Prototyping Digital Libraries Handling Heterogeneous Data Sources – An ETANA-DL Case Study

by

Unnikrishnan Ravindranathan

A thesis presented to the faculty of
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Science

Dr. Edward A. Fox - Chairman
Dr. Roger W. Ehrich
Dr. Weiguo Fan

April 26, 2004
Blacksburg, VA

Keywords: Digital Libraries, archaeological information systems,
DL architecture, componentized DLs

Prototyping Digital Libraries Handling Heterogeneous Data Sources – An ETANA-DL Case Study

Unnikrishnan Ravindranathan

Committee Chairman: Dr. Edward A. Fox

Abstract

Information systems used in archaeological research have several needs that can be summarized as follows: interoperability among diverse, heterogeneous systems, making information available without significant delay, providing a sustainable approach to long-term preservation of data, and providing a suite of services to users of the system. In this thesis, we describe how digital library techniques can be employed to provide solutions to these problems and describe our experiences in creating a prototype for ETANA-DL. ETANA-DL is a model-based, componentized, extensible, archaeological Digital Library that manages complex information sources using the client-server paradigm of the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH). We have designed and developed the prototype system with the following main goals: 1) to achieve information sharing between different heterogeneous archaeological systems, 2) to make primary archaeological data rapidly available to users, 3) to provide useful services to users of the DL, 4) to elicit requirements that users of the system will have beyond the services that it supports, and 5) to provide a sustainable solution to long-term preservation of valuable archaeological data. Consequently, we describe our approach to handling heterogeneous archaeological information from disparate sources; suggest an architecture for ETANA-DL, to be validated through prototyping; and show that given a pool of components that implement common DL services, a prototype DL can be rapidly created that supports several useful services over integrated data. Further, and most fundamentally, we note that understanding complex information systems is a difficult task. Finally, therefore, we describe our efforts to model complex archaeological information systems using the 5S framework, and show how we have used the resulting partial models to implement ETANA-DL with cross-collection searching and browsing capabilities.

This work is supported by the National Science Foundation's Information Technology Research program (ITR-0325579). Any opinions, findings, and conclusions or recommendations expressed in this thesis are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Dr. Fox for the support he has provided me during the entire course of my graduate studies at Virginia Tech. I have benefited a lot from his guidance, on scholarly as well as spiritual levels, and my work would never have been a success without his encouragement.

I am thankful to Dr. Fan for helping me get started with this project, and correcting my course as necessary. I am also thankful to Dr. Ehrich for his valuable suggestions towards improving my thesis, and for helping me defend it.

I thank Dr. Flanagan for his encouragement, support, and patience in answering every question that I had for him over the past two years. His constant feedback has been the single most important factor in motivating me to work harder.

I thank Marcos for his constant guidance, helping me out whenever I ran out of ideas, and helping me outline my thesis. I thank Ricardo for his support, guidance, motivation, and helping me to review this thesis. I thank Rao for helping me with several aspects of this work. I wish all of them very successful careers ahead.

I thank all members of DLRL. The interactions that I have had with many of you over the last two years have been wonderful learning opportunities for me.

I thank the numerous directors of excavations and their staff who not only allowed us to use their primary data but also attended meetings, answered queries, and offered critiques of the proto-type and my work as it was developing. I thank especially Drs. Larry Herr and Douglas Clark ('Umayri'), Randall Younker (Jalul), Paul Jacobs (Halif), David McCreery (Nimrin), and Mr. Andrew Graham (Madaba). Without the contributions of these and others, my thesis work would have been more difficult and protracted.

Last, but not the least, I am grateful to my parents whose love and encouragement have made this work possible. I would not be who I am today without their blessings.

Table of Contents

1. INTRODUCTION	1
1.1. Problem Statement	1
1.2. Approach	2
1.3. Hypotheses	3
1.4. Contributions	3
1.5. Thesis Outline	3
2. BACKGROUND AND RELATED WORK.....	4
2.1. Archaeological Information Systems.....	4
2.1.1. Information systems in archaeology – a brief history	4
2.1.2. Overview of some related systems	4
2.2. Introduction to Digital Libraries and Concepts	5
2.2.1. What is a Digital Library?	5
2.2.2. Examples of Digital Libraries.....	6
2.2.2.1. MARIAN.....	6
2.2.2.2. NDLTD	7
2.3. OAI (Open Archives Initiative).....	7
2.3.1. Underlying technologies.....	8
2.3.1.1. HTTP	8
2.3.1.2. XML	8
2.3.2. Open Archive Concepts.....	8
2.3.2.1. Harvesting vs. Federation.....	8
2.3.2.2. Data and Metadata.....	9
2.3.2.3. What is an Open Archive?.....	9
2.3.2.4. What is a Data Provider?	9
2.3.2.5. What is a Service Provider?.....	10
2.3.3. Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH)	10
2.3.3.1. Related concepts and definitions	10
2.3.3.2. OAI Protocol for Metadata Harvesting (OAI-PMH).....	11
2.4. ODL (Open Digital Libraries).....	12
2.4.1. Example of an ODL-based DL: CITIDEL	13
2.4.2. ODL components used in ETANA-DL prototype.....	14
2.5. 5S Framework.....	14
2.5.1. Streams	15
2.5.2. Structures.....	15
2.5.3. Spaces.....	15
2.5.4. Scenarios	15
2.5.5. Societies.....	15
2.5.6. Automatic DL generation	16
3. ETANA-DL System.....	19
3.1. ETANA-DL prototype requirements.....	19

3.1.1. Application requirements	19
3.1.2. Services requirements.....	20
3.2. ETANA-DL Architecture	20
3.2.1. Handling heterogeneous archaeological data	20
3.2.2. Data provider architecture	21
3.2.3. Service provider architecture.....	21
3.3. Modeling ETANA-DL	22
3.3.1. 5S meta-model for archaeological information systems	22
3.3.2. 5S ETANA-DL model.....	24
3.3.3. Data design for ETANA-DL artifacts	25
3.3.4. Data mapping.....	26
3.4. Implementation.....	27
3.4.1. Implementation technologies.....	27
3.4.2. Data provider implementation.....	28
3.4.3. Database design overview	29
3.4.4. ETANA-DL application	31
3.4.4.1. Union Catalog.....	31
3.4.4.2. Indexing harvested data.....	31
3.4.4.3. Interfacing with ODL components.....	31
3.4.4.4. Designing an extensible and portable application	32
4. ETANA-DL Services.....	34
4.1. Information satisfaction services.....	35
4.1.1. Searching	35
4.1.1.1. Motivation	35
4.1.1.2. Implementation.....	35
4.1.2. Multi-dimensional Browsing.....	39
4.1.2.1. Motivation	39
4.1.2.2. Related concepts.....	39
4.1.2.3. Implementation.....	40
4.1.3. Recommendations	43
4.1.3.1. Motivation	43
4.1.3.2. Implementation.....	43
4.2. Domain-specific services	44
4.2.1. Object Comparison.....	44
4.2.1.1. Motivation	44
4.2.1.2. Implementation.....	45
4.2.2. Marking items for analysis and quality control	47
4.2.2.1. Motivation	47
4.2.2.2. Implementation.....	47
4.3. Value-added services.....	49
4.3.1. Annotations.....	49
4.3.1.1. Motivation	49
4.3.1.2. Implementation.....	50
4.3.2. Items of Interest (Binding service).....	51
4.3.2.1. Motivation	51
4.3.2.2. Implementation.....	51
4.3.3. Recent searches/discussions	52

4.3.3.1. Motivation	52
4.3.3.2. Implementation.....	52
4.4. Miscellaneous services.....	53
4.4.1. User management services	53
4.4.2. Collections description	54
5. ANALYSIS.....	55
5.1. Heterogeneous data handling	55
5.2. Rapid prototyping	57
5.3. User analysis	60
6. Conclusions and Future Work.....	64
6.1. Conclusions	64
6.2. Future work	65
REFERENCES.....	68
Appendix A: Installing ETANA-DL	73
Appendix B: ETANA-DL prototype XML Schema	79
Vita	82

List of Figures

FIGURE 1: MARIAN DL ARCHITECTURE (ADAPTED FROM [16])	7
FIGURE 2: ODL-BASED NDLTD ARCHITECTURE (ADAPTED FROM [18])	7
FIGURE 3: HARVESTING VS. FEDERATION.....	9
FIGURE 4: BASIC ODL MODEL	13
FIGURE 5: CITIDEL ARCHITECTURE.....	13
FIGURE 6: 5S MODEL OVERVIEW (ADAPTED FROM [3]).....	16
FIGURE 7: AUTOMATIC DL GENERATION (ADAPTED FROM [29])	18
FIGURE 8: ETANA-DL PROTOTYPE ARCHITECTURE	20
FIGURE 9: ARCHAEOLOGICAL DL META-MODEL (PARTIALLY DEVELOPED)	23
FIGURE 10: ETANA-DL 5S MODEL (PARTIALLY DEVELOPED)	24
FIGURE 11: ETANA-DL DATA ORGANIZATION	25
FIGURE 12: MAPPING BONE DATA FROM NIMRIN TO THE ETANA-DL SCHEMA (GENERATED USING MAPFORCE)	26
FIGURE 13: ETANA-DL PROTOTYPE HOME PAGE	32
FIGURE 14: CONTENT RENDERING IN ETANA-DL – DESIGN OVERVIEW	33
FIGURE 15: ETANA-DL SEARCHING SUB-SYSTEM OVERVIEW	36
FIGURE 16: SEARCHING ETANA-DL – SEARCH INTERFACE	37
FIGURE 17: SEARCH RESULTS DISPLAYED IN A PAGE-BY-PAGE FASHION	37
FIGURE 18: ADVANCED SEARCHING: A) USER SELECTS TYPE OF OBJECT; B) USER CREATES STRUCTURED QUERY ON THE BASIS OF THE SELECTED TYPE OF OBJECT.....	38
FIGURE 19: THREE POSSIBLE DIMENSIONS OF BROWSING IN ETANA-DL - SITE, PERIOD, AND ARTIFACT TYPE	40
FIGURE 20: DYNAMIC SELECTION OF BROWSABLE CATEGORIES - ALL LOCI THAT CONTAIN OBJECTS THAT HAVE A PERIOD VALUE OTHER THAN THE ONE SELECTED ARE ELIMINATED.	41
FIGURE 21: SEARCHING WITHIN A CONTEXT - ONLY RECORDS WITHIN NIMRIN'S NW PARTITION, IN N30/W20, AND DATED AS BELONGING TO IRON II, WILL BE SEARCHED	42
FIGURE 22: VIEWING SAVED BROWSING CONTEXTS - THE USER CAN RETURN TO THE CONTEXT WITH A SINGLE CLICK	42
FIGURE 23: RECOMMENDATIONS GENERATED BY THE SYSTEM - USER IS VIEWING A FIGURINE OBJECT IN THE SYSTEM.....	44
FIGURE 24: ADDING ITEMS IN ETANA-DL FOR COMPARISON	45
FIGURE 25: SELECTING AVAILABLE ATTRIBUTES FOR DIFFERENT KINDS OF OBJECTS FOR COMPARISON.....	46
FIGURE 26: EDITING ATTRIBUTES FOR OBJECT TYPE "BONE"	46
FIGURE 27: COMPARISON RESULT FOR SELECTED BONES – BONE RECORDS ARE FROM DIFFERENT COLLECTIONS HOUSED IN ETANA-DL	47
FIGURE 28: MARKING ITEMS - MARKING AN ITEM ALLOWS THE USER TO WHOM THE ITEM HAS BEEN MARKED TO VIEW COMMENTS ASSOCIATED WITH THE MARKING.....	48
FIGURE 29: VIEWING MARKED ITEM - USER CAN CHOOSE TO VIEW THE COMPLETE RECORD OF THE CORRESPONDING DIGITAL OBJECT, ADD THE OBJECT TO ITEMS OF INTEREST, RE-MARK THE ITEM, OR UNMARK IT	48
FIGURE 30: RE-MARKING AN ITEM - USER CAN CHOOSE TO REMARK THE ITEM TO THE OWNER OF THE DIGITAL OBJECT, THE ORIGINAL SOURCE OF THE MARKING, OR TO A DIFFERENT USER	49
FIGURE 31: VIEWING THE MULTI-THREADED DISCUSSION BOARD FOR AN OBJECT – AUTHORIZED USERS CAN POST NEW MESSAGES/REPLIES TO MESSAGES	50
FIGURE 32: VIEWING MESSAGE DETAILS AND POSTING REPLIES TO MESSAGES IN A DISCUSSION BOARD.....	51

FIGURE 33: ITEMS OF INTEREST FOR A USER - THE USER CAN VIEW THE COMPLETE DIGITAL RECORD, MARK THE ITEM, OR REMOVE THE ITEM FROM THE LIST	52
FIGURE 34: RECENT SEARCHES AND DISCUSSIONS INTERFACE.....	53
FIGURE 35: NEW USER ACCOUNT CREATION	53
FIGURE 36: USER LOGIN PROMPT	54
FIGURE 37: DESCRIPTION OF VARIOUS COLLECTIONS IN ETANA-DL.....	54
FIGURE 38: BREAKDOWN OF TIMES REQUIRED FOR VARIOUS STAGES IN CONVERTING AN ARCHAEOLOGY SITE INTO AN OPEN ARCHIVE FOR THE CURRENT PROTOTYPE.....	56
FIGURE 39: LINES OF CODE FOR IMPLEMENTING SERVICE VS. LINES OF CODE REUSED FROM COMPONENTS	58
FIGURE 40: PERCENTAGE OF EFFORTS REQUIRED FOR DIFFERENT STAGES OF THE PROTOTYPING LIFE CYCLE: A) ALL SERVICES, B) COMPONENT-BASED SERVICES, AND C) NON-COMPONENTIZED SERVICES	60
FIGURE 41: XML SCHEMA OVERVIEW (GENERATED USING XMLSPY).....	79
FIGURE 42: DETAILED SCHEMA DIAGRAMS FOR THREE DIGITAL OBJECTS IN ETANA-DL (GENERATED USING XMLSPY)	79

List of Tables

TABLE 1: OAI VERBS	11
TABLE 2: ODL COMPONENTS (USED IN ETANA-DL).....	14
TABLE 3: THE 5 SS IN THE 5S MODEL (ADAPTED FROM [3])	16
TABLE 4: MAPPING SPATIAL DATA FOR VARIOUS SITES TO THE CORRESPONDING 5S STRUCTURAL MODEL	26
TABLE 5: ETANAMULTIDBROWSE DATABASE TABLES – AN OVERVIEW	29
TABLE 6: ETANA DATABASE TABLES - AN OVERVIEW	30
TABLE 7: OVERVIEW OF ETANA-DL SERVICES	34
TABLE 8: HETEROGENEOUS DATA IN ETANA-DL - AN OVERVIEW	55
TABLE 9: CONVERTING PARTNER ARCHAEOLOGY SITES INTO OPEN ARCHIVES	56
TABLE 10: ANALYSIS OF PROTOTYPE USING THE METRIC OF LINES OF CODE	57
TABLE 11: ANALYSIS OF PROTOTYPE USING THE METRIC OF TIME FOR SERVICE DEVELOPMENT (IN DAYS).....	59
TABLE 12: USER ANALYSIS OF VARIOUS SERVICES IN ETANA-DL.....	61
TABLE 13: SUMMARY OF HYPOTHESES AND SECTIONS/CHAPTERS WHERE THEY ARE DEMONSTRATED/PROVEN	64

1. INTRODUCTION

1.1. Problem Statement

Archaeological research, like many other scientific areas, results in the production of vast quantities of information. An important characteristic of this information is its heterogeneity – field records, GIS records, photos, drawings, 3-D models, texts, databases, video, audio, statistical programs, and many more that are used in the numerous disciplines that work together on archaeological projects. Information systems are starting to be widely used in archaeological research to store and process the information generated in the field and in laboratories. These systems are tailored to meet the individual requirements of the projects, are monolithic in nature, and most importantly use different schemes to organize and store the data. This makes it very difficult to achieve **interoperability** amongst these systems. Archaeological research greatly depends on typologies, comparisons, and existence of relationships with information from other projects/sites, things that are possible to accomplish in a real-time fashion only if these various systems are highly interoperable.

Publication of results in archaeological research is difficult because of traditional methods of publication. Typically, results of a season of excavation are published in a preliminary single-season report at the end of the season and years later in a full final report. The first publication of results is often in an official record issued by the government in the country of the excavation. This information is only available to limited audiences because of the obscurity of some of the publications, the fact that they are issued in a variety of countries, and high costs involved in paper-based publication. The increasing use of information systems in archaeology is slowly overcoming some of these limitations. However, making full information available only after substantial delay, or long after it is produced, hinders the activities of researchers who work in parallel on the same site or other sites. What is desirable is a highly efficient information system that would **make the primary data available as soon as it is produced**, and which can be used both in the field and out. Moreover, many of the tailor-made archaeological information systems do not address the issues related to **long-term preservation** and dissemination of information. Distributed and replicated existence of valuable information is necessary to ensure that the information is preserved for future use.

Archaeological information systems that, in addition to storing and retrieving the information, provide **services** similar to those provided by modern Digital Libraries (DLs), would be highly desirable. Having a single unifying system that is able to intelligently manage heterogeneous information from several sites along with providing a rich array of services, including those specific to archaeology – like GIS visualization systems, object comparisons, complex workflow management, etc. – would greatly help archaeological research. Such a system would support one-stop shopping for students studying archaeology and for the public interested in archaeology.

1.2. Approach

Our approach to dealing with the problems presented in the previous section is to create a Digital Library for archaeology – ETANA-DL [1]. ETANA-DL is a model-based, extensible, componentized DL that manages complex information sources using the client-server paradigm of the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) [2].

