Applying the 5S Framework To Integrating Digital Libraries

Rao Shen

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Science and Applications

Advisory Committee:

Edward A. Fox, Chair
Joanne D. Eustis
Weiguo Fan
Chang-Tien Lu
Manuel A. Pérez-Quiñones

04/17/2006

Blacksburg, Virginia

Keywords: digital libraries, integration, interoperability, exploring service, quality.

Applying the 5S Framework To Integrating Digital Libraries

Rao Shen

Abstract

We formalize the digital library (DL) integration problem and propose an overall approach based on the 5S (Streams, Structures, Spaces, Scenarios, and Societies) framework. We then apply that framework to integrate domain-specific (archaeological) DLs, illustrating our solutions for key problems in DL integration. An integrated Archaeological DL, ETANA-DL, is used as a case study to justify and evaluate our DL integration approach. We develop a minimum metamodel for archaeological DLs within the 5S theory. We implement the 5SSuite toolkit set to cover the process of union DL generation, including requirements gathering, conceptual modeling, rapid prototyping, and code generation. 5SSuite consists of 5SGraph, 5SGen, and SchemaMapper, which plays an important role during integration. SchemaMapper, a visual mapping tool, maps the schema of diverse DLs into a global schema for a union DL and generates a wrapper for each individual DL. Each wrapper transforms the metadata catalog of its DL to one conforming to the global schema. The converted catalogs are stored in the union catalog, so that the union DL has a global metadata format and union catalog. We also propose a formal approach to DL exploring services for integrated DLs based on 5S, which provides a systematic and functional method to design and implement DL exploring services. Finally, we propose a DL success model to assess integrated DLs from the perspective of DL end users by integrating 5S theory with diverse research on information systems success and adoption models, and information-seeking behavior models.

ACKNOWLEDGEMENTS

I have been very fortunate to start my Ph. D. program at Virginia Tech. Very special thanks to those people who make this journey possible, enjoyable, and meaningful.

Professor Edward A. Fox is the most important to me. It is my heartfelt joy to acknowledge all his guidance and warm support. I admire his commitment and devotion to his vocation. He is not only my thesis advisor but also a very special friend and great teacher for my spiritual and personal growth. I would have never survived my graduate program without the energy, encouragement, and inspiration from him.

I am also extremely grateful to Dr. Patrick Fan. Dr. Fan's dedication to his profession is reflected by his concern for students. He has been one of my best friends who I can always bother with questions and problems.

Special thanks also go to Dr. Chang-Tien Lu for his ideas and feedback on my work. Many thanks to Dr. Manuel A. Pérez-Quiñones for bringing his expertise and introducing Rob Capra's work to me. I also deeply appreciate Dr. Joanne D. Eustis for joining my committee as an external faculty member and contributing comments and suggestions from the library science perspective.

I have been blessed with the friends in our lab. In addition, I am very much indebted to Srinivas Vemuri, Doug Gorton, Johnny L. Sam-Rajkumar, Marcos André Gonçalves, Ananth Raghavan, and Unni Ravindranathan. Thank you all!

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1. Introduction

1.1. The Problem

Digital Libraries (DLs) are transforming research, scholarship, and education at all levels. One of the intriguing aspects of DL research is that challenges exist at both the fundamental technology level and at the large-scale integration level. A decade of government and private funding of DL research projects has led to important results at the fundamental technology level. The successes in large-scale integration are arguably less evident. Even the notion of "DL integration" is ambiguous in the sense that different approaches and proposed solutions exist. Work on DL integration focuses to an extent on three issues [59]:

1) Distribution: geographical spread;

2) Heterogeneity: difference at both the technical level (e.g., hardware platform, operating system, programming language, etc.) and conceptual level (e.g., different understanding and modeling of the same real-world entity);

3) Autonomy: the extent to which the components are self-sufficient, as opposed to being delegated a role only as components in a larger system.

By "DL integration", we mean hiding distribution and heterogeneity, while at the same time enabling and making visible component autonomy (at least to some degree).

Many DLs belonging to different autonomous organizations were developed independently without plans to provide open and easy automated access to their data and functionality. The inability to seamlessly and transparently access knowledge across DLs is a major impediment to knowledge sharing. The goal of DL integration then is to utilize various autonomous DLs in concert to provide knowledge from such island-DLs. The needs for DL integration are well known, and better known than the solutions [79].

Challenges to DL integration are a direct result of DL characteristics. DLs are complex information systems due to their inherently interdisciplinary nature, both with regard to application domains and technologies involved in building the systems. Concerning the latter, DL system implementations integrate findings from disciplines such as hypertext, information

retrieval, multimedia services, database management, and human-computer interaction [42]. Hence, an integrative theory for DL is needed. [53] summarizes key early work on the 5S (Streams, Structures, Spaces, Scenarios, and Societies) framework, and related efforts to construct such an integrative theory for DLs. The 5S framework allows us to define digital libraries rigorously and usefully. Streams are sequences of arbitrary items used to describe both static and dynamic (e.g., video) content. Structures can be viewed as labeled directed graphs, which impose organization. Spaces are sets with operations that obey certain constraints. Scenarios consist of sequences of events or actions that modify states of a computation in order to accomplish a functional requirement. Societies are sets of entities and activities, and the relationships among them. Together these abstractions provide a formal foundation to define, relate, and unify concepts – among others, of digital objects, metadata, collections, and services – required to formalize and elucidate "digital libraries" [50].

DL integration can be at different levels, e.g., information level and service level. Integrated information makes distributed collections of heterogeneous resources appear to be a single union. Integrated services afford users more comprehensive usage of DL resources through more coherent and easier to use interfaces that hide syntax and semantic differences in the DLs to be integrated. While many efforts have looked into the DL integration problem, most developed their own approaches in an ad hoc and piecemeal fashion.

Developing an infrastructure to address all perspectives of the DL integration problem is an ambitious task. In this dissertation, we formalize the DL integration problem and propose an overall approach based on the 5S framework. We apply our framework to integrate domain specific (archaeological) DLs, illustrating our approaches to key sub-problems (e.g., semantic interoperability) of DL integration.

1.2. Hypothesis and Research Questions

We claim that the 5S framework provides effective solutions to DL integration. This hypothesis leads to the following research questions.

1. Can we formally define the DL integration problem, using the 5S framework?

2. Can the 5S framework guide integration of domain/discipline focused DLs (e.g., integrate systems for diverse archaeological sites into a union archaeological DL)? If yes, how? Specifically:

How can we formally model such domain specific DLs in the 5S framework?

How can we integrate DL models into a union DL model?

How can we use the union DL model to help design and implement high quality integrated DLs?

3. Can we assess an integrated DL based on some indicators and metrics? What are those? How well does the integration work in practice?

1.3. Problem Formalization and Overall Approach

Formalizing DL integration facilitates the development, comparison, and evaluation of solutions; makes clear to users what a solution means; and helps users evaluate the applicability of a solution. Furthermore, it allows us to leverage special-purpose techniques for the DL integration process. In this section, we first give a background to the 5S framework, based on which we formally define the DL integration problem. We then propose an overall approach and a toolkit for DL integration.

1.3.1. Background on the 5S framework

Gonçalves et al. [53] presented a formal framework for the DL field, summarized in Fig. 1.3.1. A "minimal digital library" (Def. 24 of [53] shown at the bottom right) was defined as the highest level concept. Fig. 1.3.1 illustrates the supporting layers of definitions: mathematical foundations (e.g., graphs, sequences, and functions), the 5 Ss (Streams, Structures, Spaces, Scenarios, and Societies), and key concepts of a DL (e.g., digital object, collection). Arrows represent dependencies, indicating that a concept is formally defined in terms of previously defined concepts that point to it. We adopt the definitions in [53] and extend them in the discussion below.

