes acting on SGML tags or sets of tags. Instead of having to develop new routines to manipulate each sub-element of a new datatype, the developer only needs to provide a DTD and a conversion routine to convert the new data type to SGML. The built-in file manipulation and indexing routines can then extract and index any tagged sub-elements of the data type for access.

In using SGML tagging for all data in the database and by adopting the SGML DTD language to define the structure of each data file, it is possible to use a common format for data types ranging from full-text documents, structured bibliographic records, to complex hypertext and multimedia documents (using the HTML DTD that defines the elements of the WWW “pages”). This has important economies in the delivery of resources across domains.

We propose to support a far broader range of SGML document types, and to provide JAVA methods for display of SGML documents on the client. The obvious candidate for this is DSSSL (Document Style Semantics and Specification Language, international standard: ISO/IEC 10179), although the use of XML (Extensible Markup Language) with XSL (Extensible Style Sheets), a restricted subset of SGML with additional formatting capabilities will also be supported. At a practical level, by creating style sheets for the most commonly used SGML data types (EAD, CIMI, DDI, TEI), it will be possible to deliver visual representations of nested data using multiple DTDs. In order to achieve this, the participants will have to agree on some common visual representation of data, requiring consultation among institutions. The University of Liverpool has already begun by developing and distributing a prototype conversion programme which will format and index archival finding aids encoded in EAD. The functionality of this programme will become part of the client-side methods for the Cheshire client (http://gondolin.hist.liv.ac.uk/~azaroth/ead2html.html).

4.5 Support of Cross-Domain Resource Discovery

This project places as a priority the development of a system that will enable true and effective cross-domain resource discovery. This need is evident in the United Kingdom to satisfy the requirements of the DNER(the Distributed National Electronic Resource) and in the United States the NSF/NASA/ARPA Digital Library Initiative projects. In doing so we suggest a number of novel solutions which address ongoing questions about access to distributed materials available via the internet. These may be of use in the strategic planning of national and international services and initiatives.

One of the primary features to enable this will be native support for SGML/XML as a primary data format. Since SGML/XML is now widely used for encoding the kinds of metadata that are most likely to comprise the DNER, the project's support for it will be fundamental to its success. We expect to deliver entire SGML/XML resources, whether they be bibliographic, full-text, or multimedia, while at the same time supporting the Z39.50 information retrieval protocol. This will enable a much broader range of integrated and complete information resources to be delivered to the user's desktop. The system will also simultaneously support multiple Z39.50 profiles (including the Bath and DC Profiles) for standardized metadata when required.

In order to get the end user quickly to the information desired (particularly with large-scale services), the system is addressing a number of research issues associated with data mining. These include: relevance feedback, probabilistic searching algorithms, use of RDF to select relevant services, and the like, which will be enabled by the new design of the project's client.

One of the most important research advances of the project is the development of an architecture enabling efficient searching across hundreds or thousands of distributed network nodes using the Z39.50 SCAN service. This capability will be required for the DNER and similar digital library projects to function effectively.

Our initial work in Cross-Domain Resource Discovery has concentrated on using the facilities of the Z39.50 protocol to implement what we are calling a “Meta-Search” capability using existing Z39.50 servers and resources. We are taking a new approach to building this meta-search based on Z39.50. Many existing attempts at distributed search and resource discovery (including the current AHDS implementation) have relied

on broadcasting of search requests to all servers making up the distributed resources to be searched. There are a number of practical problems with this approach. One of the chief drawbacks is that all systems must be searched, before the user or the search controller can determine which systems are most likely to provide the results that the user is seeking.

Instead of using broadcast search we are using SCAN service of Z39.50 servers to build combined indexes containing information "harvested" from the individual servers. The SCAN service permits Z39.50 requests directly to server indexes, and returns results containing index information including the words or keys in the index along with their frequency of occurrence information for the database. With this information indexes combining information from many servers and databases can be combined and statistical ranking methods can be used to rank those servers and databases according to the probability that they contain relevant information for a given user query.

The Z39.50 SCAN service is included in all Cheshire II servers, permitting us to test this method. We are currently implementing a special indexing mode for the Cheshire II system , which will take a list of servers and use the Explain and SCAN services to build combined "Meta-Indexes" for those servers. Once this capability comes online it will be easy for any Cheshire II server to function as a Meta-Search server for some group of other servers. We plan to make this facility one that can be recursively executed, so that hierarchies of Meta-Search servers can be constructed (opening the possibility for layers of topically-oriented Meta-Search servers, as well as global servers that summarize index information from each of the lower layers in the search hierarchy).

We will also Investigate:

- How to choose databases using the index
- How to merge search results from multiple sources
- Hierarchies of servers (general/meta-topical/individual)

The basic method is then:

- For all servers in a list (which may be a topical subse)
    - Get Explain information. The Z39.50 Explain facility will provide a list of all databases and their indexes (or supported searchable attributes). We will focus initially on the Dublin Core (DC) attributes accessible from each server, since these provide a minimal shared set of metadata that is fairly standard across different databases. We will also attempt to exploit geographic index information indicated by appropriate attributes from the BIB-1 or GEO attribute sets.
    - For each index (or each DC or other supported attribute)
        - Use SCAN to extract index terms and frequency information from the indexes
        - Add the terms, frequency data, source index and database to the meta-search index
        - Post-Process indexes for special types of data (e.g. to create "geographical coverage" indexes using information about place names, etc.)