The target audience of ETANA-DL, mainly archaeologists and the public, are not generally aware of the powerful services an archaeology DL can provide. Analogously, computer scientists are not well versed in archaeological research and the requirements that archaeologists have regarding such information systems. Requirements elicitation is critical to the success of our DL, because the services that ETANA-DL will support depends upon the requirements of the archaeologists. Software engineering research has shown that rapid **prototyping** of complex information systems can be a very powerful tool for collecting requirements from users of the system, as well as testing the feasibility of the approach to be used in creating the system. We describe the design and implementation of our prototype Digital Library as well as the various services that the prototype DL supports. Our prototype is designed mainly from **components** in a very rapid fashion.

In order to handle heterogeneous data from several different sites (information sources), each of them using custom schemes to store their information, we convert these sites into **Open Archives** and expose their data using a common metadata format. Having this global view of data makes it easier to integrate and handle data at the service provider. We harvest the data into a central Union Catalog and provide services over the harvested data. Using the prototype, we show that our approach to creating ETANA-DL can successfully handle heterogeneous data from different data sources, and we can provide useful services on top of the aggregated data. By harvesting data from the different sites on a frequent basis, we can make data available to users of the DL without significant delay. In addition, by increasing the number of service providers that harvest data from various sites, we ensure replicated existence of data, thus providing an easily maintainable approach to long-term preservation of data.

Understanding such complex information systems is a very difficult task. We use the 5S approach to model certain aspects of the prototype. The **5S (Streams, Structures, Scenarios, Societies, Spaces) framework** [3] is a comprehensive modeling tool that allows a DL designer to describe most aspects of a DL. We use the resulting partially extended 5S meta-model as the basis for providing integrated search and browse services across multiple collections/sites.

1.3. Hypotheses

We hypothesize that:

- a). the development of information systems (e.g., digital libraries) that address the needs described in Section 1.1 (i.e., managing interoperability among diverse heterogeneous systems, making primary data available in a timely fashion, supporting long-term preservation of data, and providing a full suite of desired services),
- b). can be efficiently and effectively accomplished in a manner summarized in Section 1.2 (i.e., employing the 5S framework, applying and extending the Open Archives approach, and building upon componentized frameworks like Open Digital Libraries).

1.4. Contributions

The main contributions of this work are:

- A scalable and easily manageable approach to handling complex, heterogeneous data from different information sources.
- An approach to prototyping information systems that handle heterogeneous data from disparate sources and provide *Services* useful to different *Societies*.
- An approach, based on the 5S theory, to integrate the harvested data and provide a unifying system to search, browse, analyze, and use the data.
- A rapidly generated, componentized, extensible, portable, archaeological DL prototype.
- A maintainable, distributed approach to long-term preservation of data that makes primary data at a site available without significant delay.

1.5. Thesis Outline

Chapter 2 presents background on archaeological information systems as well as Digital Libraries in addition to introducing Digital Library theory and concepts.

Chapter 3 presents the ETANA-DL prototype's requirements, architecture, and design. It also discusses our efforts to model ETANA-DL and several key implementation issues.

Chapter 4 explains the services that the ETANA-DL prototype supports.

Chapter 5 presents our analysis of this approach to creating Digital Libraries.

Chapter 6 presents our conclusions, as well as recommendations for future work on the prototype.

2. BACKGROUND AND RELATED WORK

In this chapter, we put together background information related to this work. The readers are introduced to information systems designed for helping archaeologists. Various concepts related to DLs and some examples of DLs are briefly discussed. The focus of this chapter is on discussing three technologies that form the cornerstone of this work: Open Archives Initiative (OAI), Open Digital Libraries (ODL), and the 5S framework.

2.1. Archaeological Information Systems

In this section, we provide a brief overview of evolution of information systems used in archaeological research (Section 2.1.1). We then discuss some related archaeological information systems (Section 2.1.2).

2.1.1. Information systems in archaeology – a brief history

During the 1970s, mainframe-based systems started being used in archaeology for recording information during excavations. Due to the high costs involved with usage of these systems, they never really became commonplace until the advent of personal computers and database software in the 1980s when computer-based recording of information started being widely used in archaeological projects. Many of these systems merely served as substitutes for traditional ways to store information – a group of related forms containing different pieces of information. However, these systems enabled rapid information access and easy management of information, though information that could be stored in them was restricted to text and numbers. During the late 1980s, personal computers that supported rich visual interfaces started replacing the traditional data entry systems. Richer content could now be handled – images, programs, documents, spreadsheets, etc. Advances in database systems led to the creation of relational database management systems that could store complex relational information. Several applications that could handle different types of content were created; they made it easy to record and retrieve various kinds of information. Then the advent of the World Wide Web (WWW) in the late 1990s lead to the creation of a large number of systems that published information on the Web [4].

2.1.2. Overview of some related systems

DANA-WH

DANA-WH is designed to function as a federation of distributed, interoperable databases, each with specific content of value to the study and preservation of human, cultural, and biological heritage. DANA-WH contains an array of textual, metric, and visual information about human heritage objects. In addition to textual descriptions and metric and non-metric attribute information about each object, the system displays two-

dimensional (2D) images of artifacts and fossils, as well as accurate, three-dimensional (3D) models of the artifact [5]. Because DANA-WH uses the federation approach to managing information, the content is held at the participating systems.

DigMaster

DigMaster is a database of electronically generated images and data on Persian/Classical (and some Iron II) Age figurines, housing three data sets, including one of figurines from Tell Halif, and one from Maresha. The third consists of objects housed in the Pierides Museum in Larnaca, Cyprus. Some of the goals of DigMaster include early publication of information and creating a vast database of Persian figurines [6].

Madaba Plains Project (MPP)

MPP has three main excavation sites - Tell al -'Umayri, Tall Jalul, and Tall Hisban. The MPP sites use a system that is built over a popular RDBMS software package, to record data from the excavations [7].

Perseus

The Perseus project is an digital library of resources for the study of the humanities and focuses on developing tools to provide users with improved access to various types of materials [8, 9].

Tell Nimrin – A virtual archaeological site

Virtual Nimrin presents findings of the Tell-Nimrin project. It includes electronic versions of field books, selected photos, and publishable drawings, as well as a comprehensive collection of reports, including materials that have been published in hardcopy. The locus sheets in Nimrin organize various kinds of records produced in the field, making it easy for an archaeologist to view records associated with a locus in addition to specific information about the locus. Nimrin uses maps and drawings to help users navigate the data space, and to make relationships between different excavation units explicit. A query interface allows the users to query various digital artifact databases, and the search results allow a user to navigate to other digital objects, like the locus sheets associated with a field record [10].

2.2. Introduction to Digital Libraries and Concepts

2.2.1. What is a Digital Library?

In a legendary article published in the *Atlantic Monthly* in July 1945 soon after World War 2 [11], Vannevar Bush, then the Director of the Office of Scientific Research and Development wrote of a “... *device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding*

speed and flexibility ...”. The device he was envisioning, Memex, as agreed to by many experts in the field, was the precursor to modern day Digital Libraries.

Much research and development has taken place in this field since then. However, despite being a scientific field, there has been little consensus on a widely accepted definition of a Digital Library. The following are two definitions that try to summarize what Digital Libraries are:

- Digital libraries are organizations that provide the resources, including the specialized staff, to select, structure, offer intellectual access to, interpret, distribute, preserve the integrity of, and ensure the persistence over time of collections of digital works so that they are readily and economically available for use by a defined community or set of communities [12].
- Digital libraries are complex data/information/knowledge systems that help: satisfy the information needs of users (societies), provide information services (scenarios), organize information in usable ways (structures), manage the location of information (spaces), and communicate information with users and their agents (streams) [13].

Though there are various definitions of DLs that exist, DLs share common characteristics. One such summary of characteristics is provided by the Association of Research Libraries [14]:

- The digital library is not a single entity;
- The digital library requires technology to link the resources of many;
- The linkages between the many digital libraries and information services are transparent to the end users;
- Universal access to digital libraries and information services is a goal;
- Digital library collections are not limited to document surrogates: they extend to digital artifacts that cannot be represented or distributed in printed formats.

2.2.2. Examples of Digital Libraries

2.2.2.1. MARIAN

Multiple Access and Retrieval of Information with AnnotationNs (MARIAN) is a DL system with an extensible set of services including browsing, searching, retrieving, automatic collection building, and uniform preservation over networked collections. MARIAN started as an Open Public Access Catalog (OPAC), to later become a complete, modern DL. MARIAN is a monolithic DL with a layered architecture [15]. The following figure highlights its architecture [16].

Figure 1: MARIAN DL architecture (adapted from [16])

2.2.2.2. NDLTD

NDLTD (Networked Digital Library of Theses and Dissertations) is an initiative for creating a DL of theses and dissertations on a world-wide basis [17]. The collections are maintained by the individual universities. Every university that wants to make its theses and dissertations globally accessible can implement data providers to expose the related metadata in standard formats (Dublin Core, ETD-MS, etc.). The metadata is then harvested into a Union Catalog, on top of which various services are provided to users of the DL. Some of the services provided include searching, browsing, viewing new items in the collection, reviewing items, etc.

The following figure shows how various ODL components are put together to create the DL [18].

Figure 2: ODL-based NDLTD architecture (adapted from [18])

2.3. OAI (Open Archives Initiative)

The Open Archives Initiative develops and promotes interoperability standards that aim to facilitate the efficient dissemination of content [2].

The early efforts to achieve interoperability amongst DLs led to the creation of custom solutions such as the Deinst system [19] and Z39.50 [20]. However, two key problems

persisted: end users were faced with multiple search interfaces making resource discovery harder, and there was no machine-based way of sharing the metadata. In response, the Santa Fe meeting led to development of the OAI, which promotes interoperability via developing OAI-PMH as an open standard. OAI-PMH is a low-cost mechanism for harvesting metadata records from one system to another [21].

2.3.1. Underlying technologies

2.3.1.1. HTTP

HTTP (Hyper-Text Transfer Protocol) is a standard for transferring Web pages over the Internet [22]. It is a request-response protocol in which a client sends a request for a page, and a web-server, that interprets the request, sends a response back to the client. HTTP is a connection-oriented, stateless protocol. The OAI-PMH makes use of HTTP for sending requests to the web-server and receiving XML responses. The request parameters are encoded in the HTTP GET and POST requests.

2.3.1.2. XML

XML (Extensible Markup Language) is a tag-based language that has been designed with the purpose of making data interchange easier amongst systems [23]. XML allows the developers of the system to devise their own tags that add meaning to the data being interchanged, and the system can then interpret the data in its own custom way. XML is used in OAI-PMH to send responses back to the client.

2.3.2. Open Archive Concepts

2.3.2.1. Harvesting vs. Federation

There are two common approaches that are used in DLs to provide services to users when the content is distributed among several DLs: Harvesting and Federation.

In case of Federation, a search request is distributed to other DLs that in turn distribute the request to others in a recursive fashion. The results are then accumulated together and returned to the originator of the request. Modern Peer-to-Peer protocols such as Gnutella use this approach to distribute the queries and fetch the results. Alternatively, in federated search, a client or DL front-end may broadcast queries and collect results directly. Federation has the advantage of distributing the load of searching across various DLs, but suffers from the fact that it depends on the high availability of the content providers. In addition, there may be significant latency in generating the search results.

In contrast, Harvesting is the process of pre-fetching the metadata from other DLs and then providing services on the local copy of harvested metadata. Such an approach is used in major commercial search engines like Google. Since the search is performed on a local copy of the metadata, results can be returned with very low latency. Moreover,

since harvesting is carried out on a periodic basis, even if any of the content-provider DLs is unavailable, we can still search its metadata.

Figure 3 illustrates the concepts of Harvesting and Federation [18].

The OAI-PMH uses harvesting as its approach. The assumption is that though there is an increased burden on the service provider, it can be overcome by increasing the resources of the service provider, thereby leveraging the other advantages of the harvesting approach, such as increased availability, low latency, etc.

Figure 3: Harvesting vs. Federation

2.3.2.2. Data and Metadata

Metadata is data about data. The question of whether to harvest data or metadata is handled by different DLs in different fashions. Some DLs harvest entire digital objects whereas others only harvest the metadata. In most cases, harvesting the metadata only is sufficient to provide services to users. An example of a digital object is an ETD (Electronic Thesis or Dissertation) whereas its metadata is information such as title, author, abstract, etc.

2.3.2.3. What is an Open Archive?

An Open Archive is an archive that implements the OAI-PMH and is able to respond to the corresponding requests and generate appropriate responses.

2.3.2.4. What is a Data Provider?

A data provider maintains one or more repositories (Web servers) that support the OAI-PMH as a means of exposing metadata [2]. In other words, data providers expose their data/metadata in standard formats so they can be harvested by others.

A data provider can be thought of as an additional layer that is implemented over an existing system without disturbing the underlying system. Application logic, embedded into this layer, can be used to determine what information is to be exposed. In other words, a data provider has complete control over the information that is available for others to harvest.

There are several components available throughout the web that can ease the task of implementing a data provider.

2.3.2.5. What is a Service Provider?

A service provider issues OAI-PMH requests to data providers and uses the metadata as the basis for providing value-added services [2]. Service providers often harvest metadata from several different data providers. Fortunately, for the support of data flow and pipeline architectures, service providers can themselves be data providers for other service providers.

2.3.3. Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH)

2.3.3.1. Related concepts and definitions

Harvester

A harvester is a piece of software that uses OAI-PMH to harvest metadata from the various OAI-compliant repositories that it is configured to harvest from. A harvester is run by the service provider on a periodic basis to ensure that the harvested metadata is in synchronization with the Open Archive.

Set

The definition of Set is not pre-set, and it can be thought of as an agreement between the Open Archive and the service provider to group objects, to enable the service provider to selectively harvest parts of the archive. Often, a set is used to aggregate objects covering similar timespans, or related to particular topics or sources. Sets may in turn have subsets.

Timestamp

Every record has an associated timestamp that is used to indicate the time the record was last updated. As with sets, timestamps are used by harvesters for selective harvesting. Timestamps are used to avoid harvesting all records from an archive each time the harvester is run, thus wasting precious network bandwidth and causing a bottleneck at the data provider as well as the service provider. Instead, OAI-PMH allows harvesting by specifying a timestamp value, upon which only records that are modified after that value are returned.

Identifier

Every item in the repository needs to have a unique, persistent identifier that identifies the item in that repository. It is advisable to have a globally unique identifier for every item, though it is not necessary.

Record

A record, associated with individual items, is what is returned in a standard metadata format by an Open Archive. A record can be thought of as XML-encoded metadata associated with an item in the repository.

2.3.3.2. OAI Protocol for Metadata Harvesting (OAI-PMH)

The OAI-PMH consists of a set of six verbs that any Open Archive repository needs to respond to. Every OAI-PMH request contains one of these verbs and may contain additional parameters to qualify the verb.

Table 1 provides a summary of the six verbs, their functions and the various parameters that can be supplied [24].

Table 1: OAI verbs

Verb	Function	Parameters
Identify	Returns description of an archive	None
ListMetadataFormats	Retrieves available metadata formats for an archive	identifier (optional)
ListSets	Retrieves set structure of a repository	resumptionToken (exclusive)
ListIdentifiers	Retrieves only the identifiers for the records in the repository	from (optional) until (optional) metadataPrefix (required) set (optional) resumptionToken (exclusive)
ListRecords	Harvests records from a repository	from (optional) until (optional) metadataPrefix (required) set (optional) resumptionToken (exclusive)

GetRecord	Retrieves individual metadata record from a repository	identifier (required) metadataPrefix (required)
-----------	--	--

2.4. ODL (Open Digital Libraries)

An Open Digital Library is a network of extended Open Archives that work together to supply the services required by information seekers [18].

The following are the major highlights of the ODL approach to building DLs:

- ODL provides us with a framework for building Digital Libraries from components, thus leveraging all the benefits of component-based software engineering: component reuse, loose coupling between different components of the system, and the ability to modify a component of the system without affecting other parts of the system if the interface exposed by the component remains unchanged.
- ODL can be considered as an extension to the works of the Open Archives Initiative. ODL components communicate with each other using the HTTP GET and POST requests. The verbs that the components can understand include the OAI verbs (Identify, ListMetadataFormats, ListSets, ListIdentifiers, ListRecords, GetRecord), as described in the previous section. An additional verb (PutRecord) has been added to the list of OAI verbs.
- The XOAI is a general extension to the OAI protocol [18]. By extending OAI, additional parameters can be supplied to the components for each verb. The semantic meanings of the parameters are set by the designer of the component. Components that are more complex depend on complex input parameters. As a solution to this problem, Suleman suggests combining the parameters to form one of the standard OAI verb parameters. The complex parameters are then interpreted by the individual components. We give an example to make this clearer. Consider a search engine component. Given a search term, a very basic search engine component returns documents relevant to the query (which are each returned as a complex XML record, and which are then parsed by the interface component for display). However, the component designer can create a more complex component that only returns documents that have ranks from x thru y . For this component, along with specifying the search term that is used to decide the relevance of the query, we also need to provide inputs suggesting an interest in documents ranked from x thru y . In this case, we can provide a more complex input parameter string to the component that encodes the query string, and the document relevance range.

Figure 4 illustrates the basic ODL model and how the model can be translated into a DL service.

Figure 4: Basic ODL model

2.4.1. Example of an ODL-based DL: CITIDEL

CITIDEL (Computing and Information Technology Interactive Education Digital Library) [25] is a part of the NSF's NSDL program and is a Digital Library that houses collections and resources in the computing field. CITIDEL acquires its content thru a variety of means: harvesting, manual input, focused crawling, etc. It provides services useful to different "societies". Some of the major services that CITIDEL provides are searching, browsing using multiple classification schemes, binding, annotation, resource submission, etc. CITIDEL is designed as a componentized DL, similar to ETANA-DL, and many services are implemented using ODL components.