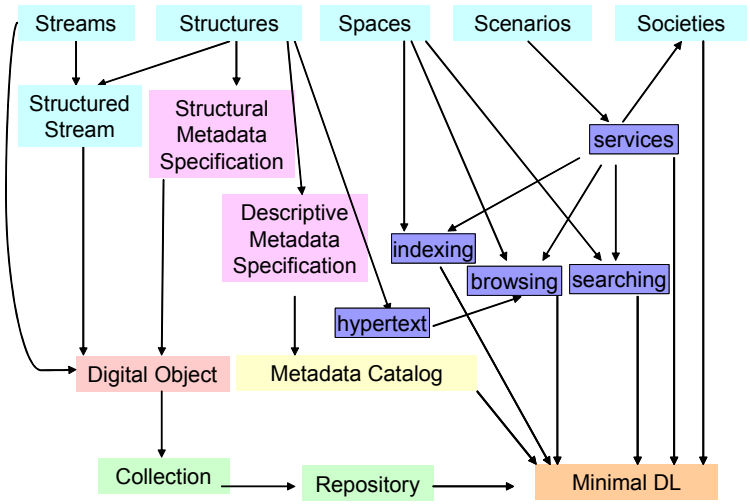


Fig. 1.3.1 5S definitional structure ([50])

1.3.2. Notation and definitions

Notation: Let $DL_1, DL_2, \ldots, DL_i, \ldots, DL_n$ be n independent digital libraries; let Id_i be a unique identifier of DL_i; let C_{ij} be the j-th collection of DL_i; let $C_i = \cup_{j=1}^{m} C_{ij}$, where m is the total number of collections of DL_i; let $UnionC = \bigcup_{i=1}^{n} C_i$ be a union collection of the n DLs; let H be a set of universally unique handles.

Following [53] we have $DL_i=(R_i, DM_i, Serv_i, Soc_i)$, where R_i is a network accessible repository, supporting some type of harvesting protocol to expose its metadata; DM_i is a set of metadata catalogs for C_i; $Serv_i$ is a set of services; and Soc_i is a society.

Definition 1: A Union Repository (***UnionRep***) of n DLs ($DL_1, \ldots, DL_n$) is a DL repository ([53]) with a *getDL_Id* function: ***UnionRep*** = *(CollSet, getDL_Id, get, store, del)*, where

1) $CollSet \subseteq 2^{\{UnionC\}}$;

2) $getDL\_Id: UnionC \rightarrow \{Id_1, Id_2, \ldots, Id_i, \ldots, Id_n\}$ maps a digital object *do* to the DL it belongs to.

3) $get: H \rightarrow UnionC$ maps a handle h to $do=get(h)$;

4) $store: UnionC \times CollSet \rightarrow CollSet$ maps $(do, \tilde{C})$ to the augmented collection $\{do\} \cup \tilde{C}$;

5) $del: H \times CollSet \rightarrow Collset$ maps $(h, \tilde{C})$ to the smaller collection $\tilde{C}$-$\{get(h)\}$;

Definition 2: A Union Catalog ***UnionCat*** $=DM_{UnionC}$ is a metadata catalog for *UnionC*.

Definition 3: Minimal Union Services ***(MinUnionServ)*** = $\{harvesting, mapping\} \cup (\cup_{i=1}^{n} Serv_i)$.

The *harvesting* service provides a mechanism to gather metadata from each DL_i; the *mapping* service supports transforming information organized by local schema to information structured according to the global schema. The *harvesting* service is formally defined in [51]; the *mapping* is defined as follows (see definitions 4-7):

Definition 4: A schema is a structure ([53]) with a domain D of data types (e.g., strings, numbers, dates, etc.). **schema** = *((V, E), L, F, D, M)*, where *(V, E)* is a graph with vertex set *V* and edge set *E*, *L* is a set of label values, *F* is a labeling function $F: (V \cup E) \rightarrow L$, and *M* is a function $M: V \rightarrow D$.

Definition 5: Given a schema *((V, E), L, F, D, M)*, its element set = $\{(v, F(v))\} \cup \{(e, F(e))\}$.

Definition 6: 1-1 mapping

Let *S* and *T* be two element sets, of *S_Schema* and *T_Schema,* respectively. 1-1 mapping is a function: $M_{1\text{-}1}: S \times T \rightarrow Sim$, where $\forall sim \in Sim,\ 0 \leq sim \leq 1$. A tuple (*s, t, sim*) indicates element *s* of *S* is similar to element *t* of *T* with confidence score *sim*. The higher a confidence score, the more semantically similar are *s* and *t.*

Definition 7: complex mapping

Let *S* and *T* be two element sets, of *S_Schema* and *T_Schema,* respectively; let *O* be a set of operators that can be applied to elements of *S* and *T,* according to a set of rules *R,* to construct formulas; and let $Formu_s$ and $Formu_t$ be two sets of formulas constructed from the elements of *S* and *T,* using *O*. Complex mapping is a function: M: $(S \cup Formu_s) \times (T \cup Formu_t) \rightarrow Sim$, where $\forall sim \in Sim,\ 0 \leq sim \leq 1$.

Definition 8: A Union Society ***UnionSoc*** = $\cup_{i=1}^{n} Soc_n$

Definition 9: A Minimal Union Digital Library integrated from *n* DLs (see notation above) is given as a four-tuple: ***MinUnionDL***$=(R_{union}, DM_{union}, Ser_{union}, Soc_{union})$, where R_{union}, DM_{union}, Ser_{union}, Soc_{union} are Union Repository, Union Catalog, Minimal Union Services, and Union Society, respectively. A Union DL is a superset of a ***MinUnionDL***. "Integrated DL" and "Union DL" will be used interchangeably in this paper.

Definition 10: DL Integration Problem Definition

Given *n* individual digital libraries ($DL_1, DL_2, ..., DL_n$), each defined as described above, to integrate the *n* DLs is to create a Union DL.

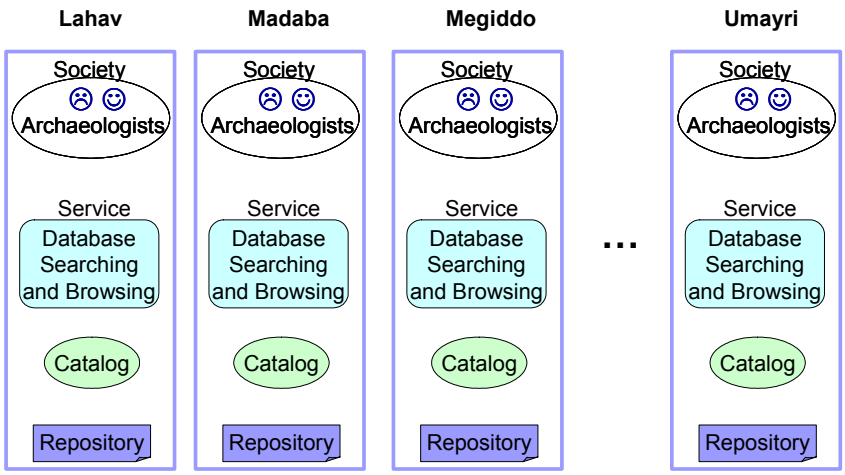
1.3.3. Architecture of an integrated DL

As above (Def. 9), an integrated DL is a 4-tuple consisting of a union repository, a union catalog, union services, and a union society. There are three popular integration architectures to deal with regarding the first two components of the definition, namely: 1) a centralized union catalog along with a centralized union repository; 2) a centralized union catalog for a decentralized union repository; and 3) a middle ground between the above two extremes of the spectrum, i.e., a centralized union catalog with a partially centralized union repository.