Although this work is still in the initial stages, we have been able to test the concept using existing Cheshire II servers and databases. As noted above we are working on automating the Meta-indexing process and will be continuing this development and testing appropriate solutions for Cross-Domain Resource discovery throughout this project.

5 Testbed Development

To help develop the appropriate technologies we propose to use two large-scale information services sponsored by JISC in the UK which offer complementary data formats: The Arts and Humanities Data Service (AHDS) and the CURL (Consortium of University Research Libraries) databases and two large-scale distributed object databases in the US (The Online Archive of California, and the Making of America II databases).

Although these will be the focus of the present proposal, we will also be bringing together a consortium of related data providers who may wish to test data using the proposed Cheshire system: these include government agencies (the Public Record Office); Universities (Glasgow, Durham, Liverpool, and Oxford); as well as hybrid library projects sponsored as part of the eLib 3 programme in the UK and the Archive Working Group at Yale University (including other participants in the US, UK and Australia) the Institute for Advanced Technology in the Humanities at the University of Virginia (Contact: Daniel Pitti) in the US. We will also be providing the Cheshire technology to and participating in the development of the NESSTAR (Networked Social Science Tools and Resources) project. The NESSTAR project (http://dawww.essex.ac.uk/projects/nesstar.html) is combining the skills and knowledge of the three main partners, The Data Archive in the UK, Danish Data Archives and Norwegian Social Science Data Services with assistance in significant areas of user analysis, usability, user validation, evaluation and quality assurance from the Institute of Journalism in Norway, ASEP and JD Systems in Spain, Central Statistics Office in Ireland and Aarhus University in Denmark. The Council of European Social Science Data Archives is a sponsoring partner for the project. The project is funded by the European Commission under the Information Engineering sector of the Telematics Applications programme. It is our intention that the technology developed as part of this research proposal will serve as the basis for full-scale information systems of international prominence. We chose the AHDS, CURL, OAC, and MOA2 databases as the focus of our work for several reasons: The data sets are large and of a diverse nature; users of these services represent a broad range of technical expertise; both have a well-founded administrative structure with existing user-evaluation mechanisms (thus reducing research overhead costs); finally, the proposed system would give considerable added value to the repositories themselves, which already comprise valuable national and international resources. Both PI's have had a long-standing connection with the AHDS (An earlier version of Cheshire was installed by the PI's for the History Data Service – one of the AHDS providers). This forms part of the AHDS gateway. In addition one of the principal investigators, Paul Watry, is a member of the CURL RDD Committee, which includes development of the COPAC service in its remit.

A brief description of the data services making up the core testbed for this project follows:

The CURL Database The Consortium of University Research Libraries currently gives access to its bibliographic database via COPAC, a Z39.50 service funded by JISC and supported by Manchester Information Datasets and Associated Services (MIDAS). The COPAC service currently consists of some 3.5 million MARC records held in a central server at MIDAS, but there are plans to extend this database to non-bibliographic data resources, such as full-text and EAD-encoded documents.

Arts and Humanities Data Service The Arts and Humanities Data Service (AHDS) is a national service funded by JISC to collect, describe, and preserve the electronic resources which result from research and teaching in the humanities. This research project will focus on the current production service of four data services. The targets for AHDS include:

1. **Archaeology Data Service (ADS):** The ADS uses a proprietary DBMS system (Fretwell-Downing's VDX system) to store extensive data incorporated as part of the resource, such as geo-spatial images, aerial photography, and CAD images.
2. **Performing Arts Data Service (PADS):** PADS is currently using Hyperwave, an object oriented information retrieval system, to locally store and retrieve information retained in a variety of formats.
3. **Oxford Text Archive (OTA):** The OTA holds its entire corpus as SGML documents.
4. **History Data Service (HDS):** The History Data Service holds its data as SGML documents which point to a number of numeric and alpha-numeric data, text, digitized boundry data, and images converted from historic source documents into computer-readable form.

These four services are available via the AHDS gateway with access points determined by an extended version of Dublin Core. In addition, there are full-text versions of the History Data Service and the Oxford Text Archive (TEI-header information only) available via Cheshire clients and servers.

2. We intend, first, to convert the metadata from the ADS and PADS databases to SGML. (Full-text from OTA and metadata from HDS are already encoded in SGML and require no conversion.)
3. Then, in order to further extend the capabilities of these databases, we intend to develop a Cheshire client to access methods for all primary data types (text, image, and map-oriented data), indexing each document by as many methods as are applicable. For example, photographs will be indexed not only by the content of their images, but also by their text, their pre-assigned categories, their location, and so forth. This will necessitate development of a Cheshire client that will support formatting of SGML documents, using DSSSL or XSL. These indexed documents will provide a basis for testing retrieval algorithms as well as for pre-processing and post-processing retrieval results sets.
4. It should be noted that a Cheshire query for these data resources may necessitate interaction with one or more interoperable clients displaying a different document data type: for example, geographic information system (GIS) data sets. A GIS-oriented browser delivered in response to a Cheshire query will make it easy to ask for information pertaining to a geographic region. In devising such a system, we will be integrating the access tools developed for the DARPA-funded Berkeley Digital Libraries project into the Cheshire environment. For example, a related client is being developed by researchers at UC Berkeley to

display a union of aerial photographs from UC Santa Barbara (Project Alexandria) and UC Berkeley databases.