Figure 5 shows the various services and societies that CITIDEL supports as well as its generic architecture (from the original proposal submitted to NSF [26]).

Figure 5: CITIDEL architecture

2.4.2. ODL components used in ETANA-DL prototype

The ETANA-DL prototype uses several ODL components for providing services. Table 2 provides a summary overview of the protocol that these components adhere to. We only list the verbs that we use in the prototype.

Table 2: ODL components (used in ETANA-DL)

Service	ODL Protocol	Key Verbs	Purpose/Comments
Union Catalog	ODL-Union v1.0	GetRecord ListIdentifiers ListRecords	Inherited from OAI-PMH.
		ListMetadataFormats	List all metadata formats that the Union Catalog supports.
Searching	ODL-Search v1.0	ListIdentifiers ListRecords	“set” parameter used to encode the query.
Annotation	ODL-Annotate v1.0	PutRecord	Used to store an annotation.
		ListRecords	Used to retrieve annotations, input encoded in the “set” parameter.
Recommendation	ODL-Recommend v1.0	PutRecord	Provides input to the recommendation engine.
		ListIdentifiers	Lists all recommended resources, input provided by the “set” parameter.

2.5. 5S Framework

Digital Libraries are complex information systems that stand at the intersection of Information Retrieval, Database Theory, Library Sciences, Hypertext, and Human-Computer Interaction. The lack of theories and definitions in this field have made it a very hard, time consuming, and expensive affair to build DLs. Most DLs were built in response to requirements of a specific community and therefore did not benefit from DL and software engineering design experiences [3]. New models and theories were required to better understand, design, and build DLs of the future. Fox et al. have proposed the 5S (streams, structures, spaces, scenarios and societies) theory as a framework for providing theoretical and practical unification of digital libraries. They also argue that the 5S theory can be used in understanding requirements and modeling other information systems [3].

2.5.1. Streams

Streams are sequences of an arbitrary type (e.g., bits, characters, etc.) [3]. Streams are used to model the digital content of a DL. Streams can represent both static and dynamic content. Examples of static streams are text documents like books and articles, photos, drawings, and graphics. Examples of dynamic streams are audio and video.

2.5.2. Structures

A structure specifies the way in which parts of the whole are arranged or organized [3]. The structural model is used to describe taxonomies, database schemas, data structures, metadata formats, etc.

2.5.3. Spaces

The spatial model is used to represent various kinds of spaces and their associated relational operations. In addition to representing various multidimensional spaces (e.g., 1D, 2D), the spatial model is used to represent various IR-related spaces like vector spaces and probability spaces, indexing schemes, and contexts for user interface interactions.

2.5.4. Scenarios

Scenarios are used to represent the various “ways” to use the system. In addition to stories related to user interface interactions, the scenarios model can be used to describe component interactions, processes, services, etc.

2.5.5. Societies

The Societies model describes the various entities that interact with the DL and their relationships. The entities can be classified as managers, responsible for providing the various DL services, and actors, responsible for using those services.

Figure 6 shows how the various Ss are related together to model a basic DL [3]. Table 3 summarizes the various Ss in the 5S model.

Figure 6: 5S model overview (adapted from [3])

Table 3: The 5 Ss in the 5S model (adapted from [3])

Models	Examples	Objectives
Stream	Text; video; audio; image	Describes properties of the DL content such as encoding and language for textual material or particular forms of multimedia data
Structures	Collection; catalog; hypertext; document; metadata; organization tools	Specifies organizational aspects of the DL content
Spaces	Measure; measurable; topological; vector; probabilistic	Defines logical and presentational views of several DL components
Scenarios	Searching; browsing; recommending;	Details the behavior of DL services
Societies	Service managers; learners; teachers; etc.	Defines managers, responsible for running DL services; actors, that use those services; and relationships among them

2.5.6. Automatic DL generation

Model vs. Meta-model

A meta-model defines the rules and the semantics for creating a model as well as the architecture of the model. A model can be thought of as an instance of the meta-model.

DL meta-models are generally specified by DL experts whereas models are created by domain experts.

5SL

5SL is a domain-specific, declarative language with formal semantics used for conceptual modeling of digital libraries [27]. 5SL incorporates different syntaxes and semantics for representation of the streams, structures, societies, scenarios, and spaces models. 5SL models are instantiations of the corresponding 5S meta-models for DLs.

5SGraph

To aid DL experts to easily specify DL meta-models and domain experts to create DL models without worrying about the complex syntax of 5SL, a graphical interface-based tool is desirable. 5SGraph is an easy-to-use, graphical interface-based tool that helps DL designers specify the meta-model and create instances of the meta-model [28]. The tool then automatically generates the necessary 5SL files.

5SLGen

Given the output of 5SGraph (5SL files that represent a DL model), the next step is to automatically generate the DL, using a pool of components that implement various services. 5SLGen is a tool that uses a scenario-based approach to automatically generate DLs [29]. 5SLGen takes as its input the 5SL model, and a pool of ODL components, and generates, as its output, framework code for the DL specified by the model.

DL Generation

The 5S approach to automatic DL generation consists of three major phases:

- **Requirements Engineering and Analysis:** This phase involves creating the meta-model of a Digital Library by DL experts. The resulting 5S meta-model is then used as an input for 5SLGraph.
- **Design:** DL designers, often domain experts, create the 5S model, which is an instance of its corresponding meta-model. 5SLGraph is a tool that helps in creating the required 5SL files for the model.
- **Implementation:** The generated 5SL files are input to a tool like the 5SLGen which, given a pool of pre-existing components, generates a skeleton for the DL automatically. This skeleton reflects the design specified using the 5S model.

Figure 7 gives an overview of the 5S approach to automatically generating DLs.

Figure 7: Automatic DL generation (adapted from [29])

3. ETANA-DL System

In this chapter, we describe the architecture of the prototype for ETANA-DL. We discuss some of the requirements that users have for archaeological information systems, that we are trying to address in the prototype (Section 3.1). We present our approach for dealing with heterogeneous data from disparate sources and provide recommendations for the data and service provider architectures as well as for the information in ETANA-DL (Section 3.2). Understanding complex information systems is made easier by trying to model them using theoretical frameworks. We describe the partially developed 5S meta-model for archaeological information systems (Section 3.3). We conclude by discussing implementation-related issues like the structure of the databases used in the prototype, the various technologies that have been used and ways in which they interact, and our efforts to make the application portable and easily extensible (Section 3.4).

3.1. ETANA-DL prototype requirements

In this section, we provide a summary of the requirements that the ETANA-DL prototype tries to address. We classify the requirements into two categories: application requirements and services requirements.

3.1.1. Application requirements

- **Heterogeneous data handling:** The DL should handle heterogeneous archaeological data from disparate sources and provide a unifying system that supports a variety of services that address the needs of the users.
- **Rapid access to information:** The information produced at the partner sites should be available to users of the DL without significant delay. The DL should provide a solution for the problem of information generated at a site not being available for long periods of time.
- **Componentized DL:** The DL should be componentized to allow customization of various components to suit individual requirements of the partner sites without affecting the rest of the system.
- **Extensibility:** The application should be extensible in order to facilitate easier ingesting of newer kinds of data into the system.
- **Portability:** The application should be designed with portability as one of the major goals, so that mirrors can be setup at several locations without having to modify large parts of the application.
- **Ease of access:** Because users of ETANA-DL will mainly be members of the archaeology community with limited experience in using DLs, the application should be easy-to-use and intuitive.

3.1.2. Services requirements

- **Information access:** Users must be able to locate relevant information in the DL easily and accurately.
- **Archaeology specific services:** The DL should support services that are needed by the archaeology community such as ability to compare objects, mark items, etc.
- **DL services:** The DL should provide services that other traditional DLs provide. Examples include personal collections, recommendations, annotations, etc.

3.2. ETANA-DL Architecture

In this section, we describe our approach to handling heterogeneous archaeological data from disparate sources. We also describe the architectures of the data and service providers. Figure 8 shows the architecture of the prototype for ETANA-DL that is explained in detail over the next few sub-subsections.

Figure 8: ETANA-DL prototype architecture

3.2.1. Handling heterogeneous archaeological data

We follow the metadata harvesting approach advocated by the Open Archives Initiative to handle heterogeneous archaeological data [1]. Each site that acts as a data provider for ETANA-DL exposes its data in a standard format. The architecture for the data provider is explained in the next sub-section. The data is harvested into a central Union Catalog at the service provider by issuing OAI-PMH requests to the archives that the service provider is configured to harvest data from, and various services are provided on the local copy of the harvested data.

Whenever a record is updated at the source, the timestamp associated with the record changes. Service providers only harvest from a data provider those records which have

been updated since the archive was last harvested. Thus, running the harvester at the service provider on a regular basis addresses the issue of the data changing at the source, and ensures these changes are visible to the users of the DL.

By using the OAI approach for data handling, we also try to address the issue of long-term preservation of data. If data at the data provider is lost or the data provider is unavailable, the service provider (in this case, ETANA-DL) has a local, harvested copy of the data and can continue providing services. Thus, by having multiple service providers, we are in effect replicating data.

3.2.2. Data provider architecture

Nimrin [10], Lahav [30], and Umayri [31] – the three sites that are data providers for the current ETANA-DL prototype – use different schemas and systems to store the information. As an example, Nimrin and Umayri use relational databases to store information, whereas the figurines in Lahav are stored as tab-delimited records in a text file. Further, Nimrin and Umayri use different schemas for storing the information. Such heterogeneity is common in several kinds of information systems, even for those outside the archaeological domain, and a unifying system should be able to handle the inherent complexity caused by this heterogeneity.

We have implemented semi-automatic data-mapping components that convert the data from its local view as known to the system at the site, to a global view as known to ETANA-DL. As shown in Figure 8, these components provide a layer of abstraction for the data provider layer, and hence pre-existing data provider components can be re-used.

The main advantages of using this architecture for data providers are:

- We shift the complexity of heterogeneous data handling from the end of the service provider to the end of the data provider.
- If the schemas for representing the data change at a local site, the system remains unaffected, as only the data-mapping components at that local site need to be re-configured.
- Sites might want to restrict sensitive information from being exposed to other users. Such kind of filtering can be incorporated into the middle layer of the architecture (i.e., the data mapping layer), thus requiring the sites to only modify one layer.

Further implementation details regarding data provider components used in the current ETANA-DL prototype and the steps to implement a data provider are described in Section 3.4.

3.2.3. Service provider architecture

The harvester at the service provider is configured to harvest data from each of the partner sites into a Union Catalog. The harvester is run on a frequent basis to ensure that

the information at the service provider (ETANA-DL) is in synchronization with information at the original sites.

The data harvested into the Union Catalog is indexed in two formats – as inverted files to provide IR-like services, and as relational databases to provide DB-like services. The search engine uses the inverted files to provide the searching service, whereas the browse engine uses the relational databases to provide the dynamic, multi-dimensional browsing service. Other services depend on custom databases in addition to the Union Catalog and the indexed archaeological data in the relational databases.

Many of the services in the current ETANA-DL prototype are pre-existing ODL components that have been modified to suit our requirements. These components communicate with each other and the web-interface using XOAI. The web-interface serves as the glue that binds these various components together.

More information about the components used in the ETANA-DL prototype and the structure of the databases is provided in Section 3.4.

3.3. Modeling ETANA-DL

Most archaeological projects approach the handling of data and information in diverse ways. In order to address the problems associated with managing such heterogeneous data and processing in archaeology, we apply 5S (Streams, Structures, Spaces, Scenarios, and Societies) to model archaeological data and procedures. 5S has already been used to create meta-models for education-oriented digital libraries [32]. We are creating a unified (meta-)-model for archaeological systems based on the 5S framework for information systems.

3.3.1. 5S meta-model for archaeological information systems

Figure 9 presents a graphical representation of an archaeological meta-model, where relationships crossing sub-models show points where concepts logically contained in one of the ‘S’ models, are composed or work together with concepts in other Ss, to define DL constructs.

In the archaeological setting, streams represent the enormous amount of dynamic multimedia information gathered and created by specialists. Examples include photos and drawings of excavation sites, loci, or unearthed artifacts; audio and video recordings of excavation activities; textual reports; and 3D models which are used to measure, reconstruct, and visualize archaeological ruins.

Structures represent the ways archaeological information is organized along several dimensions. Examples include site organization, temporal organization, and taxonomies of specific unearthed artifacts like bones and seeds. Particularly important is the structure of sites, since it defines the core units of knowledge in the archaeological DL. It can be generalized that specific regions of archaeological interest are subdivided into sites,

normally administered and excavated by different groups. Each site is further subdivided into fields, excavation units and loci, the latter being the nucleus of the excavation. Material or artifacts found in different loci are organized in containers for further reference and analysis.

Figure 9: Archaeological DL meta-model (partially developed)

Spaces model spatial and geographic distribution of found artifacts, user interfaces, and metric or vector spaces. The latter are used to support retrieval operations, calculate distances, and constrain searches spatially. User classes defined in the society model include archaeologists and the public who use DL services, the behavior of which is specified in the scenario model. Besides societies of users, service managers, which are electronic entities responsible for running services, are also specified in the societies model. Scenarios and Societies act together to capture and model not only the services used by the public (e.g., search, browse, annotate, recommend), but also domain specific services used by archaeological experts. More specifically, there are four main general classes of DL services: repository building, value added, domain specific, and information satisfaction. Components in the space and structure models also interact with each other. For example, co-ordinate systems and different taxonomies are used in metadata records to describe different parts of the site.

3.3.2. 5S ETANA-DL model

The generic meta-model for archaeological information systems can be instantiated to create specific models of archaeological DLs. An example is the ETANA-DL model shown in Figure 10, which uses the meta-model to try to unify the models of several archaeological systems.

Figure 10: ETANA-DL 5S model (partially developed)

Current streams present in ETANA-DL include figurine images, and preliminary (final) reports. The core of ETANA-DL is its Union Catalog which merges metadata harvested from three sites: Umayri, Nimrin, and Halif. Each site has its own organization that can be mapped to portions of the meta-model. In particular, quadrants represent fields in Nimrin; square (Umayri, Nimrin) and area (Halif) represent the more general notion of an excavation unit in the respective sites; bags, pails or baskets collect artifacts (bones, seeds, and figurines) found in different sites' loci during excavation. Each site also has its own archaeological periods (chronology). The earliest occupation in Nimrin is represented by stone and mud-brick walls from the Middle Bronze I (MBI) period (c. 2000 B.C.E.); Halif has occupation history from chalcolithic through Modern Arab; the chronology of Umayri is from Paleolithic through the modern age. In ETANA-DL, each site mentioned above has its site-specific coordinate system. An example is Nimrin that uses a Cartesian/Polar system. The ETANA-DL user interface is web-based. Repository building services of ETANA-DL includes harvesting and converting services. Indexing,

annotation, and binding service are valued added services built over the repository. These services support information satisfaction services such as searching and browsing, and archaeology specific services. The archaeology specific services include artifact analysis and interpretation; an object comparison service, which allows archaeologists interested in similar artifacts to make comparisons across sites; and a statistics generation service which may support construction of hypotheses that correlate different types of historical and pre-historical societies, with the found objects, and with their spatial/temporal/physical properties.

3.3.3. Data design for ETANA-DL artifacts

In a unifying system, the schema for representing data is under continuous evolution. As newer kinds of sites and data are handled, the schema keeps getting richer. Thus, the schema for the ETANA-DL prototype has been designed with flexibility and extensibility as important goals. The 5S model for ETANA-DL has helped us in understanding the manner in which information needs to be structured. Appendix B shows the XML schema designed for the prototype.

Figure 11: ETANA-DL data organization

Figure 11 illustrates the design of the schema. In the current prototype, ETANA-DL handles three kinds of digital artifacts: bones, seeds, and figurines. The digital records of each of these kinds of artifacts have attributes that are specific for that artifact type. For example, name of animal and type of bone are important attributes for the bone artifact type. In the case of seed records, the seed species is an important attribute. However, many of the artifacts share common attributes. For example, in the current prototype, every digital object belongs to a collection (site), a partition within the site, a locus within the partition, and a container within the locus. If we consider an object-oriented view to

designing data, it suggests that such attributes should be members of a base class. All derived classes should inherit members from the base class, and can further specialize the class by adding attributes specific to the class.

3.3.4. Data mapping

In Section 3.3, we described a structural model for the spatial organization of digital artifacts in the current prototype. Table 4 shows examples of spatial mappings for three partner sites. Each column in the table shows how we map corresponding spatial units for various sites to our structural model. For each site, we give the nomenclature for the spatial unit as known to archaeologists associated with the site, and an example of the spatial unit.

Table 4: Mapping spatial data for various sites to the corresponding 5S structural model

Site	Partition	Sub-partition	Locus	Container
Lahav	Field	Area	Locus	Basket
	I	A8	A8074	224
Nimrin	Quadrant	Quadrant Value	Locus	Bag
	NW	N25/W50	96	240
Umayri	Field	Square	Locus	Pail
	A	7J59	001	12

Figure 12: Mapping bone data from Nimrin to the ETANA-DL schema (generated using MapForce)

Figure 12 shows the mapping from one of the *Bone* database tables in Nimrin to the ETANA-DL XML schema, generated using a commercially available data-mapping tool. Direct mappings between data may not always exist. As an example, consider the spatial mapping of *Partition* for Nimrin. As shown in the figure, we concatenate values from two columns in a table to achieve a meaningful mapping. For some sites, we had to deal with data being stored in multiple relational tables, which are not always normalized. In other cases, data was stored outside databases in custom formats.