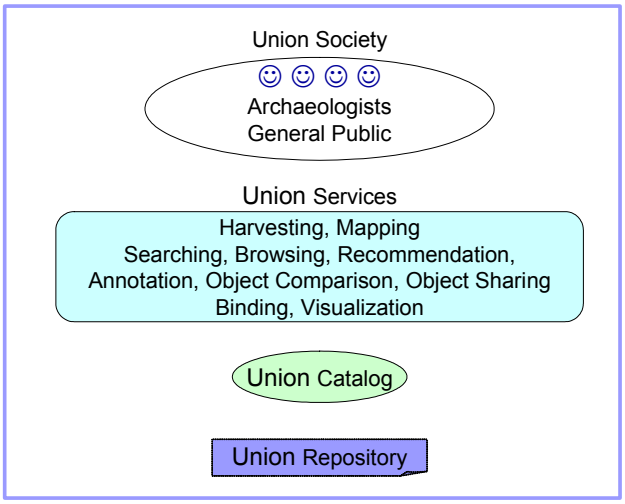
Decision on the architecture to be used to develop an integrated DL is based on 1) what contents (metadata, digital objects, or both) the DLs to be integrated would like to share; and 2) what the integrated DL needs to harvest. The former relates to copyrights and publication rights. The latter may involve issues such as scalability, consistency, and preservation.

Having both a centralized union catalog and a centralized union repository in an integrated DL can guarantee adequate performance at information seeking time. No burden is placed on the remote DLs to retrieve results. Storing digital objects in the integrated DL redundantly can help preservation. However, delivery of the most current information to users cannot always be guaranteed. Changes to the metadata and digital objects by the individual DLs need to be propagated to the integrated DL.

Assumed for a decentralized union repository is that the metadata contains links to concrete realizations of digital objects. Its main disadvantage is that retrieval of digital objects relies on remote DLs. CITIDEL [114] is a DL that has a centralized catalog and decentralized repository; sustainability of the centralized portion of such a system also can be a challenge.



(a) Member DLs of ETANA-DL



(b) Architecture of ETANA-DL, with centralized catalog and partially decentralized repository

Fig. 1.3.2 An example of an integrated DL: ETANA-DL

To create a centralized catalog, ETANA-DL provides a harvesting service and a mapping service. Beside these two, it should provide all the services supported by its member DLs (e.g., searching and browsing), and other services (e.g., clustering and visualization). The visualization service may integrate searching, browsing, and clustering. EtanaViz [43] is an example of such an integrated service. It provides a visual interface to ETANA-DL. Search results can be classified by predefined classes. Grouped documents are displayed in several ways to help browsing.

Table 1 lists 12 of the 31 suggestions for services of an integrated National Science Digital Library (NSDL) from the results of interviews reported in [45]. Those suggestions appeared three or more times in the interview notes, and help guide a taxonomy of typical union DL's union services.

Table 1 Suggested NSDL Services

Suggested NSDL Services
1. Register & share interactive widgets or services (beyond OAI)
2. Service to support annotation
3. Browse framework for finding tools
4. Ref. desk with encyclopedia, glossaries, & glossary register
5. Facilitation of collaboration and interaction
6. Contextualization services for NSDL content
7. Metadata validation (informed by analyses), incl. "de-duping"
8. Learning/training about NSDL technologies & tools
9. Content reusability support (at the sub-object level)
10. Schema registry with analyses & (pairwise) mapping
11. Specialized library views, including RSS services
12. Support & guidelines to address granularity differences

Table 2 Taxonomy of Union Services

Infrastructure Services		**Information Satisfaction Services**	
Essential	*Add_Vaue*	*Essential*	*Add_value*
indexing **harvesting** **mapping** **10 (Schema registry with analyses & mapping)**	(data) cleaning (focused) crawling copying (replicating) logging (format) translating 2 (Service to support annotation) 7 (Metadata validation)	**searching** **browsing**	access control binding comparison (forum) discussion (query) expansion filtering recommendation visualization 1, 3, 4, 5, 6, 8, 9, 11, 12

Table 2 shows a taxonomy of union services extended from the taxonomy of DL services in [50]. The key aspects of defining such a taxonomy are: 1) to divide the set of services into those dealing with repositories (and their collections and respective catalogs) and those dealing with societies, more specifically digital library patrons (those for whom the DL is ultimately designed); 2) to characterize essential services belonging to a minimal union DL (shown in bold) and value added services. Services in Table 1 are numbered from 1 through 12, and are shown in Table 2 using those numbers.

The union services illustrated in

Fig. 1.3.2 aim to satisfy users of ETANA-DL's member DLs. The user society in an integrated DL may be simplified as a union of the users of the DLs to be integrated. However, special cases need to be considered, e.g., how to deal with the situations where a user (or her partners) belongs to different user groups of various DLs to be integrated.

1.3.4. Integration toolkit: 5SSuite

The 5S framework allows a new approach to DL development (see Fig. 1.3.3). 5SGraph [156, 157] supports analysis and specification, while 5SGen [72] melds together suitable components from a large software pool to yield a running system. To semi-automatically build an integrated DL, we extend this approach and develop the 5SSuite toolkit to cover the process of union DL generation, including requirements gathering, conceptual modeling, rapid prototyping, and code generation (see Fig. 1.3.4). The 5SSuite tool consists of 5SGraph, 5SGen, and SchemaMapper (described in Chapter 3), which plays an important role during integration.

A DL designer interacts with the 5SGraph tool to model the DLs to be integrated and the union DL, when a metamodel is fed to 5SGraph. Each produced DL model contains a structure sub-model and a scenario sub-model as well as the other three sub-models (i.e., stream, space, and society sub-models). Schemas (metadata formats) are described in the structure sub-model, whereas services are described in the scenario sub-model.

A DL designer interacts with SchemaMapper, which maps a local schema into a global schema for a union DL and generates a wrapper for the DL to be integrated. The wrapper transforms the metadata catalog of its DL to one conforming to the global schema. The converted catalogs are stored in the union catalog, so that the union DL has a global metadata format and union catalog. The mapping process is iterative. When another DL needs to be integrated, the DL designer may use SchemaMapper to help complete mapping and then updating of the union catalog. The complexity of the mapping and updating process can be affected by several factors, such as knowledge of the application domain, the number of elements in the local schema, and the size of the collection to be integrated.

To integrate domain specific DLs, a metamodel for that particular domain needs to be developed based on the 5S formal theory. Chapter 3 describes an archaeological DL (ArchDL) metamodel and the use of 5SGraph to model ArchDLs.