4. Finally, we intend to use the HDS data as a testbed for integrating numerical statistical databases and geographic databases within the Cheshire system. Online Archive of California : The Online Archive of California Project, is a two-year pilot project to develop a UC-wide prototype union database of 30,000 pages of archival finding aid data encoded using the Encoded Archival Description (EAD) SGML document type definition. This database will serve as the foundation for the development of a full-scale digital archive for the University of California System (UC) available via the Internet to diverse user communities. Making of America II : The Making of America II is a Digital Library Federation project to continue and extend research and demonstration projects that have begun to develop best practices for the encoding of intellectual, structural, and administrative data about primary resources housed in research libraries. The Making of America II Testbed Project collection will be ”Transportation, 1869-1900”, particularly the development of the railroads and their relationship to the cultural, economic, and political development of the United States. It will comprise multimedia information coordinated by SGML metadata and “hub” documents.

In addition to using data from the data services cited above, we will also focus on identifying additional text and bibliographic resources to interact with the numerical and geo-spatial data sources. We plan to include, for example NESSTAR (discussed above) and the UC Data archive, and the existing Digital Library Initiative database at Berkeley as part of these extended resources.

6 Dissemination and Communication Activities

The primary dissemination activity in the United Kingdom for the current Cheshire system has been through the HE Archives Hub Steering Group, Data Contributor's Committee, and the CURL RDD committee. This is expected to be extended to cover a number of HE and FE repositories throughout the United Kingdom.

We are supporting a number of large-scale Framework V, JISC/CEI, and RSLP initiatives, including those at MIMAS, London University, and DeMontfort, which will result in widespread dissemination and documentation of the system throughout the United Kingdom and Europe.

In the US, the Cheshire system is being disseminated through the Cheshire FTP site (ftp://cheshire.berkeley.edu/pub/cheshire) and the Cheshire project WWW site (http://cheshire.berkeley.edu/). In addition publications and presentations at public meetings are used to inform potential users of the system about its features and availability. The Berkeley project and Cheshire II system has been registered as a Z39.50 implementer #171 with the Library of Congress Z39.50 Maintenance Agency (the entry is http://lcweb.loc.gov/z3950/agency/register/entries.html#171) where contact information about the system is publicly available for those interested in Z39.50 implementations.

Current implementations of Cheshire II will be replaced by implementations of Cheshire III as the project progresses.

6.1 Publications

Larson, R. and Carson, C. "Information Access for a Digital Library: Cheshire II and the Berkeley
Environmental Digital Library" Proceedings of the 62nd ASIS Annual Meeting, Nov. 1999.

Larson, R. “Berkeley’s TREC 8 Interactive Track Entry: Cheshire II and ZPRISE” Text Retrieval Conference (TREC-8) November 16-19, 1999.

Paepcke, A., Brandriff, R., Janee, G., Larson, R., Ludaescher, B., Melnik, S. and Raghavan, S. "Search Middleware and the Simple Digital Library Interoperability Protocol." D-Lib Magazine Vol. 6 No. 1 March 2000.

6.2 Public Presentations

Ray R. Larson

- Invited presentation at the Digital Gazetteer Workshop sponsored by the NSF. (October 12, 1999)
- Presentation at the NSF Digital Libraries 2 Principle Investigator's Meeting (Cornell, October 17 1999)
- Presentation of "Information Access" paper (above) at the American Society for Information Science annual meeting (November 2, 1999)
- Presentation at TREC (Text Retrieval Evaluation Conference(November 18, 1999)
- Presentation at UC Berkeley Digital Library Seminar on SDLIP (November 22, 1999)
- 2 Presentations at the NSF Digital Libraries PI meeting (Stratford-upon-Avon, June 12-13, 2000)

Paul B. Watry

- 2 Presentations at the NSF Digital Libraries PI meeting (Stratford-upon-Avon, June 12-13, 2000)
- Presentation at the Public Record Office (January 1999)
- 2 Presentations at LUCAS Centre for Archival Studies (October 1999)

Robert Sanderson

- University of Sheffield, Humanities Research Institute (May 2000)
- University of Liverpool, International Conference (July 2000)

6.3 Visitors received

Liverpool

- Republic of the Philippines, Department of Science and Psychology
- The Council for Museums, Archives, and Librarians (Re:source)
- National Archives, Nigeria
- National Archives, Virgin Islands
- Natural History Museum, London
- HMC
- University of Michigan

UC Berkeley

- Electronic Cultural Atlas Initiative (ECAI)
- NASA Ames Research Center (Computational Sciences Division)
- Berkeley GIS Consortium
- SleepyCat Software
- Documentum
- INFour, Inc.