3.4. Implementation

In this section, we describe several key implementation related issues. We describe some of the technologies used in the ETANA-DL prototype and outline the steps to be taken to convert a site into an Open Archive, so that data can be harvested by the DL. The structures of the key databases used in ETANA-DL are explained. Finally, we describe how the various ODL components fit together in the context of ETANA-DL as well as the design of the Web application.

3.4.1. Implementation technologies

The ETANA-DL prototype targets usage of Open Source software and components. We briefly list some of the technologies that have been used in ETANA-DL.

Linux

For the current version of the prototype, we use Red Hat Linux version 9 as the platform for the application [33]. However, none of the components used in the system mandates this choice, as all of them can be used across platforms.

MySQL

MySQL is an open-source relational database management system [34]. All of the databases in ETANA-DL reside on a MySQL server.

Apache Tomcat

Because ETANA-DL is a web-based application centered on Sun's Java servlets technology, we need a web-container to manage and run the servlets. Apache Tomcat is an open-source J2EE web-container and is one of the popular web-servers used on the Internet [35].

Perl

Perl is a programming language well suited to creating web-applications because of its powerful regular expressions, large number of pre-existing modules, and a vast user base. All the ODL components used in the ETANA-DL prototype are written in Perl.

Java Servlets

The application logic and the web-interface of the current ETANA-DL prototype have been developed using Sun Microsystems' Java Servlets technology [36]. This choice makes the application portable across platforms.

3.4.2. Data provider implementation

The ETANA-DL prototype uses XML file-based data provider components at each site that is a current ETANA-DL data provider. These components use information encoded as XML records. To use such a component, we have to convert all the relevant information to be exposed into XML files at every site. The XML records adhere to the unifying XML schema that we have designed for ETANA-DL, as explained in Appendix B. In future versions of the system, it is envisioned that the XML file-based data provider components will be replaced with data providers that read the information directly from databases at local sites to eliminate the overhead associated with generation of the XML records. However, the modular nature of the system allows us to change the data provider components without affecting the rest of the system.

Implementing a data provider for a new site, in the current version of the system, consists of the following four main stages:

- **Data analysis:** During the data analysis stage, we study the characteristics of the data at a local site. We also try to analyze the dependencies that exist in the data (e.g., if locus information for a bone record is stored in a different table, there is a dependency between the two tables).
- **Data mapping:** In this stage, we try to map the schema at a local site to the unifying schema that is used in ETANA-DL. From our experience, this stage is extremely difficult because direct mappings between schemas rarely exist. This stage requires expert knowledge of the schema used to represent the information at a local site, as well as knowledge about the information system being used at the site. Once the mappings are well understood, we create data mapping components that map the data at the local site to the global schema. Often, archaeological data contains characters that are disallowed in XML records. Hence, it is necessary to encode these characters, and this is a part of the data mapping stage.
- **Generating XML-native databases:** In the current version of the system, since we are using XML file-based data providers, we have to convert all the relevant records in the databases at a site into XML records. This step can be integrated with the second stage, wherein the data mapping components generate XML files as their output.
- **Implementing the Data Provider:** Several kinds of data providers exist and are available to use and modify. However, pre-existing components might need to be setup to suit the requirements of a site. For example, if a site needs to restrict access to certain records, this logic should be a part of the data provider layer in our architecture. Once a data provider is installed and configured, the site

becomes an Open Archive, and other data and service providers can harvest data from the site using OAI-PMH.

3.4.3. Database design overview

ETANA-DL uses several databases in its data tier to store information. However, rather than describing each database in detail, we will give an overview of the relationship of these databases and the tables contained therein, to the various services/components in terms of which service/component uses which database/table. Hence, the emphasis is on “What is stored in which database/table?” rather than “How is the information stored?”

etanaunion

The *etanaunion* database stores the Union Catalog, with records harvested from the partner sites. The tables contained in *etanaunion* are created automatically by the Union Catalog component. Detailed information about digital objects is retrieved from the *etanaunion* database that uses the OAI identifier of the digital object as the primary key.

etanasearch

This database is used by the ETANA-DL search engine component. The database contains the inverted files necessary to perform Information Retrieval tasks. Tables within this database are created automatically when the IRDB search engine component is configured.

etanaannotate

The *etanaannotate* database stores the annotations attached to digital objects in ETANA-DL. The tables in this database are created automatically by the annotation engine when the component is installed.

etanasuggest

The *etanasuggest* database stores information required by the recommendation engine component to generate recommendations. As with all other databases used by the ODL components, the tables in *etanasuggest* are automatically created when the component is installed and configured.

etanamultidbrowse

The *etanamultidbrowse* database stores the indexed archaeological data required to provide the dynamic, multi-dimensional browsing service. The tables in this database are created by running a script during installation of the application. Table 5 lists the various tables in the database and describes the information stored in each table.

Table 5: Etanamultidbrowse database tables – an overview

Table Name	Purpose
idtable	Main table for all the digital objects. Primary key is the OAI identifier of the digital object. Contains classification information about each object along the three browsing dimensions in ETANA-DL – information is

	stored using foreign keys that point to the appropriate classification. Also contains information about the type of object and its parent collection. Used to provide the browsing service.
boneindex	Contains the classification of bones that allows a user to browse by bone name. The primary key of this table is a foreign key in the idtable.
seedindex	Contains the classification of seeds that allows a user to browse by name of the plant species. The primary key of this table is a foreign key in the idtable.
spaceindex	Contains the various classes a site is divided into based on the structural organization of a site. Every digital object in the current ETANA-DL prototype has a spatial class associated with it. The primary key of this table is a foreign key in the idtable.
periodindex	Contains the various time period classes. The primary key of this table is a foreign key in the idtable.

etana

The *etana* database contains many tables that are used by various services. All services that are not implemented as components store their information in this database, with the exception of the browsing service. Table 6 provides an overview of the various tables in this database.

Table 6: Etana database tables - an overview

Table Name	Purpose
usertable	Contains user related information such as username and password. Used by user management services.
tmpusertable	Temporary table to aid user registration. Stores temporary user information before account is activated, including the registration key.
userobjtable	Stores the <i>Items of Interest</i> for a user. Used by the binding service.
recentsearches	Stores the ten most recent queries issued by a user. Used by the service that lists most recent queries issued by a user.
reviewobjtable	Used by the <i>Mark Items</i> service.
comparison	Used to store the attributes of digital objects that form the basis for comparison of the objects in ETANA-DL. Used by the <i>Object Comparison</i> service.
objectcomparison	Used to store the objects that have been marked by a user for comparison. Used by the <i>Object Comparison</i> service.
contexts	Used to store the browsing contexts of a user. Used by the multi-dimensional browsing service.
collmetadata	Used to store collection-level metadata for every collection in ETANA-DL.
collectioncontacts	Stores contact information about the people in-charge of the various collections in the ETANA-DL prototype.

3.4.4. ETANA-DL application

3.4.4.1. Union Catalog

The databases required for storing the data harvested from our partner sites are created when installing the components, and by running the database setup scripts. However, we need to configure the harvester to harvest data from each partner site that is an Open Archive. Appendix A outlines the steps required to install the ETANA-DL application on a different server, and contains instructions for setting up the various ODL components.

While setting up the Union Catalog component, we specify the URLs of the various Open Archives that are data providers for ETANA-DL, metadata formats supported by the archives, harvesting frequency, etc. Once the necessary configuration files have been setup, the harvester automatically harvests the data from the various Open Archives and creates the Union Catalog. The schema for the XML records corresponding to the various digital objects is the unifying ETANA-DL XML schema.

3.4.4.2. Indexing harvested data

As explained in Section 3.2, the information in the Union Catalog needs to be indexed to provide the searching and browsing services. Indexing the information for searching is done by the search engine component. During installation and configuration of the component, we provide the URL of the Union Catalog, which itself is an Open Archive. The component then uses OAI-PMH requests to fetch the records and index them. In the current version of the prototype, we had to modify the search engine component to enable indexing of numeric and alphanumeric data. This modification was necessitated by the information needs of the archaeologists, who need to pose queries to the system that contains such data.

The archaeological data also is indexed in relational databases. For this purpose, we have written custom components that read the records from the Union Catalog and index it. To index a record, the component assumes that the classification information is stored in the database. For each dimension to be indexed, the component determines the correct class from the appropriate classification table, and assigns the identifier of the class to the index table. The classifications in the ETANA-DL prototype assume the existence of a unifying ontology that accommodates information harvested from all three sites. We have written programs that, given an ontology as a structured XML file, can create the classification tables in the database.

3.4.4.3. Interfacing with ODL components

The ODL components used in ETANA-DL, as explained in Section 2.4, are invoked by the Web server. The parameters are encoded as GET and POST requests. To invoke a component, the application opens a socket with the Web server. The application then creates well-formed HTTP requests that are sent over the socket established with the Web

server. The XML responses are sent back by the Web server over the same socket, which are parsed by the application to get the necessary information.

Opening a socket with the Web server requires extra overhead that might slow the application considerably. In cases where the application needs to send multiple requests to the Web server to provide a service, we use connection pooling, wherein the same socket connection is used for sending multiple requests.

3.4.4.4. Designing an extensible and portable application

The user interface components of the prototype have been developed with the goal of making the application easily extensible and portable. Figure 13 provides a first look at the ETANA-DL prototype¹ home page.

Figure 13: ETANA-DL prototype home page

Every page rendered by the application has a header and a footer section, separated by the contents of the page. The header contains the title, a search box, and links to various services. The footer contains links: to the ETANA-DL home page, to Web pages of partner universities, and to the National Science Foundation. We have created generic

¹ ETANA-DL prototype home page: <http://feathers.dlib.vt.edu>

functions for generating the header and footer sections, and these functions render the header and footer based on the context of a user. Figure 14 illustrates this design.

Figure 14: Content rendering in ETANA-DL – design overview

Because ETANA-DL is a web-based system, its information and services are linked together by hyper-links. We have not hard-coded any hyper-links in the application. Certain configuration files contain all the link information, and modifying these files has a system-wide effect, as all other components of the system use the information in the configuration files. Thus, the application can be easily ported to other servers, making setting up mirrors for the valuable archaeological data an easy task. We also use these configuration files for setting parameters such as number of records displayed on a page, display properties, etc., and this will make personalizing the application easier in future versions of the system.

We expect much richer content to be supported by future versions of the system, with several kinds of digital objects from various collections, and with many attributes. The users of the system would expect multiple views of the same data to be generated by the system. As an example, digital objects like seeds are stored in containers. Researchers working on the Nimrin project know containers that contain digital objects as *Bags* whereas those working on the Umayri project know them as *Pails*. To handle such scenarios, we have created display functions that can intelligently generate different views of the same data. Adding newer kinds of data into the system would only require creating new functions that render the data.

4. ETANA-DL Services

This chapter describes the various services that the ETANA-DL prototype supports. In describing each service, we first state the motivation for having the service in the prototype followed by its description. Wherever necessary, we will explain how the service fits into the overall architecture of the system. Table 7 provides an overview of the various ETANA-DL prototype services and their classification [37].

Table 7: Overview of ETANA-DL services

Service	Description
Information satisfaction services	
Searching	Allows users to search for specific information in the DL. Users can use the advanced search option to formulate queries that are more complex.
Browsing	Allows users to discover new information in the DL. Users can browse along many dimensions, and categories for browsing are generated dynamically based on the DL content. Users also can search within a context, save browsing contexts, etc.
Recommendation	Recommends digital objects in the DL that the user is not aware of, based on similarity of interest with other users.
Domain-specific services	
Object Comparison	Allows users to perform comparisons between different digital objects and view the results. Users specify the various parameters that form the basis for comparison.
Marking items	Workflow for allowing users to direct specific digital objects to other users of the system. Users can include annotations that are only visible to specific other users.
Value-added services	
Annotations	Allows users to discuss the various digital objects in the system. Users can post messages, and other users can respond to these posted messages.
Recent searches/discussions	Allows users to view their most recent searches and recent on-going discussions.
Items of Interest	Binding service that allows users to create personal collections out of items in the DL that interest them.
Miscellaneous services	
User management	User registration, system login, and other user management functions.
Collections Description	Allows users to view detailed information about the various collections in ETANA-DL.

4.1. Information satisfaction services

This class of DL services tries to satisfy the information needs of end-users. Typically, every DL provides two basic services to seek information from the system: searching and browsing. These services are examples of the “PULL” model of retrieval, wherein the user tries to *pull* information from the system. Alternatively, there are “PUSH” models of retrieval, wherein the system tries to *push* information towards the user. An example is a recommender system that generates recommendations after learning the information needs of a user.

4.1.1. Searching

4.1.1.1. Motivation

Archaeologists need to be able to locate information in the DL accurately and in an easy-to-use manner. Moreover, their information needs are very complex. Archaeological research depends on cross-site comparisons of objects, and having the capability to issue powerful queries that can search across multiple collections housed in the DL would be a very useful feature. In a DL such as ETANA-DL, where there are multiple kinds of digital objects from multiple collections, searching for relevant information is difficult. The users might only be interested in specific collections or specific types of digital objects. Being able to constrain the search space is very important. The data that we are dealing with is highly structured in nature. Thus, being able to issue queries that take advantage of the structured nature of the data would improve the utility of this service.

4.1.1.2. Implementation

ETANA-DL prototype uses the IRDB search engine component as the backbone to provide the searching service [38]. This component can index information in any metadata format, and hence can be used with the custom XML schema that ETANA-DL uses. The component uses a custom query language that is used to specify the query to the search engine. The queries that are specified can be structured, and the results returned by the search engine are ranked based on the structure of the data.

Figure 14 illustrates the design of the searching sub-system as implemented in the prototype. To issue a query, users can type the query in one of the search boxes as indicated in Figure 16. The query either can be a simple string like “Metatarsal or Pig Bone” or highly structured like “+BoneName:Metatarsal +AnimalName:Pig +Site:Lahav”. In the latter case, the query language as specified by the ODL-IRDB search engine component is used [18]. The servlet that handles the search functionality interprets the query and encodes it (if necessary) in the ODL-IRDB search engine query language. The search engine component returns the relevant records encoded in XML, which are then translated into HTML by the servlet. The results are finally presented to the user.

Figure 15: ETANA-DL searching sub-system overview

The search results are presented to the user in a page-by-page fashion, as shown in Figure 17. Users can easily navigate from one page to another, or to the results that are most relevant, and hence at the top of the result set, or to the ones those are least relevant, and hence are at the end of the result set. One major issue in displaying the data is to choose between the local view of the data and its global view as known to the system. We choose to convert the data back to its local view. (As an example, the names of the fields change from the global schema to the name as it known in a local site.) We also have created generic transformation functions for each kind of object in the system. These functions are designed to be intelligent, to perform the mappings from the global to the local view.

For issuing complex, structured queries, ETANA-DL presents an advanced search option (see Figure 18). Because ETANA-DL has several kinds of digital objects and several sites, the user initially has to select a starting point for issuing the query. There can be several different starting points, but the prototype currently only supports the type of object as the starting point. Upon selection of the object type, the user is presented with search fields for the selected object type. The user can add or delete search fields as necessary. To create the query, the user has to select a field and enter the value to be searched in that field. When the user clicks the search button, the selections are submitted to the controller servlet, which then creates the structured query in the query language understood by the ODL-IRDB search engine component.

ETANA-DL
Managing complex information applications:
An archaeology digital library

[Login](#) | Search ETANA-DL for

[Browse](#) | [Advanced Search](#) | [Help](#)

Search ETANA-DL for

[Login](#) | [Create User Account](#)

Want to know more about our collections? Click [here](#)

Your comments, suggestions and feedback are important to us! Please click [here](#) for the same.

Welcome to ETANA-DL

This research proposes to develop a digital library (DL) for ancient Near Eastern studies with two archaeological components: DigBase (DB) - a repository and an archive for archaeological data from the Near East and beyond, and DigKit (DK) - a compatible field tool for collecting and recording archaeological data during archaeological surveys and excavations. DB is a model-based, extensible, archaeological componentized DL that will manage complex archaeological information sources based on the client-server paradigm of the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH). DK is a compatible field tool for collecting, recording and exposing archaeological data in an OAI compliant manner during archaeological surveys and excavations.

If this is the first time you are visiting the Digital Library, you might want to go thru the [short tutorial](#).

Recent Discussions

3/6/04 - [Re: test](#) by [unni@vt.edu](#)
2/18/04 - [Re: test](#) by [unni@vt.edu](#)
2/4/04 - [test](#) by [unni@vt.edu](#)

[Home](#) | [About ETANA](#) | [CWRU](#) | [Vanderbilt](#) | [VT](#) | [Contact us](#)

This project is [funded](#) in part by [National Science Foundation's ITR](#) program

Figure 16: Searching ETANA-DL – search interface

Total number of hits for **metatarsal bone** : 9541
Showing 1-10 out of 9541 records

1 2 3 4 5 6 > >>

Nimrin	Bone ID 1392	QUAD NW	NS/EW N35/W20	LOCUS TRIM	BAG 171	PIECES 1
AGES - AGE -						
☐	BONE METATARSAL	ANIMAL	MEDIUM MAMMAL			
COMMENTS UNIDENTIFIED						
[View complete record] [Add item to Items of Interest] [Mark Item]						
Nimrin	Bone ID 7540	QUAD NW	NS/EW N40/W20	LOCUS TRIM	BAG 121	PIECES 1
AGES - AGE 0						
☐	BONE METATARSAL	ANIMAL	SHEEP / GOAT			
COMMENTS FRAG						
[View complete record] [Add item to Items of Interest] [Mark Item]						
Nimrin	Bone ID 4220	QUAD NW	NS/EW N40/W20	LOCUS BALK	BAG 404	PIECES 1
AGES - AGE 0						
☐	BONE METATARSAL	ANIMAL	MEDIUM MAMMAL			
COMMENTS PROX BROK						
[View complete record] [Add item to Items of Interest] [Mark Item]						

Figure 17: Search results displayed in a page-by-page fashion

ETANA-DL Advanced Search Options

ption if you want to search within any specific type of object. This will result in the
ctions. An example query would be to search for all bones where the animal name is

Select Object Type

ETANA-DL Advanced Search Options

ption if you want to search within any specific type of object. This will result in the
ctions. An example query would be to search for all bones where the animal name is

Select Object Type

AND

AND

Add Search Field

Search ETANA-DL

Figure 18: Advanced searching: a) user selects type of object; b) user creates structured query on the basis of the selected type of object

4.1.2. Multi-dimensional Browsing

4.1.2.1. Motivation

While searching is a useful feature for retrieving information that exists in the system, and which the user is aware of, information discovery is made easier by systems that enable users to browse their contents. In ETANA-DL, we are dealing with several kinds of digital objects from various collections and one of our goals is to create a unifying system for the different kinds of information from those collections. Hence, we have implemented an easy-to-extend multi-dimensional browsing system in the prototype that will aid users of the DL to discover new kinds of information.