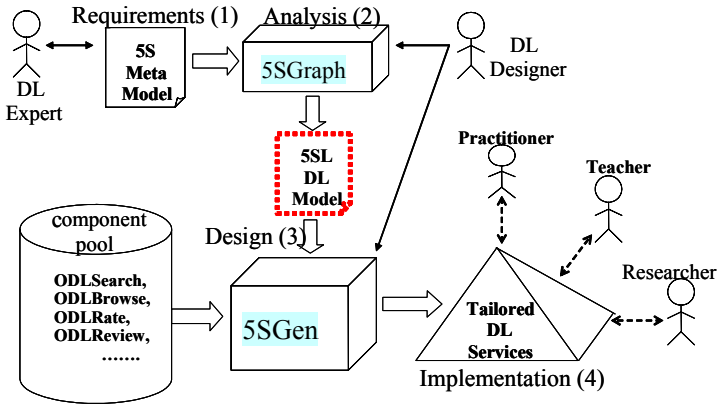


Fig. 1.3.3 5S related tools and their use in developing DLs [50]

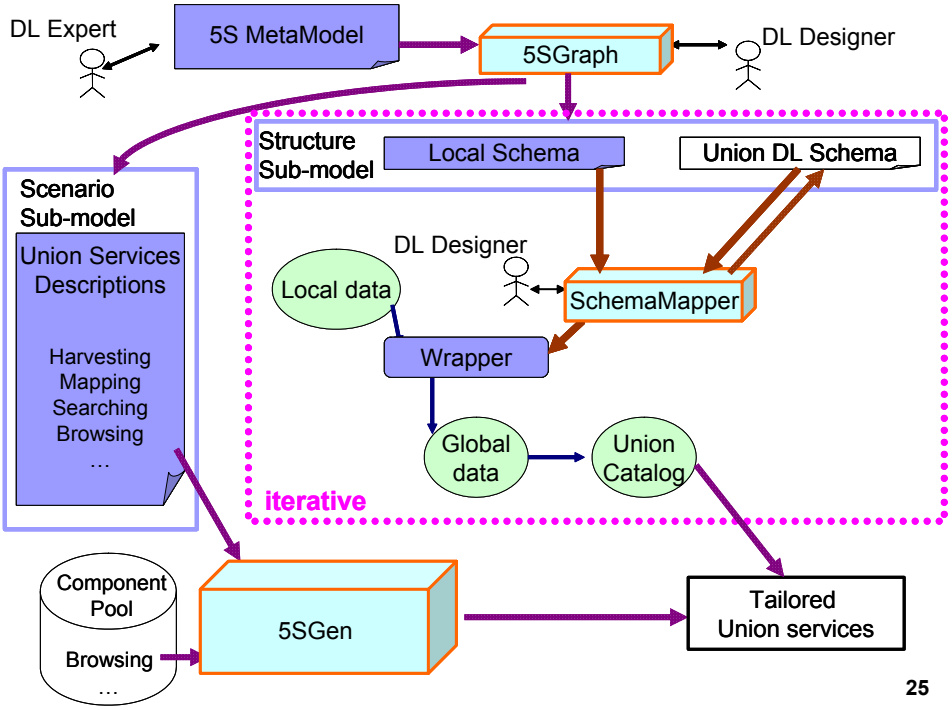


Fig. 1.3.4 5S related integration toolkit and process

1.4. Dissertation Organization

The rest of this dissertation is organized as follows. Chapter 2 is the literature review. Chapter 3 describes DL modeling in the archaeological domain and the use of the 5SGraph [156, 157] tool to create archaeological DL models. Chapter 4 illustrates how the mapping service (one of the services in an integrated DL) provided by the SchemaMapper [118] tool addresses semantic interoperability and helps generate a union catalog. Chapter 5 defines exploring

services in an integrated DL. Chapter 6 proposes a DL success model to assess integrated DLs from the perspective of DL end users. Chapter 7 gives conclusions and outlines future work.

Chapter 2. Literature Review

DL integration can be at different levels, e.g., information level and service level. Integrated information makes distributed collections of heterogeneous resources appear to be a single collection. Integrated services afford users more comprehensive usage of DL resources through more coherent and easier to use interfaces that hide syntatic and semantic differences in the DLs to be integrated. This chapter presents related work at these two levels. Since interoperability is the most important issue when integrating heterogeneous DLs, we conducted a survey on DL interoperability, particularly on DL semantic interoperability. Information integration applications and related research on semantic interoperability are discussed in section 2.1; integrated DL exploring services are discussed in section 2.2.

2.1. Interoperability in Digital Libraries

Interoperability is the most important issue when integrating heterogeneous DLs [1, 106, 120]. It has many dimensions [106, 107] and has been the subject of many initiatives. It is a broad problem domain. It has been typically investigated within a specific scope, such as within a particular community (e.g., libraries, commercial entities, and scientific communities), within a particular type of information (e.g., electronic records, technical reports, and software), or within a particular information technology area (e.g., relational databases, digital imaging, and information visualization) [113]. Various aspects of DL interoperability are depicted in Fig. 2.1.1, a concept map for the problem.

Research on interoperability in DL architectures addresses the challenges of creating a general framework for information access and integration across many of the above domains. A common goal of these efforts is to enable different communities, with different types of information and technologies, to achieve a general level of information sharing and, through the process of aggregation and computation, to create new and more powerful types of information.

There are many approaches to achieving interoperability. Paepcke et al. [106] have categorized many of the prevalent approaches and have provided an informative discussion of the challenges inherent in creating interoperable DLs with global scope. Some of the common

approaches have included: 1) standardization (e.g., schema definition, data model, and protocol), 2) distributed object request architectures (e.g., CORBA), 3) remote procedure calls, 4) mediation (e.g., gateways, wrappers), and 5) mobile computing (e.g., Java applet).

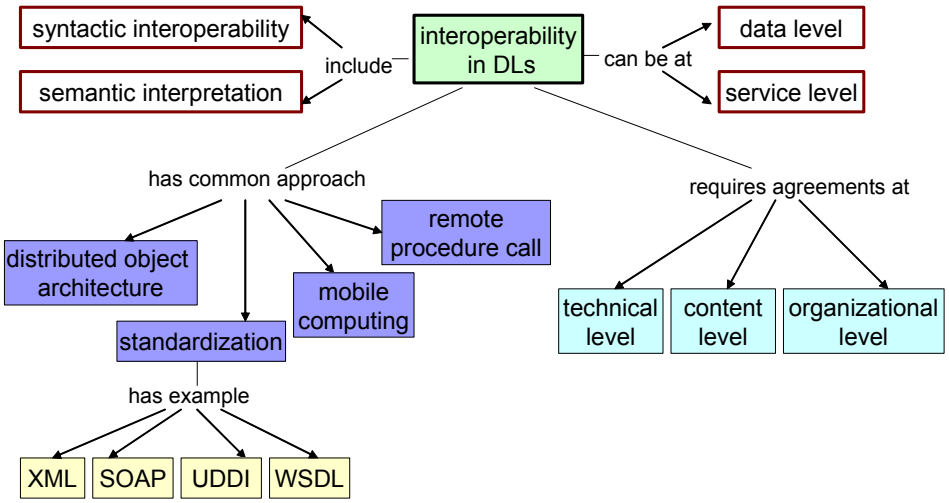


Fig. 2.1.1 Various aspects of interoperability in DLs

To achieve DL interoperability requires agreement to cooperate at three levels: technical, content, and organizational [6]. Technical agreements cover formats, protocols, security systems, etc., so that messages can be exchanged. Content agreements cover the data and metadata, and include semantic agreements on the interpretation of the information. Organizational agreements cover the ground rules for access, preservation of collections and services, payments, authentication, etc.

There are two different types of interoperability for DL integration [108]: syntactic interoperability and semantic interoperability. Syntactic interoperability is the application-level interoperability that allows multiple software components to cooperate even though their implementation languages, interfaces, and execution platforms are different. Semantic interoperability is the knowledge-level interoperability that allows DLs to be integrated, with the ability to bridge semantic conflicts arising from differences in implicit meanings, perspectives, and assumptions, thus creating a semantically compatible information environment based on agreed-upon concepts (among various DLs). Standards such as XML, and Web services based on SOAP (Simple Object Access Protocol), UDDI (Universal Description Discovery and Integration), and WSDL (Web Service Description Language), can resolve many application-

level interoperability problems [108]. However, establishing semantic interoperability among heterogeneous information sources from various DLs continues to be a critical issue. The NSF Post Digital Libraries Futures Workshop [83] identified it as being of primary importance in digital library research. DELOS WP5 [109] reported many issues relating to semantic interoperability in DLs. We present related work concerning semantic interoperability in DLs as follows.

2.1.1. Semantic interoperability in digital libraries

Semantic interoperability in DLs means the capability of different information systems to communicate information consistent with the intended meaning [109]. Information integration is only one possible result of a successful communication. Since the emergence of different human languages, communication could be achieved in two ways: 1) force everyone to learn and use the same language; 2) find translators who know how to interpret sufficiently the information of one participant for another. The first approach is proactive standardization, while the second one is reactive interpretation. This choice applies to all levels and functions of semantic interoperability and is a major distinctive criterion of various methods. Related work concerning DL semantic interoperability is presented in a concept map as shown in Fig. 2.1.2.

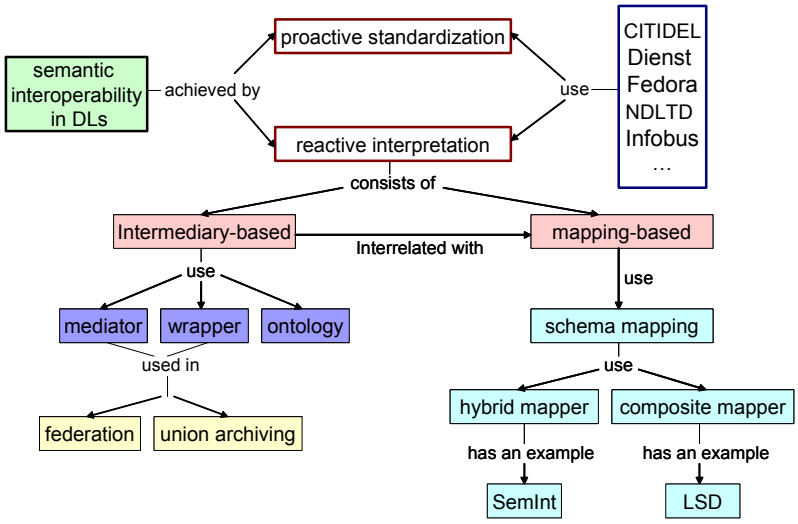


Fig. 2.1.2 A concept map for related work on semantic interoperability in DLs

2.1.1.1. Standardization

One of the traditional approaches to interoperability is for all participants to agree to use the same standards such as metadata standards and transaction protocols.

Standardization has the following advantages:

- Information can be immediately transferred and integrated without transformation and alteration.
- Information can be kept in a single form.
- Information can be enforced to be functionally complete for an envisaged integrated service.

The disadvantages are:

- Information needs adaptation to the standard. The adaptation may require interpretation (manual or automatic) and may result in information loss.
- The effort of producing a standard may be very high.
- A standard has to foresee all future use. Introducing a new element may be time-consuming and may cause upwards-compatibility problems.
- A standard is designed to suit its domain. It may not be optimal for all applications.

A standard is elegant and efficient for specific applications. It is appropriate for problems with low degree of necessary diversity and with high long-term stability.

A model based on the concept of "spectrum of interoperability" shows a way to look at the tradeoffs (see Fig. 2.1.3) [6]. The vertical axis shows the cost to an organization of adopting a standard. The horizontal axis shows the functionality the organization gains. If the cost of adopting a standard is high, it will be adopted only by those organizations that truly value the functionality provided. Conversely, when the cost is low, more organizations will be willing to adopt it, even if the functionality is limited. For example, libraries have developed a framework for interoperability based on the Z39.50 protocol [88], the Anglo American Cataloguing Rules, and MARC. This combination of standards provides an excellent choice for libraries, because they value the functionality and have catalog records in the required form. However, the cost of adopting these standards is high unless an organization already has metadata that conforms.

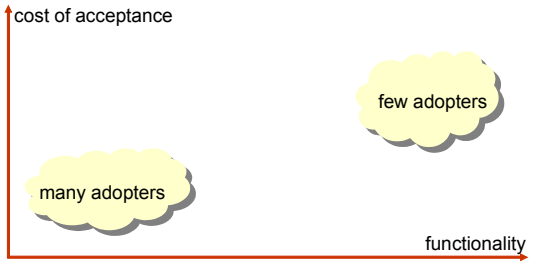


Fig. 2.1.3 Cost of adoption against functionality

Examples of metadata standards are DC (Dublin Core), Dublin Core with DC-Ed extensions, FGDC (Content Standard for Digital Geospatial Metadata), IEEE LOM (Learning Objects Metadata), and METS (Metadata Encoding and Transmission Standard). DC can be considered as a minimal standard, while METS provides a flexible mechanism for encoding descriptive, administrative, and structural metadata for a digital library object, and for expressing the complex relationships between these various forms of metadata. METS therefore can provide a useful standard for the exchange of digital library objects between repositories. In addition, METS provides the ability to associate a digital object with behaviors or services.

2.1.1.2. Interpretation

Examples of interpretation in a DL are mapping of one metadata framework to another (metadata crosswalk) and correlation of concepts defined in knowledge organization systems (KOS).

Interpretation has the following advantages:

- Information to be integrated, in particular legacy data, needs no adaptation. Only application relevant parts need interpretation.
- Interpretation can be optimized for multiple functions and interpreters can easily be adapted to changes.

The disadvantages are:

- The manual effort of producing the knowledge base (e.g., correlation tables for terminologies) for an interpreter can be very high though automatic generation is feasible.
- Interpretation of information may result in information loss.
- The number of interpreters needed increases drastically if the number of formats in use increases. In such situations, interpretation may need to go through a common switching language, which reduces the number of interpreters needed, but increases the loss of precision. Effectively, such a switching language is but another standard.

Interpreters are effective in environments with a high degree of necessary diversity and low long-term stability.

As shown in the concept map about related work concerning DL semantic interoperability in Fig. 2.1.2, two approaches to interoperability through interpretation are interrelated. They are

the intermediary-based approach and the mapping-based approach [108]. The intermediary-based approach depends on the use of intermediary mechanisms such as mediators, wrappers, and ontologies to achieve interoperability. The mapping-based approach attempts to construct mappings between semantically related information sources. It is usually accomplished by constructing a global schema and by establishing mappings between the global schema and the local schemas. Approaches based on intermediaries may rely on mapping knowledge, domain-specific knowledge, or rules established by mapping-based approaches.

Intermediary Mechanisms

1. Wrappers and mediators

Wrappers and mediators [47, 147] provide information manipulation services over a reconciled view of heterogeneous data. Wrappers encapsulate details of each information source, allowing data access under a homogeneous data representation and manipulation style (common data model or standardized schema). Mediators offer an integrated view of the information sets of several information sources associated with corresponding wrappers or other mediators. Some systems adopt multiple levels of mediators in order to modularize the information transformation and integration along successive levels of abstraction.

Two wrappers and one mediator providing integrated access to two different information sources are shown in Fig. 2.1.4. The mediator brokers the requests from the application into requests to the wrappers of the corresponding information sources involved. On receiving the replies from the source wrappers, the mediator composes the results to return an integrated result to the application. Information transformation and mapping specifications may be used to drive the functioning of mediators and wrappers. Wrapper generators and data mapping specification languages enable the specification of data integration in a more intelligible manner than using conventional programming languages to hard code wrappers and mediators [47].

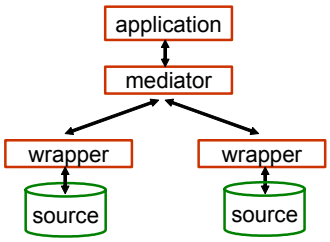


Fig. 2.1.4 Wrappers and mediators

2. Ontology

One of the well accepted mechanisms for achieving semantic interoperability is the utilization of ontologies. Structure knowledge embedded in ontologies supports information retrieval and interoperability [56]. Ontologies also help investigation of correspondences between elements of heterogeneous data sources [5, 12, 94, 95]. Related research proposes the development of information systems components by translating ontologies into object-oriented hierarchies for implementation, giving rise to the concept of ontology-driven information systems [37, 57].