Browsing systems are based on the existence of ontologies, as explained in the next section. Though several ontologies exist for archaeological information, various projects use differing ontologies. An example is the period ontology that can be used to classify time-related data into various periods. Different archaeological sites have different ontologies for periods. Hence, creating a unifying browsing system that will allow a user to browse the data from different sites along the “period” dimension is a very difficult problem. Alternatively, DLs might use multiple schemes to classify similar information and render the information to be browsable [39].

4.1.2.2. Related concepts

In this sub-section, we discuss some of the concepts that will help in understanding the browsing service supported by the ETANA-DL prototype.

4.1.2.2.1 Ontology

Ontology specifies a conceptualization of a domain in terms of concepts, attributes and relations. Ontologies are represented as taxonomy trees, where every node represents a concept and every parent node is a generalization of all its children [40].

4.1.2.2.2 Multi-dimensional browsing

If we consider the information space to be n -dimensional, where each dimension corresponds to a way to navigate the system, a system that allows multi-dimensional browsing allows a user to move along any of the n dimensions or a combination thereof. In the ETANA-DL prototype, as illustrated by Figure 19, there are three dimensions for browsing: by site, by period, and by type of artifact. Browsing by site is based on the 5S structural model that we have partially developed. Browsing by time-period is based on the existence of an ontology for time-periods. Browsing by the type of artifact allows a user to browse by specific characteristics of individual artifacts in the system.

Figure 19: Three possible dimensions of browsing in ETANA-DL - Site, Period, and Artifact Type

4.1.2.2.3 Browsing context

In an n -dimensional browsing system, we define the browsing context of a user as the state of the system the user is in when x out of the n dimensions have values associated with them. Browsing contexts enable users to save the state of the system and return to it without having to setup values for the x dimensions again.

4.1.2.3. Implementation

4.1.2.3.1 Browsing service

ETANA-DL provides a dynamic multi-dimensional browsing system. Multi-dimensional browsing systems were explained in the previous section. Dynamic browsing systems only show those categories for a dimension for which digital objects exist. For creating such dynamic browsing systems, all fields that are to be rendered browsable need to be indexed in a database so as to dynamically determine if records, that satisfy a particular set of value constraints, exist or not.

The browsing system is not componentized in the current ETANA-DL prototype. We have already explained the process of indexing the data in the previous chapter. The design of the browsing system is based on the Model-View-Controller (MVC) software design paradigm.

Figure 20 illustrates the dynamic nature of the browsing system. When the user selects a value along the period dimension, the number of locus categories that can be browsed

decrease because all the categories that do not have any objects (in this case, bone records) with the selected period value in the “AGE” field are truncated.

Figure 20: Dynamic selection of browsable categories - all loci that contain objects that have a period value other than the one selected are eliminated

4.1.2.3.2 Searching within a browsing context

ETANA-DL will have various kinds of information from various collections. Often, users working with the system would only be interested in certain kinds of data from certain collections. We have provided a service that allows the users to search within a subset of the entire dataset in ETANA-DL. To narrow the search space, we use the information from the values set by the user in a browsing context to reformulate the user query. This query is then evaluated by the search engine that returns results from a constrained search space. The advantages of combining the searching and browsing paradigms have been shown to be important by systems like WebGlimpse [41] and HIBROWSE [42].

To illustrate this service and our approach to transforming queries to constrain the search space, consider the context shown in Figure 21. Here the user has selected Nimrin’s NW

partition, a sub-partition value of N30/W20, and Iron II as the age. Searching for “equus” would result in the query string being transformed to the following:

+equus +Site: Nimrin +Partition:NW +Subpartition:N30/W20 +AGE:IRON II”

The above query enforces the search engine to consider only those digital objects that have the corresponding values in the fields specified in the query as being relevant.

The screenshot shows the ETANA-DL search interface. At the top, there's a blue bar with "Browse ETANA-DL's collections" and a "Save this Navigation Path" button. Below this, a breadcrumb trail reads "You are in: Main > IRON II > Nimrin > NW > N30/W20" with a "View all records" button. A search bar contains the text "equus" and a "Go" button. Below the search bar, there are three yellow highlighted sections for browsing within the context:

- Browse within Sub-partition by Locus :** > [Nimrin](#) > [NW](#) > [N30/W20](#)
[13](#) , [11](#) , [37](#) , [7](#) , [16](#) , [15](#) , [14](#) , [17](#) , [26](#) , [24](#) , [33](#) , [30](#) , [32](#) , [29](#) , [34](#) , [35](#) , [36](#)
- Browse within Period :** > [IRON II](#)
[900-800 BC](#) , [800-750 BC / E 8BC](#) , [750-700 BC / L 8BC](#) , [700-500 BC](#)
- Browse by Artifact Type**
[Bone](#)

Figure 21: Searching within a context - only records within Nimrin's NW partition, in N30/W20, and dated as belonging to Iron II, will be searched

4.1.2.3.3 Saving and restoring contexts

The current ETANA-DL prototype allows users to save their browsing contexts. To save a browsing context, the user needs to select the “Save this Navigation path” link while using the browsing service after he is logged onto the system. The saved contexts are stored separately for each user. The user can return to his browsing context with a single click. Figure 22 shows several browsing contexts that have been saved for a user.

Viewing all your saved Browse contexts		
1.	<ul style="list-style-type: none"> • Period: > IRON I • Site: Nimrin • Partition: NW • Sub-partition: N40/W20 	[Go to Context]
		[Delete]
2.	<ul style="list-style-type: none"> • Artifact: Seed • Site: Nimrin • Partition: NW • Sub-partition: N20/W60 	[Go to Context]
		[Delete]
3.	<ul style="list-style-type: none"> • Period: > IRON I • Site: Nimrin • Partition: NW • Sub-partition: N35/W20 	[Go to Context]
		[Delete]

Figure 22: Viewing saved browsing contexts - the user can return to the context with a single click

4.1.3. Recommendations

4.1.3.1. Motivation

In a collaborative environment, we envision that several people on a team will be working on things of common interest. For example, bone specialists associated with site X could work on bones from site X. Alternatively, pottery specialists associated with site Y could work on bone data and images from site Y. In such a setting, it would be very useful to have a service that would make useful recommendations about digital objects similar to the ones that interest a user. Recommender systems are widely used in e-commerce applications to make recommendations to users. Recommender systems require to be trained to learn the interests of a user and make recommendations. These systems increase the value of the application because they provide new information to the user. Two interesting articles that explain recommender systems are [43, 44].

The current prototype of ETANA-DL uses a recommendation engine component to make recommendations based on users' similarity of interests in digital objects of the system.

4.1.3.2. Implementation

The recommendation service uses a recommendation engine ODL component. Recommendations are only provided to users who are logged onto the system, as recommendations are dependent on similar interests. To understand how the engine works, we will use an example scenario:

User X logs onto the system and adds items A, B, and C to his personal collection (which we call *Items of Interest* in ETANA-DL. Now, user Y logs onto the system and decides to view item B. In this case, the system would recommend items A and C to user Y because user X was interested in items A and C along with item B.

As more users use the system, the system keeps learning and the value of the recommendations made by the system keeps increasing.

To train the recommendation engine component, the inputs that need to be provided consist of a username and an identifier that uniquely identifies the digital object in the system, which in our case is the OAI ID of the object. To obtain recommendations for a digital object, we only need to provide the username and the OAI ID of the object for which recommendations are sought.

The working of the recommendation sub-system is very similar to the searching sub-system. The display controller servlet generates an XOAI request for the component and the component returns a list of recommendations, in the form of an XML tree. The display controller then selects the top five recommendations that the user is not aware of, by comparing the OAI ID of each recommended object to the identifiers of the objects in the *Items of Interest* list of the user. Figure 23 shows recommendations generated by the ETANA-DL prototype for a digital object that the user is viewing.

Figure 23: Recommendations generated by the system - user is viewing a figurine object in the system

4.2. Domain-specific services

Domain-specific services are dependent on the domain that the DL is being constructed for, and depend on the requirements users have from that particular domain. The current prototype of ETANA-DL supports two such services: Object Comparison and Marking Items.

4.2.1. Object Comparison

4.2.1.1. Motivation

One of the core services that an archaeological information system needs to provide its users is the ability to compare digital objects, and statistics generated from contents of the system. The parameters on which the comparisons are based vary, and are dependent on the complexity of the query, type of digital object and its collection, etc. Hence, the service needs to be highly flexible and configurable. Moreover, user interface issues are important, like the visualization of comparison results, ability to add and delete parameters that form the basis for comparisons, etc.

4.2.1.2. Implementation

The current ETANA-DL prototype allows archaeologists to make comparisons at the level of individual digital objects. We have developed a workflow for accomplishing this task that involves multiple steps.

The first step is to collect various digital objects (examples include bones, seeds, and figurines in the current prototype) that interest individual users of the system, and which they would like to compare with other objects of similar kinds. We maintain collections of different kinds of objects for every user registered with the system. The user can add or delete objects from this collection by using the interface provided. To add objects, the users only need to select the appropriate objects from the search or browse results. Figure 24 highlights the process of adding objects, for making comparisons at a later stage.

☒	Nimrin Bone ID 1406	QUAD NW	NS/EW N35/W20	LOCUS 63	BAG 166	PIECES 1
	AGES IRON II	AGE 800-700 BC				
	☒ BONE METATARSAL	ANIMAL CAPRA				
	COMMENTS -					
	[View complete record]	[Add item to Items of Interest]	[Mark Item]			
<hr/>						
☒	Nimrin Bone ID 897	QUAD NW	NS/EW N35/W20	LOCUS 50	BAG 68	PIECES 1
	AGES IRON II	AGE 800-700 BC				
	☒ BONE METATARSAL	ANIMAL LEPUS				
	COMMENTS -					
	[View complete record]	[Add item to Items of Interest]	[Mark Item]			
<hr/>						
☐	Nimrin Bone ID 7498	QUAD NW	NS/EW N40/W20	LOCUS 263	BAG 142	PIECES 1
	AGES MIDDLE BRONZE	AGE 1700-1500 BC				
	☐ BONE METATARSAL	ANIMAL BOS				
	COMMENTS -					
	[View complete record]	[Add item to Items of Interest]	[Mark Item]			
<hr/>						
1 2 3 4 5 6 > >>						
<div>Save checked objects for comparison</div>						

Figure 24: Adding items in ETANA-DL for comparison

The second step involves setting up the attributes that form the basis for comparisons. The digital objects in the ETANA-DL prototype are complex, and each object has several kinds of information associated with it. We provide the users with the ability to select the fields that interest them. The users also have the ability to update their selections. Figure 25 shows the interface for setting up the attributes. To edit the list of selected attributes, the users have to select the “Edit” link corresponding to the object type. The interface for editing attributes is shown in Figure 26.

The following table lists the available attributes for the various types of objects in ETANA-DL and the ones that you have selected for comparison. You can edit the lists by clicking the **edit link** of the corresponding object type.

Object Type	Available Attributes	Selected Attributes	
Bone	ID, COLLECTION, OWNERID, PARTITION, SUBPARTITION, LOCUS, CONTAINER, RANK, AGES, AGE, PAGE, LINE, PIECES, BONENAME, SIDE, ANIMAL, COMMENTS, DATE_DUG, CDATE, UDATE, ONE, TOTS, SQRS	ID, COLLECTION, PARTITION, SUBPARTITION, LOCUS, CONTAINER, PIECES, BONENAME, SIDE, ANIMAL, AGE	[Edit]
Figurine	ID, COLLECTION, OWNERID, PARTITION, SUBPARTITION, LOCUS, CONTAINER, SHORDESCRIPTION, LONGDESCRIPTION, CONDITION, TECHNIQUE, DIMENSIONS, WAREPASTE, COMPARISON, DIG, IMAGEURL	--None Selected--	[Edit]
Seed	ID, COLLECTION, OWNERID, PARTITION, SUBPARTITION, LOCUS, CONTAINER, NAME, SPECIES, QUANTITY, COMMENTS, DATE_DUG, CDATE, UDATE	ID, COLLECTION, PARTITION, SUBPARTITION, LOCUS, CONTAINER, NAME, QUANTITY	[Edit]

Figure 25: Selecting available attributes for different kinds of objects for comparison

Add/Remove comparison attributes for Bone

OWNERID

RANK

AGES

PAGE

LINE

COMMENTS

DATE_DUG

CDATE

>>

<<

ID

COLLECTION

PARTITION

SUBPARTITION

LOCUS

CONTAINER

PIECES

BONENAME

List of available attributes

List of selected attributes

Save Changes

Figure 26: Editing attributes for object type "Bone"

The third step is the actual comparison step. The users have to select objects out of the objects that were marked for comparison in the first step. The comparison results are shown in a tabular format, as shown in Figure 27. Selection of a different kind of object or different objects of the same kind would generate different results.

Comparison Results for selected Bones					
ID	COLLECTION	PARTITION	PIECES	BONENAME	ANIMAL
1006	Nimrin	NW	1	HUMERUS	SHEEP / GOAT
1007	Nimrin	NW	1	METATARSAL	SHEEP / GOAT
1005	Nimrin	NW	3	VERTEBRA	MEDIUM MAMMAL
1024	Umayri	H	1	-	Mammal
1025	Umayri	H	20	-	Sheep/Goat

Choose different Bones

Figure 27: Comparison result for selected bones – bone records are from different collections housed in ETANA-DL

4.2.2. Marking items for analysis and quality control

4.2.2.1. Motivation

ETANA-DL is envisioned to be a system that manages all kinds of archaeological data, including records that are generated in the field. Often, there are data entry errors in these field records. The DL needs to support workflows that will allow users to point out specific objects to other users, in this case the user in charge of the incorrect records, so that corrections can be made at the source of the records. In other cases, a user might simply want another user to view specific digital objects in the system, and might want to make a note or an observation that is only visible to the intended user.

These requirements necessitate the availability of a service that makes directed sharing of information possible. By directed sharing, we mean that users should be able to share information with specific other users and the service should provide an easy way for a user to view all the information directed towards him. The *Marking Items* service provides the basis for building such user-oriented services in the DL.

4.2.2.2. Implementation

The *Marking Items* service, in the current ETANA-DL prototype, is not componentized. We use the Model-View-Controller (MVC) software design paradigm for implementing the service with the model being stored in the database. The servlets that provide the functionality have to query a database for the necessary information.

To mark an item, a user has to choose the *Mark Item* link associated with objects shown in the search or browse results, or with the detailed display of an object. Figure 28 shows

the interface for marking an item. The user can choose to mark an item to its owner (e.g., the director of a dig for field records) or to another user of the system. If the user chooses to mark the item to another user, a username is required. The user also can enter a detailed comment indicating why the object is being marked.

The screenshot shows a web interface for marking an item. At the top, there is a dialog box titled "Mark item to". It contains two radio buttons: "Owner (jwf2@case.edu)" and "Another User". The "Another User" option is selected, and a text input field next to it contains "rsmith@email.com". Below this, there is a "Comments" label and a text area containing the text: "Look at this bone record. I looks an interesting deviation from the others in its area." At the bottom of the dialog box is a "Mark Item" button.

Below the dialog box, there is a section titled "Viewing Detailed Record for Bone with ID number 1060". It contains a table with the following data:

ID	1060
PARTITION	NW
SUBPARTITION	N20/W65

Figure 28: Marking items - marking an item allows the user to whom the item has been marked to view comments associated with the marking

Once the user, to whom the item has been marked, logs onto the system, the system shows a message indicating there are new items that have been marked to him. Figure 29 shows a listing of items marked to a user. The user can choose to view the details of the digital object marked to him, add the item to his *Items of Interest*, re-mark the item to another user, or unmark the item. Unmarking the item removes the item from the list of marked items.

Items marked to Robert Smith			
From	Date	Identifier	Comments
rshen@vt.edu	3/19/04	oai:etanafigurineslahav:1852	Comments sought on type of soil for the locus where this figurine was found...
[View Complete Record] [Add item to Items of Interest] [Re-mark Item] [Unmark Item]			
unni@vt.edu	3/19/04	oai:etanabonesnimrin:1060	Look at this bone record. I looks an interesting deviation from the others in its area.
[View Complete Record] [Add item to Items of Interest] [Re-mark Item] [Unmark Item]			
Home About ETANA CWRU Vanderbilt VT Contact us			

Figure 29: Viewing marked item - user can choose to view the complete record of the corresponding digital object, add the object to Items of Interest, re-mark the item, or unmark it

Re-marking the item can be used to mark the item back to the original source or other users of the system. The user interface for remarking items is shown in Figure 30. Re-marking the item removes the item from the list of marked items. A comment can be added to items being re-marked.