There is a difference in the definition of ontology in the philosophical sense and in the way the term is used in the artificial intelligence (AI) field [57]. In philosophy, ontology is characterized as a particular system of categories reflecting a specific view of the world. In AI, ontology is seen as an engineering artifact that describes a certain reality with a specific vocabulary, using a set of assumptions regarding the intended meaning of the vocabulary words. A terminological distinction between reality-based ontology (R-ontology) and elicited ontology (E-ontology) was suggested in [115]. An R-ontology is a theory about how the whole universe is organized, and corresponds to the philosopher's point of view. An E-ontology, on the other hand, fits the purpose of software engineers and computer and information scientists, and is defined as a theory about how a given individual, group, language, or science conceptualizes a given domain.

In this dissertation, by ontology we mean E-ontology. Ontologies are shared conceptualizations of knowledge about delimited domains [56, 57, 95, 144]. An ontology organizes definitions and interrelationships involving a set of concepts. It captures the meaning of classes and instances from a universe of discourse, by arranging the symbols referring to them, according to semantic relationships.

An ontology embodies a particular viewpoint of a given domain. This viewpoint must be shared by a group of individuals, formed according to factors like geographic proximity, cultural background, profession, interests, or involvement in particular enterprises. These people establish agreements with respect to their views of the world and the symbols used to communicate their views. Ontologies can be explicit or implicit, formal or informal. To be represented and processed by computers, they must be explicit and formal.

There is no convention with respect to the form of a machine-processable ontology. A simple hierarchy specifying classes and their subsumption relationships (e.g., a taxonomy) is an ontology. By specifying relationships and integrity constraints in a database, a relational database schema can be considered as an ontology.

Several languages and formalisms have been proposed to express knowledge in ontologies [49, 56]. OWL (Web Ontology Language) is one of the most prominent ontology languages for the semantic Web. It was extended from RDF [67]. Other ontology languages are described in [35, 49, 111]. The relationship and integration of XML with ontology representation languages and formalisms is addressed in [4, 5, 110, 111]

The development of ontologies is a labor intensive and error prone task if it is done manually. Ontology engineering tools [64, 103, 142] can automate parts of this task and hide the idiosyncrasies of the ontology specification languages and formalisms. Through visual interfaces, those tools help knowledge acquisition, remote access to knowledge repositories, syntax checking, and quality validation. Protégé [103] is an example of an open source graphic tool for ontology editing and knowledge acquisition. Altova SemanticWorks™ [64] is a commercial tool that allows users to graphically create and edit RDF instance documents, RDFS vocabularies, and OWL ontologies with full syntax checking.

Federation, Union Archiving, and Related Projects

Within the intermediary-based approach there are two possible architectures to deal with the problem of integration, namely, federation and union archiving (see Fig. 2.1.2). Federation refers to the case where the DL sends search criteria to multiple remote repositories (e.g., using Z39.50 [88]) and the results are gathered, combined, and presented to the user. Federation is a more expensive mode of operation in terms of network and search system constraints since each repository has to support a complex search language and fast real-time response to queries. Union archiving is based on a mechanism to gather or harvest data from the sources and to load them into a centralized data store. The difficulty of creating large federations is the motivation behind the efforts of union archiving. The underlying concept is that the participants agree to take small efforts that enable some basic shared services.

The libraries that share online catalog records using Z39.50 are an example of federation. Z39.50, a protocol developed by the library community, permits one computer (the client) to

search and retrieve on another (the server). Most of its implementation emphasizes searches that use bibliographic attributes to search MARC catalog records and present them to the client. Though Z39.50 is better suited to federation than union archiving, it can be used to harvest if there is special support at the server end. Another federation is the ADEPT project for geospatial material [62]. Smete.org is an NSDL (National Science Digital Library) project that built a federation among some of the leading collections of education materials.

As for union archiving, there are several paradigms for harvesting data from heterogeneous sites, such as Harvest [17], OAI (Open Archives Initiative) [81], and SRU (Search/Retrieve URL Service) [97]. The underlying concept of Harvest is to divide the principal functions of a centralized search system into separate subsystems consisting of gatherers and brokers. Gatherers collect indexing information from DL collections, while brokers build a combined index of information about many collections. Harvest defined formats and protocols for communication among gatherers and brokers. OAI is a multi-institutional project to address interoperability of DLs. It provides a simple but extensible metadata harvesting protocol, OAI-PMH (the Open Archives Initiative Protocol for Metadata Harvesting), to support the streaming of metadata from one repository to another. It emphasizes the distinction between data providers and service providers. The former administer systems that support the OAI-PMH as a means of exposing metadata. The latter use metadata harvested via the OAI-PMH as a basis for building value-added services [141]. SRU and OAI-PMH are complementary protocols. Both have a goal to harvest metadata from remote sites, but each provides functionality that the other does not. They differ the most when it comes to retrieval. SRU provides a much more granular approach, at the expense of requiring support for a complex CQL (Common Query Language).

CITIDEL (Computing and Information Technology Interactive Digital Educational Library) makes use of union archiving [114]. As part of the NSDL, it uses OAI-PMH to harvest resource metadata from its member collections. A union catalog also is a key component of the Site for Science project [1], whose interoperability strategy was used by the NSDL Core Integration team.

Other efforts that have looked into the issue of interoperability amongst heterogeneous DLs include Dienst [26], InfoBus [105], Fedora [112, 139], and NDLTD [40]. Dienst, the foundation for the original version of NCSTRL (the Networked Computer Science Technical Reference Library) provides for communications with services in a distributed digital library.

The architecture of InfoBus is based on a hardware bus metaphor and was implemented with CORBA distributed object technology. It tried to solve interoperability based on federation and to use high-level descriptions for mapping between different metadata standards. The descriptions are used to manually develop mappings between metadata attributes of these standards. The scope of the work by Fedora [112, 139] is restricted to interoperability for repositories and digital objects. Fedora's approach was strongly influenced by the abstraction of repositories and digital objects [70], by the container architecture described in the Warwick Framework [80], and by the notion of active digital objects [25].

NDLTD (Networked Digital Library of Theses and Dissertations) first explored semantic interoperability by adapting MARIAN [54] as mediation middleware. The MARIAN data model is based on a semantic network of explicit nodes and links organized into a hierarchy of classes in an object-oriented fashion. This hierarchical system helps to join diverse harvested data into a single collection view for the user. More recent work with NDLTD has employed metadata standards (DC, ETD-ms [66], and OAI-PMH).

Mapping-based Approach

While many research projects have developed semantic mediators and wrappers to address interoperability, few have tackled the problem of (partially) automating production of these mediators and wrappers (which contain specific domain knowledge, such as mappings between source schema and the integrated schema) [101, 121, 158]. Schema mapping is typically performed manually, perhaps supported by a graphical user interface, such as Microsoft BizTalk Schema Mapper [65] and Altova Mapforce [63]. Manual schema mapping is a tedious, time-consuming, error-prone, and expensive process. This has spurred numerous solutions to automate the mapping process.

While fully automating the mapping process to automatically generate wrappers is generally infeasible, it is possible to implement techniques that reduce the amount of human interaction.

An implementation of a mapper may involve multiple mapping algorithms, wherein each algorithm computes a mapping based on a single criterion. A hybrid mapper uses multiple mapping criteria, while a composite mapper combines multiple mapping results produced by different mapping algorithms, including hybrid mappers [119]. The ability of composite mappers

to combine different mappers is more flexible than that of hybrid mappers. A hybrid mapper typically uses a hard-wired combination of particular mapping techniques that are executed simultaneously or in a fixed order. However, a composite mapper allows selection from a repository of modular mappers, and has the ability to extend the system when additional mappers are needed. SemInt [86] is an example of a hybrid mapper, while LSD [29] is a composite mapper.

SemInt uses neural networks, clustering, and classification techniques to identify similar attributes from different schemas. It uses a combination of schema and instance information. Schema information includes: data types, field lengths, and constraint information, which are available from the catalog of a relational DBMS. Instance information includes such information as value distributions, character ratios, numeric means, and variances. SemInt exploits up to 15 constraint-based and 5 content-based mapping criteria. For each criterion, a possible numerical value is mapped onto the interval [0, 1]. Then for N mapping criteria, a tuple of N such numerical values for one attribute is the signature of the attribute. Since each signature corresponds to a point in the N-dimensional space, the Euclidian distance between signatures can be used as a measure of the degree of similarity and thus for clustering. Similar attributes of the first input schema are clustered in the same groups. The signatures of the centroids of each cluster then are used as input to train a back-propagation neural network classifier to output an attribute category. Given a new schema, SemInt determines the signature of each schema attribute using the same type of schema and instance information used for training. These signatures are then fed into the neural network to be classified into the categories derived from the clustered attributes of the first schema.

Based on their experiments, the authors found that the straightforward mapping approach based on Euclidian distance does well when finding almost identical attributes, while the neural network is better at identifying less similar attributes that match. SemInt presents a powerful and flexible approach to hybrid mapping, since multiple match criteria can be selected and evaluated together. However, when the attributes of the first schema are clustered, it is difficult to estimate the accuracy of the classifier built later to classify other attributes into the clusters. The clustering step needs to be rather conservative; few clusters containing a large number of attributes are generated to prevent attributes in other schema from being classified into the wrong clusters.

Consequently, a large amount of human evaluation is still needed to select the properly matched attributes from the large cluster.

In case we cannot find the correct mapping from either the schema information or the instance information, using only schema or data instances is potentially inadequate for the mapping process. The LSD system uses machine-learning techniques to automatically combine several mappers and map a new schema against a previously determined global schema. It operates in two phases: training and mapping. Before training, LSD first asks the user to manually specify the mappings, for several sources. Training examples are then created from each source. Different mappers will require different sets of training examples and are trained to output a set of internal classification models. A composite map is trained to know how much weight to give to each individual mapper. During the mapping phase, the trained mappers are used to match new source schemas based on the patterns and rules discovered during training. LSD uses domain constraints and user feedback to eliminate some of the previously determined mapping candidates for improving mapping accuracy.

2.2. Integrated Services

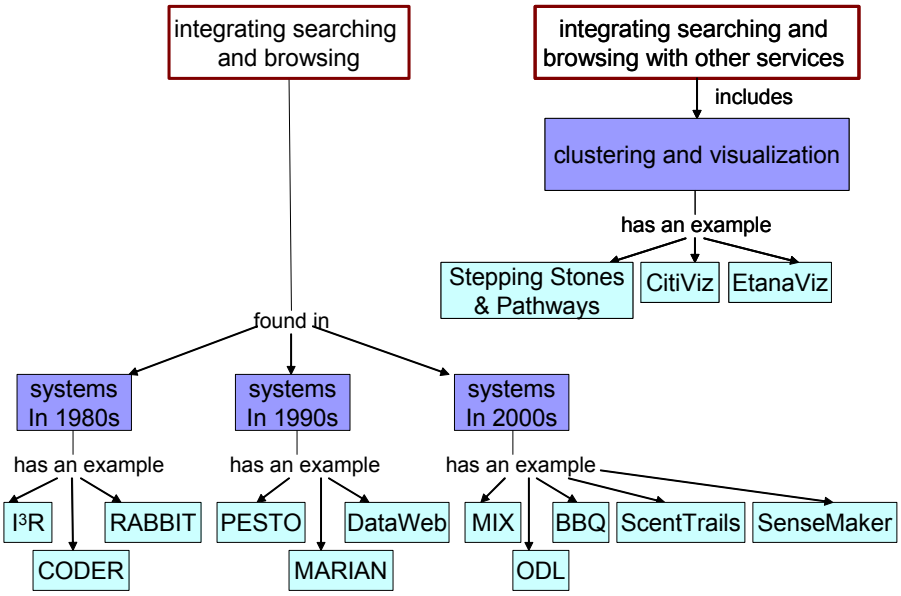


Fig. 2.2.1 Related Work on Integrating Services in DLs

There are some related works on integrating services in DLs as shown in Fig. 2.2.1. Some integrate searching and browsing (see section 2.2.1); while others integrate searching and

browsing with other services (see section 2.2.2). For example, CODER [38], a retrieval and hypertext system using SGML and a lexicon developed in the 1980s, was used as a testbed for the study of artificial intelligence concepts in the field of information retrieval; MARIAN [39], an indexing, search, and retrieval system optimized for digital libraries, was developed in the 1990s; ODL [140], a system built as networks of extended open archives, was developed in the 2000s.

2.2.1. Integrate searching with browsing

A synergy between searching and browsing is required to support user's information-seeking goals [10, 11, 46, 48, 91]. The idea of integrating searching and browsing can be found in some early systems in the 1980s, such as I^3R [23] and RABBIT [148]. Though I^3R had that idea, it did not implement it. While affording compelling browsing experiences, the interface to a database provided by RABBIT is based on the paradigm of 'retrieval by reformulation'.

About ten years after RABBIT and I^3R appeared, searching and browsing integration resurfaced in many efforts, such as PESTO [19] and DataWeb [96]. PESTO integrated browsing and querying via a "query-in-place" paradigm for exploring the contents of object databases. It allowed a user to issue a query relative to the point that her navigation had reached. However, PESTO was not equipped for browsing semi-structured data.

Navigation is the primary mode for DataWeb to interact with the database. DataWeb viewed navigation as a process of query rewriting and query refinement. One can browse or search to attain a different hierarchy at any point while interacting with the DataWeb system. While in this context queries induce hierarchies, there is also an initial set of pre-existing hierarchies available as exemplars for a user to browse prior to querying. Thus, a user may begin an information-seeking activity in the DataWeb system with a query, or browse an extant hierarchy.

Typically, XML data elements are nested, making XML documents conducive to browsing hierarchically. Thus, interactively blending browsing and querying of XML is quite natural. The MIX project [98] provided virtual (i.e., non-materialized) integrated views of distributed XML sources and facilitates the interleaved browsing and querying of the views at both the front-end level and the programmatic level. At the front-end level it provided the BBQ GUI [99], which adopted PESTO's feature of "query-in-place". At the programmatic level MIX provided an API

called QDOM (Querible Document Object Model) supporting interleaved querying and browsing of virtual XML views, specified in an XQuery-like language. The navigation commands are a subset of the navigation commands of the standard DOM API. QDOM allowed an "in-place-query" to be issued from any node in the result of previous queries. The query generates a new "answer" object from which a new series of navigation commands may start.

Though searching and browsing integration were embraced in the database area as shown in some projects mentioned above, the combined paradigm is exhibited by Web users during their information-seeking, and presented in many research efforts such as AMIT [151], WebGlimpse [89], ScentTrails [104], and SenseMaker [8]. AMIT (Animated Multiscale Interactive TreeViewer) is a Java applet that integrates fisheye tree browsing with search and filtering techniques. WebGlimpse allows the search to be limited to a neighborhood of the current document.