Viewing Detailed Record for Bone with ID number 1060	
ID	1060
PARTITION	NW
SUBPARTITION	N20/W65

Figure 30: Re-marking an item - user can choose to remark the item to the owner of the digital object, the original source of the marking, or to a different user

4.3. Value-added services

The current ETANA-DL prototype supports many services that add value to the system. Some of these are annotations, *Items of Interest*, recent searches/discussions, etc. We shall describe these services in this sub-section.

4.3.1. Annotations

4.3.1.1. Motivation

ETANA-DL is envisioned to provide a collaborative environment for the archaeological community. One of the steps in that direction is to provide a means for discussing various digital objects in the DL, wherein users can post their comments and views about items, point out mistakes in digital records, etc. Other authorized users of the system should be able to view the comments and post replies to them. This necessitates having a multi-threaded discussion board associated with every digital object in the system.

4.3.1.2. Implementation

ETANA-DL uses a component that encapsulates the annotation engine to provide the multi-threaded annotation service [38]. The component exposes a very simple interface to add annotations and retrieve them. The way the annotation component works is similar to the search engine component. Every annotation in the annotation database has a unique Open Archives Initiative ID (OAI ID). Given the identifier of the object for which the annotations are to be retrieved, the component will retrieve a tree view (encoded in XML) of the corresponding annotations. The servlet that handles the annotations then parses the XML to create the multi-threaded view. To retrieve the contents of any message, the component requires an input of the OAI ID of the annotation. To insert an annotation, the component uses the PutRecord XOAI verb, and only requires the OAI ID of the parent node (we give the identifier of the digital object for root nodes) and the message encoded in XML. Refer to [18] for more information on the annotation engine component protocols.

The multi-threaded discussion boards associated with each object can be viewed with the detailed description of that object. Figure 31 illustrates a multi-threaded discussion associated with an object. Figure 32 shows the contents of a message and the interface to post replies to messages. Detailed description of the service and the various related available features can be obtained by going thru the tutorial available at [45].

TOTS	Information unavailable in database
SQRS	Information unavailable in database

[\[Back to Search Results\]](#)

Discussions about this item
[\[View Threads\]](#) | [\[New Thread\]](#)

[Message Index]

- >> [Information unavailable?](#) - unni@vt.edu - Thu Mar 18 18:41:59 EST 2004
- >> [Re: Information unavailable?](#) - tomh@quickmail.com - Thu Mar 18 18:47:20 EST 2004
- >> [Re: Re: Information unavailable?](#) - rsmith@email.com - Thu Mar 18 18:48:36 EST 2004
- >> [Re: Re: Re: Information unavailable?](#) - unni@vt.edu - Thu Mar 18 18:49:24 EST 2004
- >> [Re: Information unavailable?](#) - rshen@vt.edu - Thu Mar 18 18:42:37 EST 2004
- >> [Re: Re: Information unavailable?](#) - unni@vt.edu - Thu Mar 18 18:43:04 EST 2004

[Post Message/Replies]

Subject

Message

Figure 31: Viewing the multi-threaded discussion board for an object – authorized users can post new messages/replies to messages

50

Discussions about this item
[\[View Threads\]](#) | [\[New Thread\]](#)

Subject: Re: Re: Information unavailable?
 Message posted by rsmith@email.com on Thu Mar 18 18:48:36 EST 2004

The problem is now fixed. Please check....It was due to an error in parsing the data, so the data was being exposed incorrectly.

[Post Message/Replies]

Subject

Re: Re: Re: Information unavailable?

Message

Thanks! I will have to harvest the collection again, only then will be the changed reflected in ETANA-DL.

Figure 32: Viewing message details and posting replies to messages in a discussion board

4.3.2. Items of Interest (Binding service)

4.3.2.1. Motivation

Many DLs have binding services that allow their users to create personal collections of items in the DL. Such collections are then used for quick reference of digital objects in the future. Possible extensions to this concept include creating lesson plans that reference the objects for educational purposes [25] and sharing these collections with other users of the system [46].

4.3.2.2. Implementation

ETANA-DL supports a binding service that we call *Items of Interest* in the current DL prototype. *Items of Interest* can be used to hold personal collections for each user, and the user can then easily look up the items that are displayed in a page-by-page fashion. In the current prototype, this service is not componentized and has been implemented using the Model-View-Controller (MVC) software design pattern.

Figure 33 shows a listing of some items that a user is interested in and has added them to his *Items of Interest*. The presentation of each item in the list includes an option to view its complete digital record, mark the item to another user of the system, or remove the item from the list.

Unni Ravindranathan's Items of Interest

Total number of items in your List: 2
Showing 1-2 out of 2 records

1

Lahav Figurine ID 2442
Rider from Horse and Rider

☐ [Field IV](#) [Area K7](#) [Locus K7006](#) [Basket IV.K7.48](#)

Dimensions 8.6 cm. h; 5.6 cm. w; 3.1 cm. d

Dig Israel Antiquities Authority Accession No. 1998-776.

[\[View complete record\]](#) [\[Remove item from Items of Interest\]](#) [\[Mark Item\]](#)

Lahav Figurine ID 1850
Horse and Rider on Pedestal

☐ [Field IV](#) [Area I8](#) [Locus I8003](#) [Basket IV.I8.4](#)

Dimensions 20.2 cm. h.; 11.5 cm. w; 4.2 cm. d

Dig May be Hellenistic? See also Tel Michal, page 363, no. 7 and fig. 32.1, no. 7 in Ze'ev Herzog et al. Excavations at Tel Michal, Israel. Minneapolis: University of Minnesota Press and Tel Aviv: Nadler Institute of Archaeology, 1989. Israel Antiquities Authority Accession No. 1998-978.

[\[View complete record\]](#) [\[Remove item from Items of Interest\]](#) [\[Mark Item\]](#)

Figure 33: Items of interest for a user - the user can view the complete digital record, mark the item, or remove the item from the list

4.3.3. Recent searches/discussions

4.3.3.1. Motivation

Often, users are interested in re-submitting previously submitted queries. Having a service that allows users to re-submit previously submitted queries using one click would be very useful because some of the queries are complex to formulate. Users would benefit from being aware of the most recent discussions as well. Providing such services would increase the utility of a DL.

4.3.3.2. Implementation

The ETANA-DL prototype includes a service that allows users to view their ten most recent search queries as well as the five most recent discussions.

Every time a user submits a query to the system, a thread is launched that records the query in a database. Figure 34 shows the interface for viewing the recent searches. Clicking on the recent discussions tab will display the five most recent discussions. The recent discussions are retrieved from the annotation engine component using the XOAI interface. The searches and the discussions allow the user to launch the query or visit the multi-threaded discussions page associated with the item being discussed with a single click respectively. Recent searches are only recorded and displayed for users who are logged onto the system.

Figure 34: Recent searches and discussions interface

4.4. Miscellaneous services

4.4.1. User management services

Many services in the DL require information about the user. The user needs to create an account with the system, and logon to the system to utilize these services. Figure 35 shows the interface presented to the user trying to register with the system. When registering with the system, the system validates the existence of the e-mail address (which also serves as the username) by sending e-mail to the specified address containing a link to activate the account. For security purposes, the system also generates a unique key required to activate the account, which is sent along with the e-mail. Upon activation of the account, the user can logon to the system using the interface show in Figure 36.

The screenshot shows a form titled 'Create new ETANA-DL user account'. It has four input fields: 'First Name' with the value 'Unni', 'Last Name' with the value 'Ravindranathan', 'E-mail address (Username)' with the value 'unni@cs.vt.edu', and 'Password' with a masked value '*****'. Below the fields is a 'Create Account' button.

Figure 35: New user account creation

Login into ETANA-DL

Username (E-mail)

Password

☐ Login automatically

Forgot your password? Click [here](#)

Do not have an account yet?
You can easily create one by clicking [here](#)!

Figure 36: User login prompt

4.4.2. Collections description

To provide more information to users unfamiliar with the collections that are a part of ETANA-DL, we have created a service for providing information about these individual collections, the interface of which is shown in Figure 37. Each collection has an associated “More Information” link that provides detailed information about that collection.

ETANA-DL's Collections

Lahav Research Project
Persian era ceramic figurines
Project URL <http://www.cobb.msstate.edu/dig/lahav> [\[More Information\]](#)

Virtual Nimrin
Primary records and secondary studies on Tell Nimrin, Jordan.
Project URL <http://www.cwru.edu/afil/nimrin/index.htm> [\[More Information\]](#)

Madaba Plains Project--` Umayri
Tall al-`Umayri is a Bronze and Iron Age site near Amman, Jordan on the frontier between the sown and the desert. The project examines intensification and abatement of sedentary/nomadic lifestyles and living strategies in a 5 km radius of the site. The present database collection is only from the central site. Its main periods of occupation include the Early Bronze I-IV, Middle Bronze II, Late Bronze II, Iron I and II, and Persian period. Smaller agricultural settlements were built on small parts of the site during Hellenistic, Early Roman, and Byzantine times.
Project URL <http://www.wvc.edu/mpp> [\[More Information\]](#)

Figure 37: Description of various collections in ETANA-DL

5. ANALYSIS

In this chapter, an analysis of our approach to creating Digital Libraries that handle heterogeneous data sources is presented, aimed to show how the validity of the hypothesis stated in Section 1.3 has been demonstrated. Section 5.1 analyzes the heterogeneity of our approach in terms of the different kinds of digital objects in ETANA-DL, and the steps for converting an archaeological site into an Open Archive, the metadata of which can be harvested by the current prototype. Section 5.2 presents the analysis of our prototyping approach. Two software metrics that support our approach to prototyping information system designed to handle heterogeneous data are presented. In Section 5.3, we provide a summary of observations made by members of the archaeological community who evaluated our prototype.

5.1. Heterogeneous data handling

The current ETANA-DL prototype deals with three different sites – Nimrin, Lahav, and Umayri. We have only partially converted these sites into Open Archives, but have harvested records of different kinds of digital objects from these sites to prove the extensibility and scalability of our approach to handling heterogeneous archaeological data. Table 8 provides an overview of the data contained in the ETANA-DL Union Catalog.

Table 8: Heterogeneous data in ETANA-DL - an overview

Site	Artifact Type	Original data source	Number of attributes in original record	Number of attributes in harvested record	Number of records harvested
Lahav	Figurine	Tab-delimited text file	15	18	564
Nimrin	Bone field record	Table in relational DB	21	24	7420
	Seed field record	Table in relational DB	12	15	430
Umayri	Bone field record	2 tables in relational DB	8	24	2123

In the current DL prototype, harvested data consists of three kinds of digital objects – figurine images, bone field records, and seed field records. We have harvested bone records from two sites (Nimrin and Umayri) to show the heterogeneity of our approach (being able to handle data from disparate sources). As shown in Table 8, more than 10,000 digital records have been harvested from the three sites. The table shows the

original formats that are used to store the records at each site, the number of attributes in the original records, and attributes in the records corresponding to the object type in the global ETANA-DL XML schema. The increase in the number of attributes for each object type in its global view is due to attributes associated with information about the collection, object type, etc. being added to the metadata associated with each record. By extending the current XML schema, and by writing new functions or modifying existing functions that render the XML records in the application, newer and richer information can be ingested easily into the current system.

Section 3.4.2 listed the steps for converting a site into an Open Archive. Table 9 shows the approximate times taken during the various stages of the conversion process for the three sites. The times taken for the data analysis and mapping stages are measured in numbers of days, while the times taken for the implementation of the data and service providers are in numbers of hours. Regarding data provider implementation time, we count the time required for installing and configuring the data provider components, as well as the time for testing the newly created Open Archive. In service provider implementation time, we count the time required to test an Open Archive and configure the harvester to harvest data from that archive.

Table 9: Converting partner archaeology sites into Open Archives

Site	Data Analysis (in days)	Data Mapping (in days)	Data Provider Implementation (in hours)	Service Provider Implementation (in hours)
Lahav	2	6	4	1
Nimrin	2	2	4	1
Umayri	1	2	4	1
Total	5	10	12	3

Figure 38: Breakdown of times required for various stages in converting an archaeology site into an Open Archive for the current prototype

Figure 38 highlights the fact that maximum efforts needed to convert an archaeology site into an Open Archive are required in the data analysis and mapping stages. Data mapping in Lahav was difficult because of the presence of special characters in the records that needed to be encoded before being exposed as XML records, and consequently required more time. Table 9 shows that implementing data providers is made extremely easy by the availability of XML file-based data provider components that can be easily configured. However, implementing data providers that read directly from the data sources would require more effort.

5.2. Rapid prototyping

In this section, we argue for the rapidity of our approach to creating DLs that can handle heterogeneous data from disparate sources and provide several useful services. We use two common software metrics to analyze our approach: lines of code, and times for the development of various services.

Lines of Code for implementing services

Table 10 shows the lines of code (LOC) needed to implement several services. We compare the various services, some of which are componentized, based on LOC needed for implementing the service and LOC that are reused from components. The final column in the table is the percentage savings in LOC gained by component re-use.

Table 10: Analysis of prototype using the metric of Lines of Code

Service	Component Based	LOC for implementing service	LOC reused from component	Total LOC	Reuse Percentage
Searching – Back-end	Yes	100	1650	1750	94
Recommendation	Yes	200	700	900	78
Annotations – Back-end	Yes	50	600	650	92
Union Catalog	Yes	-	680	680	100
Searching – Front-end/Interface	No	1600	-	1600	0
Browsing	No	1390	-	1390	0
Object Comparison	No	650	-	650	0
Marking Items	No	550	-	550	0
Items of Interest	No	480	-	480	0
Annotations – Front-end/Interface	No	200	-	200	0
Recent Searches/Discussions	No	230	-	230	0
Collections Description	No	250	-	250	0
User Management	No	600	-	600	0

Framework Code	No	2000	-	2000	0
Total		8280	3630	11910	30

For the component-based services in Table 10, other than the LOC reused from the components, our application needs to interface with the components, requiring additional code lines. However, the additional code is only a very small percentage of the total lines of code required for implementing the service. In addition, we have broken down some of the services into back-end (which involves the component and the code needed to interface with it) and front-end (which handles generating complex user interfaces) components.

In the current ETANA-DL prototype, only a few important services are componentized. However, Figure 39 make it clear that we can re-use a very significant percentage (approx. 30%) of DL code by designing common DL services as components. Moreover, for creating prototypes rapidly where quality of the service is as important as the speed with which services can be put together and modified, the approach to building DLs using pre-existing components is very useful. The components that we have developed for the prototype can be re-used in other DLs, thus resulting in an even higher re-use percentage value for future extensions of our work.

Figure 39: Lines of Code for implementing service vs. Lines of Code reused from components

Service development times

Table 11 lists the times required for development of various services. In developing the services, there is time required for analyzing the requirements that users of the DL will have out of the service, designing the service and the related user interfaces, implementation, and testing. In most cases, many of these phases of the software lifecycle are overlapping because we use rapid prototyping for requirements elicitation, and hence times are harder to distinguish. As observed in the development times for the various services, the availability of services as ODL components with well-designed interfaces is

very useful for creating prototypes rapidly. The services can then be modified to incorporate requirements that users of the DL have for the system.

Table 11: Analysis of prototype using the metric of time for service development (in days)

Service	Component Based	Requirements Analysis and Design	Implementation	Testing	Total
Searching – Back-end	Yes	1	1	0.5	2.5
Recommendation	Yes	0.5	2	0.5	3
Annotations – Back-end	Yes	1	1	0.5	2.5
Union Catalog	Yes	2	1	2	5
Searching – Front-end/Interface	No	5	10	2	17
Browsing	No	8	20	3	31
Object Comparison	No	3	5	1	9
Marking Items	No	1	3	1	5
Items of Interest	No	1	1	1	3
Annotations – Front-end/Interface	No	3	7	2	12
Recent Searches/Discussions	No	2	4	1	7
Collections Description	No	2	1	1	4
User Management	No	2	6	2	10
Total		31.5	62	17.5	111

For certain services (e.g., searching and browsing), we had to develop multiple versions of the service to reflect changes in user requirements. For example, the first version of the browsing service allowed users to browse the data space only along one dimension (by spatial organization of a site). The current version of the browsing system allows users to browse along three dimensions. Table 11 only reflects the development times for the most recent version of the service.

Figure 40 shows a comparison of efforts required during the various stages of the prototyping cycle. Chart A, which gives the percentages for all services in ETANA-DL, shows that maximum amount of time is used in implementing services for the current prototype. Chart B shows the percentages of times required for the component-based services, and Chart C shows the same for non-componentized services. Comparison of Charts B and C clarify that more efforts can be spent on analysis, design, and testing for component-based services as compared to non-componentized services. Thus, a DL implementer can save a significant percentage of implementation time by using components that implement common DL services.

Figure 40: Percentage of efforts required for different stages of the prototyping life cycle:
A) all services, B) component-based services, and C) non-componentized services

5.3. User analysis

Our prototype was evaluated by many members of the archaeology community. Their evaluations mainly focused on the following aspects of the system: practical usefulness of the services for archaeologists, usability of the service, performance, and user interface. In Table 12, we summarize analysis of the various services in the current ETANA-DL

prototype. We state the pluses and minuses for each service, as indicated by the users, who provided many comments.

Table 12: User analysis of various services in ETANA-DL

Service	Analysis
Searching	Positive feedback
	<ul style="list-style-type: none"> • Search engine returns correct results for well-formed queries. • Performance of the searching system is adequate. • Search results are presented in an easy-to-navigate fashion. • Advanced searching is helpful to formulate complex structured queries.
	Negative feedback
	<ul style="list-style-type: none"> • Searching cannot depend on user knowing a query language to structure queries. • Advanced search is difficult to use. Fields for structuring queries are not well organized. Possible values for the search fields should be generated dynamically. Removing additional search fields is not possible.
Multi-dimensional Browsing	Positive feedback
	<ul style="list-style-type: none"> • Browsing is likely to be the most useful service, as it allows users to navigate the complex data space. • Ability to navigate along multiple dimensions is very helpful, as it allows a user to constrain the data space. • Spatial organization of records is very useful. • Searching within a browsing context is very useful. • Saving and restoring contexts is useful.
	Negative feedback
	<ul style="list-style-type: none"> • Browsing should be able to show records in a class grouped by some attribute (e.g., object type, locus), rather than as a long list. • Browsing should support richer ontologies.
Recommendations	Positive feedback
	<ul style="list-style-type: none"> • Recommending related digital artifacts would be a useful service.

	Negative feedback
	None
Object Comparison	Positive feedback
	<ul style="list-style-type: none"> • Selecting attributes for generating comparisons is very useful. • Organization of selected objects by type of object is useful. • Service is easy to use.
	Negative feedback
	<ul style="list-style-type: none"> • The service can be very useful if extended to allow comparisons at various levels, as compared to the object level comparisons it currently provides. • Layout of comparisons results can be more flexible. • Presentation of attributes during the comparisons setup stage needs to be re-designed.
Marking Items	Positive feedback
	<ul style="list-style-type: none"> • Service will make collaborations between archaeologists possible.
	Negative feedback
	None
Items of Interest (Binding Service)	Positive feedback
	<ul style="list-style-type: none"> • Binding is a useful service that will allow archaeologists to keep track of important digital objects.
	Negative feedback
	<ul style="list-style-type: none"> • Ability to share lists with other users, e-mail or print the lists, etc. would be useful. • User should be able to remove or mark multiple objects with a single click.
Annotations (Discussion Board)	Positive feedback
	<ul style="list-style-type: none"> • Associating discussion boards with every digital artifact is useful. Service will be extremely important in the final version of the system. • Service is easy to use.

	Negative feedback
	None
Recent Searches/Discussions	Positive feedback
	<ul style="list-style-type: none"> Both services are very useful.
	Negative feedback
	<ul style="list-style-type: none"> Service can be extended to the keep track of all searches of a user, with the ability to add and delete searches from the list. Recent discussions should have the option of showing most recent discussions in the system AND most recent discussions that involve the user.

From the analysis provided in Table 12, we can conclude that all the services provided in the current ETANA-DL prototype are useful, and we have been able to elicit many requirements for improving the services that when addressed, will increase the utility of the system. Our approach to handling heterogeneous archaeological data is further validated by the feedback we have received regarding cross-collection searching and browsing services. Nevertheless, the current prototype has some interface related issues that will need to be addressed in future versions of the system.

6. Conclusions and Future Work

6.1. Conclusions

In this thesis, we have described our experiences with creating a prototype for an archaeological DL that can handle heterogeneous data sources, and provide useful services [47]. We have argued and presented results supporting the hypothesis stated in Section 1.3, and summarized in Table 13.

Table 13: Summary of hypotheses and sections/chapters where they are demonstrated/proven

Hypothesis	Demonstrated/Proven in
Achieving interoperability among archaeological systems to make sharing of information easy	Sections 3.2, 4.1, 4.2, and 5.1
Making primary data available without significant delay	Section 3.2.1
Providing a sustainable solution for long-term preservation of archaeological information	Section 3.2.1
Providing various useful services over the harvested data	Chapter 4 and Section 5.3
Rapid prototyping of DLs given a pool of components that implement common DL services	Chapter 4 and Section 5.2
Applying the 5S framework to better understand complex archaeological information systems	Sections 3.3, 4.1, and 4.2

The following are the main conclusions that we derive from this work:

- We have demonstrated a scalable, extensible, and componentized approach for handling heterogeneous archaeological data from disparate sources. The architecture that we suggest for ETANA-DL can be extended for creating DLs in domains outside archaeology as well.
- The suggested architecture, combined with the portability of the DL that we have constructed, makes it possible to create mirrors at various locations. These mirrors ensure replicated existence of data, and thus we have tried to provide a solution to the problem of lack of sustainable approaches to long-term preservation of valuable archaeological data. In addition, by frequently harvesting data from the data sources, we can make primary archaeological data available without significant delay.

- We have created a prototype for ETANA-DL that supports several useful services. Some of these services are specific to the archaeology community, whereas others are commonly available in a variety of DLs.
- The analysis of our approach to prototyping DLs proves that given a pool of components that implement common DL services, a DL can be rapidly constructed.
- One of our goals was to rapidly prototype services that could be demonstrated to members of the archaeology community, to gather requirements that they have from such a unifying system. We have been successful in eliciting several requirements by making them use our prototype, and future versions of the system will try to address more of the requirements that we have gathered. Many of the services in the current prototype can be re-used in future versions with little or no modifications.
- Archaeological information systems are very complex. Trying to model them using the 5S theoretical framework has helped us understand the systems in a better fashion, and we have used the resulting partially extended meta-model to implement cross-collection searching and browsing systems.

6.2. Future work

Creating next generation DL services

The current ETANA-DL prototype supports several services that have been implemented mainly as a proof-of-concept, and are working prototypes for archaeologists to clarify the requirements that they will have for the system. Many of the services can be re-used, with little modifications, in future versions of the system. Integrating richer content into the DL may require a model-driven browsing system that will enable users to browse the DLs' contents easily. The service for generating comparisons of objects may be extended to allow archaeologists to perform comparisons at different levels – object level, collection/sub-collection level, etc. Services that handle GIS information may be incorporated. Visualization systems that allow archaeologists to generate visualizations, which could be used for testing various hypotheses, could be integrated.

Integrating richer content

Standards to represent archaeological information are necessary to make interchange of information among various systems easy [48]. Several projects are working towards that goal, and ETANA-DL could benefit from their efforts [49]. For integrating richer content in ETANA-DL, we need XML schemas that are flexible and extensible. Our current schema is designed to meet both of these goals. However, integrating new kinds of data would require modifications to several components – data mapping, data provider, harvester, services, etc. Mapping information at each local site to the unifying schema is a very tough issue and approaches to automatic schema mapping may be integrated [50]. Semi-automatic data mapping components that can be easily configured may be

developed that will allow easier conversion of a site into an Open Archive. Schemes to link various kinds of related digital artifacts may be integrated into the system.

Enhancing information access

The advanced searching service may be extended to allow archaeologists to submit complex queries to the system, as well as allowing queries to be issued in multiple constrained search spaces (e.g., searching all bones and seeds from three specific sites). The current prototype uses a simple IR component for providing the searching service. This component stores the inverted files in databases, and this may result in scalability problems. Search engine components that create in-memory inverted files may be used as an alternative for faster retrieval performance [38]. Archaeological data in ETANA-DL data is highly structured in nature. Techniques to combine information and database retrieval could be employed to better satisfy user information needs [51]. Related research in automatically structuring queries may be applied as an alternative to manually structuring queries to reduce the burden on users [52]. ETANA-DL will have large image collections as a part of its content. Systems that allow us to query images based on their content can be integrated to allow archaeologists to find images similar to a query image [53].

Componentizing current DL services

We have shown that having a pool of components that provide common DL services makes it easier to build DLs. By componentizing many of the services supported by the ETANA-DL prototype, we may reuse these components in building other DLs.

(Semi-) Automatic handling of DL content

The current prototype assumes that harvested data does not change at the source. Future versions of the DL should be able to handle the data being changed at the source, and these changes should be made visible to the users. This may require modifying the harvester to update the search and browse indexes upon the data being changed at the source. Techniques such as automatically updating the harvesting frequencies based on the rate at which data changes at a source could be used to keep the harvested data in synchronization with the data at the original site.

Interface redesign

The user interface may be redesigned to make the system more intuitive to novice users. Aids in this direction include easy-to-access, context-based help, and comprehensive usability tests that will allow users to work with the various services and provide feedback on ways to develop more usable interfaces. The current prototype redirects users to appropriate pages upon invalid access of a page, incorrect parameter supplied, etc., but does not show context for the redirection. Future versions of the system could incorporate comprehensive error handling mechanisms. Integrating newer kinds of

information in ETANA-DL would require creating display components that render the information.

Integrating DL logging mechanisms

We have developed an XML log standard and related tools for logging in DLs and we could integrate these into ETANA-DL [54]. Tools that analyze the usage logs and generate statistics are available, and can be used to obtain information about usage patterns, commonly used services, frequently accessed collections, etc. Logging user actions can be useful to provide customized services to users.

REFERENCES

- [1] Ravindranathan, U., Shen, R., Goncalves, M. A., Fan, W., Fox, E. A., and Flanagan, J. W. *ETANA-DL: A Digital Library For Integrated Handling Of Heterogeneous Archaeological Data*. To be presented at Joint Conference on Digital Libraries (JCDL 2004), Tucson, AZ, June 7-11, 2004.
- [2] Open Archives Initiative, 2004. www.openarchives.org (as of April, 2004).
- [3] Goncalves, M. A., Fox, E. A., Watson, L. T., and Kipp, N. A. *Streams, Structures, Spaces, Scenarios, Societies (5S): A Formal Model for Digital Libraries*, in ACM Transactions on Information Systems (TOIS), vol. 22(2), pp. 270-312, 2004.
- [4] Ryan, N. Managing Complexity: Archaeological Information Systems Past, Present and Future, <http://www.cs.kent.ac.uk/people/staff/nsr/arch/baas.html> (as of April, 2004).
- [5] Digital Archive Network for Anthropology and World Heritage (DANA-WH), 2004. <http://www.dana-wh.net/home/> (as of April, 2004).
- [6] The Cobb Institute DigMaster Database, 1998. <http://www.cobb.msstate.edu/dig/start.html> (as of April, 2004).
- [7] Madaba Plains Project, 2004. <http://www.wvc.edu/academics/departments/theology/mpp/> (as of April, 2004).
- [8] Crane, G., Wulfman, C. E., Cerrato, L. M., Mahoney, A., Milbank, T. L., Mimno, D., Rydberg-Cox, J. A., Smith, D. A., and York, C. *Towards a cultural heritage digital library*. Presented at ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2003), Houston, TX, June, 2003.
- [9] Perseus Digital Library, 2004. <http://perseus.uchicago.edu/> (as of April, 2004).
- [10] Tell Nimrin - A virtual archaeological site, 2004. <http://www.cwru.edu/affil/nimrin/> (as of April, 2004).
- [11] Bush, V. *As We May Think*, in The Atlantic Monthly, vol. 176, pp. 101-108, July, 1945.
- [12] Digital Library Federation, 2004. www.diglib.org (as of April, 2004).
- [13] Fox, E. A. Digital Library Definitions, 1999. <http://ei.cs.vt.edu/~dlib/def.htm> (as of April, 2004).

- [14] Drabenstott, K. M. "Analytical review of the library of the future", Council on Library Resources, Washington, DC, 1994.
- [15] Goncalves, M. A., Mather, P., Wang, J., Zhou, Y., Luo, M., Richardson, R., Shen, R., Xu, L., and Fox, E. A. *Java MARIAN: From an OPAC to a Modern Digital Library System*. Presented at 9th String Processing and Information Retrieval Symposium (SPIRE 2002), Lisbon, Portugal, 2002.
- [16] MARIAN Digital Library Information System, 2004.
<http://www.dlib.vt.edu/products/marian.html> (as of April, 2004).
- [17] Fox, E. A., Eaton, J. L., McMillan, G., Kipp, N. A., Mather, P., McGonigle, T., Schweiker, W., and Devane, B. "Networked Digital Library of Theses and Dissertations: an international effort unlocking university resources", in *D-Lib Magazine*, 1997.
- [18] Suleman, H. *Open Digital Libraries*. Ph.D. Dissertation. Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA, 2002. <http://scholar.lib.vt.edu/theses/available/etd-11222002-155624/>.
- [19] Lagoze, C. and Davis, J. R. *Dienst - An Architecture for Distributed Document Libraries*, in Communications of the ACM, vol. 38(4), pp. 47, 1995.
- [20] ANSI/NISO. *Information Retrieval (Z39.50): Application Service Definition and Protocol Specification (ANSI/NISO Z39.50-1995)*. Bethesda, MD, NISO Press, 1995.
- [21] History and development of OAI-PMH, 2004.
<http://www.oaforum.org/tutorial/english/page2.htm> (as of April, 2004).
- [22] Berners-Lee, T., Fielding, R., and Frystyk, H. Hypertext Transfer Protocol - HTTP/1.0, 1996. <http://ftp.ics.uci.edu/pub/ietf/http/rfc1945.html> (as of 2004, April).
- [23] W3C. Extensible Markup Language (XML) 1.0, 2000.
<http://www.w3.org/TR/2000/REC-xml-20001006> (as of April, 2004).
- [24] Lagoze, C., Sompel, H. V. d., Nelson, M., and Warner, S. The Open Archives Initiative Protocol for Metadata Harvesting - Version 2.0, Open Archives Initiative, 2002. <http://www.openarchives.org/OAI/2.0/openarchivesprotocol.htm> (as of April, 2004).
- [25] CITIDEL (Computing and Information Technology Interactive Education Digital Library), 2004. www.citidel.org (as of April, 2004).

- [26] Fox, E. A., Knox, D., Cassel, L., Lee, J. A. N., Pérez-Quinones, M., Impagliazzo, J., and Giles, C. L. CITIDEL (Computing and Information Technology Interactive Digital Educational Library) Proposal, 2002. <http://researchsmp2.cc.vt.edu/~fox/tmp/cfgiklp.pdf> (as of April, 2004).
- [27] Goncalves, M. A. and Fox, E. A. *5SL - A Language for Declarative Specification and Generation of Digital Libraries*. Presented at Joint Conference on Digital Libraries (JCDL 2002), Portland, OR, 2002.
- [28] Zhu, Q. *5SGraph: A modeling tool for Digital Libraries*. Masters Theses. Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA, 2002. <http://scholar.lib.vt.edu/theses/available/etd-11272002-210531/>.
- [29] Kelapure, R. *Scenario-Based Generation of Digital Library Services*. Masters Theses. Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA, 2003. <http://scholar.lib.vt.edu/theses/available/etd-06182003-055012/>.
- [30] Lahav Research Project, 2004. <http://www.cobb.msstate.edu/dig/lahav/> (as of April, 2004).
- [31] Madaba Plains Project – Tall al-‘Umayri, 2004. <http://www.wvc.edu/academics/departments/theology/mpp> (as of April, 2004).
- [32] Kelapure, R., Goncalves, M. A., and Fox, E. A. *Scenario-Based Generation of Digital Library Services*. Presented at European Conference on Digital Libraries (ECDL 2003), Trondheim, Norway, August 17-22, 2003.
- [33] Red Hat Linux, 2004. <http://www.redhat.com> (as of April, 2004).
- [34] MySQL (Relational Database Management System), 2004. <http://www.mysql.com> (as of April, 2004).
- [35] The Apache Jakarta Project, 2004. <http://jakarta.apache.org> (as of April, 2004).
- [36] Java Servlet Technology, 2004. <http://java.sun.com/products/servlet/index.jsp> (as of April, 2004).
- [37] Goncalves, M. A. *Digital Libraries: formal theory, language, design, generation, quality, and evaluation*. Ph.D. Dissertation Proposal. Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA, 2003.
- [38] DL-in-a-box, 2004. <http://dlbox.nudl.org/> (as of April, 2004).

- [39] Krowne, A. and Fox, E. A. *An Architecture for Multischeming in Digital Libraries*. Presented at 6th International Conference on Asian Digital Libraries (ICADL 2003), Kuala Lumpur, Malaysia, 2003, Springer, Lecture Notes in Computer Science, pp. 563-577.
- [40] Fensel, D. *Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce*, Springer-Verlag, 2001.
- [41] Manber, U., Smith, M., and Gopal, B. *WebGlimpse - Combining Browsing and Searching*. Presented at Usenix Technical Conference, Jan 6-10, 1997.
- [42] Pollitt, A. S. "The key role of classification and indexing in view-based searching", University of Huddersfield, UK, Technical Report, 1998.
- [43] Resnick, P. and Varian, H. *Recommender Systems*, in Communications of the ACM, vol. 30(3), pp. 56-58, March, 1997.
- [44] Goldberg, D., Nichols, D., Oki, B., and Terry, D. *Using Collaborative Filtering to Weave an Information Tapestry*, in Communications of the ACM, vol. 35(12), pp. 61-70, December, 1992.
- [45] ETANA-DL prototype, 2004. <http://feathers.dlib.vt.edu:8080/etana/servlet/Start> (as of April, 2004).
- [46] ACM Digital Library, 2004. <http://portal.acm.org/dl.cfm> (as of April, 2004).
- [47] Ravindranathan, U., Shen, R., Goncalves, M. A., Fan, W., Fox, E. A., and Flanagan, J. W. *ETANA-DL: Managing Complex Information Applications – An Archaeology Digital Library*. To be presented at Joint Conference on Digital Libraries (JCDL 2004), Tucson, AZ, June 7-11, 2004.
- [48] Wise, A. and Miller, P. *Why metadata matters in archaeology*, in Internet Archaeology, 1997. http://intarch.ac.uk/journal/issue2/wise_index.html.
- [49] Schloen, J. D. *Archaeological Data Models and Web Publication Using XML*, in Computers and the Humanities, vol. 35(2), pp. 123-152, May, 2001.
- [50] Doan, A., Domingos, P., and Halevy, A. *Reconciling Schemas of Disparate Data Sources: A Machine-Learning Approach*. Presented at ACM SIGMOD 2001, Santa Barbara, CA, May 21-24, 2001.
- [51] Fuhr, N., Großjohann, K., and Kriewel, S. *A Query Language and User Interface for XML Information Retrieval*, in Intelligent Search on XML Data, Applications, Languages, Models, Implementations, and Benchmarks, vol. 2818, pp. 59-75, September, 2003.

- [52] Goncalves, M. A., Fox, E. A., Krowne, A., Calado, P., Laender, A. H. F., Silva, A. S. d., and Ribeiro-Neto, B. *The Effectiveness of Automatically Structured Queries in Digital Libraries*. To be presented at Joint Conference on Digital Libraries (JCDL 2004), Tucson, AZ, June 7-11, 2004.
- [53] Torres, R. d. S., Medeiros, C. B., Goncalves, M. A., and Fox, E. A. *An OAI Compliant Content-Based Image Search Component*. To be presented at ACM-IEEE Joint Conference on Digital Libraries (JCDL 2004), Tucson, AZ, June 7-11, 2004.
- [54] Goncalves, M. A., Panchanathan, G., Ravindranathan, U., Krowne, A., Fox, E. A., Jagodzinski, F., and Cassel, L. *The XML Log Standard for Digital Libraries: Analysis, Evolution, and Deployment*. Presented at Joint Conference on Digital Libraries (JCDL 2003), Houston, TX, May 27-31, 2003.

Appendix A: Installing ETANA-DL

Note: The following document has been updated to reflect changes made to the application. It corresponds to the following versions of installation files:

ODL Components: [odl3.tar](#)

ETANA-DL Application: [etana14.zip](#) / [etana14.tar](#)

Setting up ETANA-DL on your server is a two-phase process. You have to setup the Open Digital Libraries (ODL) component pool and the related databases in the first phase. Second phase consists of setting up the application on your server.

Section 1

This section describes the steps to download and install the ODL Component pool and create the necessary databases on your server.

Requirements:

Linux based server (we used Red Hat 9.0 during the development/testing process) and MySQL server installed and running on that machine. You will need root access on the Linux machine and permissions to create the databases and grant access to them on the MySQL server.

1. Setup Linux user account

- Create a user account 'etana' and define the home directory to be /home/etana
- You will need to give permission to run server scripts in the cgi-bin directory residing in /home/etana. The cgi-bin directory will be created when you install the ODL components.
- Define an alias for the cgi-bin directory. If your web-server is the Apache web-server that is installed when you install Linux, you can include the following in the httpd.conf file at the appropriate location:
> **ScriptAlias /etana/ /home/etana/cgi-bin/**

2. Download the ODL component pool and other needed scripts

- Download the latest version of the component pool setup from <http://feathers.dlib.vt.edu/~etana/Versions/odl/>. Download the file with the greatest version number, odl<version>.tar.
- Extract the contents of this file.
> **tar -xvf odl<version>.tar**
- You might need to change the ownership of the cgi-bin folder and the permissions on that folder.
> **chown -R etana cgi-bin/**
> **chmod -R 755 cgi-bin/**

3. Setting JAVA_HOME and CLASSPATH

- Create system environment variable in your profile named JAVA_HOME, and set its value to <java installation directory>/;
- Create system environment variable named CLASSPATH, and set its value to “.:”<javainstallationdirectory>/bin/”:<tomcatdirectory>/common/lib/servlet.jar”
- For Windows users, the newly set environment variables will not have any effect on previously open terminal windows. Hence, it is advisable to start a new terminal window for the remaining steps. (For Linux users, you will have to make sure your profile changes are applied before proceeding further. One way is to logout and login back again.)

4. Setting JDBC driver for MySQL

- Download JDBC driver for MySQL (mysql-connector-java-3.0.10-stable.tar.gz or later version) from
<http://www.mysql.com/downloads/api-jdbc-stable.html>
- Unzip the downloaded tar file to extract mysql-connector-java-3.0.10-stable-bin.jar, and put that .jar file under /<java installation directory>/jre/lib/ext

5. Setup the Databases

- Change the path to /home/etana/cgi-bin/EtanaDBSetup/
- For each script file in the folder, for each grant statement in the file, change “feathers.dlib.vt.edu” to your server-name.
- Execute all the scripts in the folder. Order of execution is not relevant.
>mysql -u root -p < <scriptname>
- If you are setting up the ODL components on one machine and the application on another one, you will have to grant appropriate access rights so that the application can access the databases from the remote machine. Setting them up on the same machine will require no extra steps.

6. Setup the Union Catalog

- Change you path to .../cgi-bin/ODL-DBUnion-1.2/DBUnion/
- Execute the configure.pl script
> ./configure.pl etanaunion
- Respond to the following queries during the setup stage as follows:
> Database driver: mysql
> Database: etanaunion
> Username: etana
> Password: etana1234
 (If database connection fails, there is some problem with setting up of the databases.)
> Table prefix: <enter>
> Repository Name: ETANA Union Engine
> Admin Email: <your email>
> Record Limit: 1000
- You now have to give a list of archives from where you want to harvest the records. As an example, we will use the Archives residing on our servers.

- Enter 'a' to add an archive.
- Enter **etanafigurineslahav** as the archive identifier (it can be anything but a meaningful name is preferable).
- For the URL, enter:

http://feathers.dlib.vt.edu/etana/OAI-XMLFile-2.1/XMLFile/etanafigurineslahav/oai.pl

- > Harvesting Interval: **<enter>**
- > Harvesting Overlap: **<enter>**
- > Harvesting Granularity: **<enter>**
- > MetadataPrefix: **etana** (If you do not see this metadata format in the list, there is something wrong.)
- > Sets: **<enter>** (all Sets)
- You can add other archives as per necessary. Some other archives URLs we are hosting at VT that you can use for testing are:

http://feathers.dlib.vt.edu/etana/OAI-XMLFile-2.1/XMLFile/etanabonesnimrin/oai.pl

http://feathers.dlib.vt.edu/etana/OAI-XMLFile-2.1/XMLFile/etanaseedsnimrin/oai.pl

http://feathers.dlib.vt.edu/etana/OAI-XMLFile-2.1/XMLFile/etanabonesumayri/oai.pl

- Enter 'd' when you are done adding archives.
- Harvest the records from these archives by running the harvester as instructed by the component (Example: > ./etanaunion/harvest.pl). Harvesting the records might take a lot of time, depending on the size of the archive.

7. Setup the Browse Engine

- Change your path to **.../cgi-bin/Browse**
- Open **BrowseIndex.java** for editing, and find and replace all "feathers.dlib.vt.edu" with your server name where the Databases reside.
- Open **periodIndex.java** and do the same as above (find and replace).
- Save and exit.
- Compile the files:
 - > **javac *.java**
- Run the setup files in the following sequence:
 - > **java periodIndex**
 - > **java BrowseIndex**

8. Setup the Search Engine

- Change you path to **.../cgi-bin/ODL-IRDB-1.3/IRDB/**
- Execute the **configure.pl** script
 - > **./configure.pl etanasearch**
- Respond to the following queries during the setup stage as follows:
 - > Database driver: **mysql**
 - > Database: **etanasearch**
 - > Username: **etana**
 - > Password: **etana1234**

<If database connection fails, there is some problem with setting up of the databases.>

> Table prefix: **<enter>**

> Repository Name: **ETANA Search Engine**

> Admin Email: **<your email>**

> Archive Identifier: **etanaunion**

> URL: **http://<server-name>/etana/ODL-DBUnion-1.2/DBUnion/etanaunion/union.pl**

(**<server name>** is the name of your server hosting the Union Catalog of harvested records. If the connection fails, there is something wrong in setting up the Union Catalog.)

> Harvesting Interval: **<enter>**

> Harvesting Overlap: **<enter>**

> Harvesting Granularity: **<enter>**

> MetadataPrefix: **etana** (If you do not see this metadata format in the list, there is something wrong.)

> Sets: **<enter>** (all Sets)

- Run the Harvester (harvest.pl) as instructed. The indexing process might take a lot of time depending on the size of the Union Catalog.

9. Setup the Recommendation Engine

- Change you path to **.../cgi-bin/ODL-Suggest-1.0/Suggest/**

- Execute the configure.pl script

> **./configure.pl etanasuggest**

- Respond to the following queries during the setup stage as follows:

> Database driver: **mysql**

> Database: **etanasuggest**

> Username: **etana**

> Password: **etana1234**

(If database connection fails, there is some problem with setting up of the databases.)

> Table prefix: **<enter>**

> Repository Name: **ETANA Recommendation Engine**

> Email: **<your email>**

> Record Limit: **<enter>**

> Archive ID: **<enter>**

> Allow Read: (Enter the IP addresses of all machines that will read from this archive, in this case, the machine where the ETANA Application will reside, and the one where the components have been setup. In our case, because the application and the components resided on the same machine, just entering 127.0.0.1 was sufficient.)

> Allow Write: **<Same IP Addresses as Allow Read>**

> Max Recent: **<enter>**

> Max Vote: **<enter>**

> Max Count: **<enter>**

10. Setup the Annotation Component

- Change you path tocgi-bin/ODL-Thread-1.1/Thread/
- Execute the configure.pl script
 > **./configure.pl etanaannotate**
- Respond to the following queries during the setup stage as follows:
 - > Database driver: **mysql**
 - > Database: **etanaannotate**
 - > Username: **etana**
 - > Password: **etana1234**(If database connection fails, there is some problem with setting up of the databases.)
 - > Table prefix: **<enter>**
 - > Repository Name: **ETANA Annotation Archive**
 - > Email: **<your email>**
 - > Record Limit: **<enter>**
 - > Archive ID: **<enter>**

11. Testing the components

- It is necessary to test the components to be sure all of them have been setup. Besides, some of these components need to be invoked to create the necessary tables in the database. There is a tool available to test these components: The OAI Repository Explorer: <http://oai.dlib.vt.edu/cgi-bin/Explorer/oai2.0/testoai>
- To test the components (Union, Search, Annotate, and Recommend), the following are the URLs:

http://<server-name>/etana/ODL-DBUnion-1.2/DBUnion/etanaunion/union.pl
http://<server-name>/etana/ODL-IRDB-1.3/IRDB/etanasearch/search.pl
http://<server-name>/etana/ODL-Thread-1.1/Thread/etanaannotate/thread.pl
http://<server-name>/etana/ODL-Suggest-1.0/Suggest/etanasuggest/suggest.pl

- To test them, you just have to click on the Identify link in the Verbs section of the page. If you get some an error returned, then setup of the corresponding component did not succeed. If not, you will get details about the components such as name of archive, e-mail of administrator, etc., and you are all set for the second phase.

Section 2

This section describes the steps to be followed to download and install the ETANA-DL application. We will describe the installation procedures on a Linux machine. Setting up the ETANA-DL application on a Windows machine is analogous.

12. Setting up the Java runtime and Apache Tomcat web-container on your machine

- Download and install Java 1.4.2 SDK from <http://java.sun.com/j2se/1.4.2/download.html>. We assume that you have installed Java at /usr/local/java (not necessary)
- Download the Tomcat binaries (we used version 4.1.29, but a later version might be available to choose) from <http://jakarta.apache.org/site/binindex.cgi>
- Install Tomcat at /Tomcat (again not necessary, but you will have to make corresponding changes if you install at a different location).
- Start the Tomcat server. You can test if the server is running by opening a browser and pointing it to <http://localhost:8080/>. You should be able to see a page with a message informing you that Tomcat is not running. If you cannot see this page, then there is some problem in setting up or starting Tomcat. You can get answers to most questions at <http://jakarta.apache.org/tomcat/faq/>

13. Getting the Application

- We frequently mirror the application at <http://feathers.dlib.vt.edu/~etana/Versions/apps/>
 - Download the latest version – etana<highest version number>.tar and unzip the contents to <tomcat directory>/webapps/.
- FOR LINUX/UNIX SETUP:
Extracting the files into a folder will also require changing the permissions (use `chmod -R 755 etana/`)

14. Modifying and compiling the application

The application will need a little bit of modification, as we have not entirely packaged the application yet. However, we have tried to make the process as easy as possible.

- Change your working directory to <tomcat directory>/webapps/etana/WEB-INF/classes/.
 - Find and replace all “feathers.dlib.vt.edu” in each of the “.java” files with your server name; example: abc.xyz.edu (assuming that the ODL components and the applications and the database all reside on the same server.). If not please, contact us and we will help you configure your application accordingly.
 - You might also want to replace the values of 2 variables, AdminName and AdminEmail, with the Name and Email of the Administrator for ETANA-DL in “Helper.java”.
 - Save all your changes.
- At the command prompt, compile all files by executing `javac *.java`.

15. Testing the application

- To test the application, **restart Tomcat**. By default, Tomcat server will need to restart each time you make any changes to the application and recompile it.
- <server-name>:8080/etana/servlet/Start will be your base URL to test the application. You should be able to see the ETANA-DL home page.
- If you get a page not found error, you might want to first check if it is a firewall related issue.

Appendix B: ETANA-DL prototype XML Schema

B.1 XML Schema Diagrams

Figure 41: XML Schema overview (generated using XMLSpy)

Figure 42: Detailed schema diagrams for three digital objects in ETANA-DL (generated using XMLSpy)

B.2 XML Schema

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema targetNamespace="http://feathers.dlib.vt.edu/~etana/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:etana="http://feathers.dlib.vt.edu/~etana/" elementFormDefault="qualified"
attributeFormDefault="unqualified">
<xs:element name="OBJECT">
  <xs:annotation>
    <xs:documentation>ETANA-DL XML Schema</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="COLLECTION" type="xs:string"/>
      <xs:element name="OBJECTTYPE" type="xs:string"/>
      <xs:element name="OWNERID" type="xs:string"/>
      <xs:element name="ID" type="xs:decimal" minOccurs="0"/>
      <xs:element name="PARTITION" type="xs:string" minOccurs="0"/>
      <xs:element name="SUBPARTITION" type="xs:string" minOccurs="0"/>
      <xs:element name="LOCUS" type="xs:string" minOccurs="0"/>
      <xs:element name="CONTAINER" type="xs:decimal" minOccurs="0"/>
      <xs:choice>
        <xs:element name="SEED" type="etana:SeedDescription"/>
        <xs:element name="BONE" type="etana:BoneDescription"/>
        <xs:element name="FIGURINE" type="etana:FigurineDescription"/>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:complexType name="SeedDescription">
  <xs:sequence>
    <xs:element name="NAME" type="xs:string" minOccurs="0" />
    <xs:element name="QUANTITY" type="xs:decimal" minOccurs="0"/>
    <xs:element name="SPECIES" type="xs:decimal" minOccurs="0"/>
    <xs:element name="COMMENTS" type="xs:string" minOccurs="0"/>
    <xs:element name="DATE_DUG" type="xs:dateTime" minOccurs="0"/>
    <xs:element name="CDATE" type="xs:dateTime" minOccurs="0"/>
    <xs:element name="UDATE" type="xs:dateTime" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="BoneDescription">
  <xs:sequence>
    <xs:element name="RANK" type="xs:decimal" minOccurs="0"/>
    <xs:element name="AGES" type="xs:string" minOccurs="0" />
    <xs:element name="AGE" type="xs:string" minOccurs="0"/>
    <xs:element name="PAGE" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
</xs:schema>
```



```

<xs:element name="LINE" type="xs:decimal" minOccurs="0"/>
<xs:element name="PIECES" type="xs:decimal" minOccurs="0"/>
<xs:element name="BONENAME" type="xs:string" minOccurs="0"/>
<xs:element name="SIDE" type="xs:string" minOccurs="0"/>
<xs:element name="ANIMAL" type="xs:string" minOccurs="0"/>
<xs:element name="COMMENTS" type="xs:string" minOccurs="0"/>
<xs:element name="DATE_DUG" type="xs:dateTime" minOccurs="0"/>
<xs:element name="CDATE" type="xs:dateTime" minOccurs="0"/>
<xs:element name="UDATE" type="xs:dateTime" minOccurs="0"/>
<xs:element name="ONE" type="xs:decimal" minOccurs="0"/>
<xs:element name="TOTS" type="xs:decimal" minOccurs="0"/>
<xs:element name="SQRS" type="xs:string" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="FigurineDescription">
  <xs:sequence>
    <xs:element name="SHORTDESCRIPTION" type="xs:string"/>
    <xs:element name="LONGDESCRIPTION" type="xs:string"/>
    <xs:element name="CONDITION" type="xs:string"/>
    <xs:element name="TECHNIQUE" type="xs:string"/>
    <xs:element name="DIMENSIONS" type="xs:string"/>
    <xs:element name="WAREPASTE" type="xs:string"/>
    <xs:element name="COMPARISON" type="xs:string"/>
    <xs:element name="DIG" type="xs:string"/>
    <xs:element name="DATESTAMP" type="xs:string"/>
    <xs:element name="IMAGEURL" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
</xs:schema>

```

Vita

Unnikrishnan Ravindranathan was born in Trivandrum, India, in April 1981. He earned his Bachelor of Engineering in Computer Science and Engineering in May 2002 from the Faculty of Technology and Engineering, M. S. University of Baroda, with distinction.

He came to the United States in pursuit of a Master of Science degree from Virginia Tech with concentration in Computer Science in fall of 2002. During his graduate studies, he initially worked as a Graduate Teaching Assistant in the Department of Computer Science, and later as a Research Assistant to Dr. Edward Fox for the Digital Library Research Laboratory. He spent the summer of 2003 as an intern with Microsoft Corp. in Redmond, WA. Other than information retrieval and digital libraries, his interests in Computer Science include peer-to-peer networking and information-rich user interfaces.

The submission of this work completes his Master of Science requirements. Upon graduation, he will be moving to Redmond, WA to take up a position as a Software Engineer with Microsoft Corp.