ScentTrails annotates the hyperlinks of retrieved Web pages with search cues: indications that a link leads to content that matches the search query. The annotation is done by visually highlighting links to complement the browsing cues (textual or graphical indications of the content reachable via a link) already embedded in each page.

SenseMaker increases the fluidity between browsing and searching DLs by introducing structured-based filtering and structured-based searching. In SenseMaker, a user issues a query and aggregates the retrieved results into bundles by "bundling criterion" (e.g., "same author"). Structured-based filtering allows users to focus on selected bundles and to employ structure to limit a collection of results quickly and at a high level of granularity. The structure-based searching involves growing selected bundles or adding related bundles. Searching by growing selected bundles involves formulating a query that describes the "template" bundles and then issuing that new query. Therefore, the "template" bundles can be viewed as surrogates of queries. Searching by adding related bundles involves identifying the key characteristics of the selected bundles, accessing an external source (e.g., a classification scheme) that records relationships among these characteristics, and issuing a query for items with newly defined characteristics.

Though many research projects have developed different interaction strategies allowing smooth transition between browsing and searching, to the best of our knowledge, none of them

generalize these two predominant exploring services in DLs. In Chapter 6, we show that related works like those above can be viewed as cases that illustrate our theoretical approach.

2.2.2. Integrate searching and browsing with other exploring services

Text mining and visualization techniques provide DLs additional powerful exploring services, with possible beneficial effects on searching and browsing. e.g., Stepping Stones & Pathway (SSP) integrates visualization, clustering, and Bayesian inference to support exploration and the resolution of complex information needs that can be met by sets of related documents [43]. CitiViz, a visual interface to CITIDEL, combines searching, browsing, clustering, and information visualization [71]. As shown in Fig. 2.2.2, it applies two major visualization techniques — a hyperbolic tree [82] of a hierarchical classification system and a 2D scatter-plot graph. It facilitates exploration that can involve rapid switching between searching, browsing in the ACM Computing Classification System, following links between works, and selecting points in a scatter-plot (2D grid) according to any of a number of pairs of facets.

Visualization exploits human vision and spatial cognition to help humans mentally organize and electronically access and manage large, complex information spaces. It has common usage scenarios supporting searching and browsing for DLs. Further, visualization of search results has much in common with gaining an overview of the coverage of a DL to facilitate browsing. Both enable the user to become oriented, and to find relevant information. They differ mainly in two respects. First is the origin of the document sets (a pre-existing static collection, or result set dynamically retrieved from a search engine). Second is the information available that relates documents to user information needs.

Thus, first, we consider visualization supporting browsing, i.e., provide a starting point for users by presenting overviews of a DL collection. Graphic overviews can display category labels hierarchically based on the facets. Categories can be visualized as a hyperbolic tree [82] or a SpaceTree [116] as well as through a traditional node-link representation of a tree. One example of a graphic overview is Map.net (http://map.net/start). It provides hierarchical (multilevel/categorical) information maps for browsing over two million Web sites from the Open Directory Project (http://dmoz.com). Rather than using conventional search engine technology to navigate the Web, it creates a landscape that spatially represents data relationships, though in a very abstract, geometric fashion. Size and position of areas on the map indicate

numbers of documents in respective categories and mutual relations between them (see Fig. 2.2.3). Users of this kind of interface gain an immediate overview of available categories and the number of documents these categories contain.

Some research has been conducted on how to present an overview of retrieved results. Envision [41] organized search results according to metadata along the X and Y-axes, and showed values for attributes associated with retrieved documents within each cell. Cat-a-Cone used ConeTree [126] to display the category labels of the documents retrieved, while the retrieved documents are organized as pages in a WebBook [18]. Hieraxes [138], in combination with a grid display, offer a simple approach to searching result sets by using categorical and hierarchical axes. Users can see an overview by color-coded dots or bar charts arranged in a grid and organized by familiar labeled categories. They can probe further by zooming in on desired categories or switching to another hierarchical variable. The RB++ [155] categorizes the collection offline and uses a uniform category structure to present overviews of the collection and the retrieval results. It provides visualized category overviews of an information space and allows dynamic filtering and exploration of the result set by tightly coupling the browsing and searching functions.

Via the use of text mining, an overview of a DL collection or searching results can be automatically derived, often through document clustering or neural networks. Examples are Scatter/Gather [24, 61], Grouper [154], Galaxy of News [125], Vivisimo (http://vivisimo.com), Kartoo (http://kartoo.com), Highlight (http://highlight.njit.edu/technology.htm), SOM [74], ThemeScapes [150], and Mooter (http://mooter.com:8080/moot).

Grouper was a dynamic clustering interface to web search results. It introduced the Suffix Tree Clustering (STC) algorithm. Vivisimo is a web search clustering interface. Its algorithm is based on an old artificial intelligence idea: a good cluster or document grouping is one that possesses a good, readable description. Kartoo is a web interface organizing search results retrieved from relevant web search engines by topics, that displays them on a 2-dimensional map. Theoretically, Kartoo provides a node-link graph. A document (Web page) node is presented by a ball. The size of the ball corresponds to the relevance of the document to the query. Links are labeled with sets of keywords shared by related documents. SOM is a neural network algorithm that takes a set of high-dimensional data and maps them onto nodes in a 2D grid. Shifting to 3D, the ThemeScapes view imposes a three-dimensional representation on the

results of clustering. The layout makes use of "negative space" to help emphasize the areas of concentration where the clusters occur.

Regarding document clustering, there are two major types of algorithms, document-oriented and concept-oriented. Document-oriented algorithms determine names for clusters after clustering documents, while concept-oriented algorithms use concepts (words or phrases) in the documents as names of clusters to form a structure before documents are clustered according to the names.

1. Document-oriented clustering

Many document-oriented clustering algorithms recursively use partitioning methods to create hierarchies in a top down approach. K-means is a classic divisive clustering algorithm. It starts with a random set of k centroids and assigns each object to its closest centroid. It iteratively optimizes assignment of objects to clusters until a certain amount of time elapses or no object reassignment is necessary. Its main advantage is its linear time complexity, $O(nkT)$, where n is the size of the document collection, k is the number of desired clusters, and T is the number of iterations. Its chief disadvantages are: necessity of knowing the number of clusters in advance; no specified way of initializing centroids; and ineffectiveness when desired clusters are not spherical. K-means assumes that each cluster in a document corpus has a compact shape. This assumption does not often hold true, and document clustering results could be terribly wrong because of a broken assumption [154]. Bisecting K-means is a basic version of K-means. It starts with 2 centroids which are symmetric to the mean of the document set.

Another divisive algorithm is PDDP [133]. It belongs to the class of Singular Value Decomposition (SVD)-based clustering algorithms such as Latent Semantic Indexing (LSI). The inputs of LSI and PDDP are different. The former uses the original term-document matrix, while the latter uses the auxiliary matrix (centralized matrix) of the term-document matrix. PDDP splits the matrix with a hyper-plane passing through its centroid. Since PDDP only use the first vector of the right singular vectors of input matrix, its computational demand is less than LSI. The disadvantage of PDDP is that it prunes off substantial information since it only keeps one principle direction for clustering.

2. Concept-Oriented Clustering Algorithm

Lawrie introduced two concepts, topicality and productiveness, to select terms for creating topic hierarchies [85]. She used Kullbck-Leibler divergence to estimate term topicality and a Dominate model to estimate productiveness [85]. The Dominate model is a statistical language model using a greedy approximation to the Dominated Set Problem. It captures main characters observed in the subsumption and lexical hierarchies. A subsumption hierarchy was described in [131]. Subsumption defines a vocabulary term, *v*, to be a dependent of a topic, *t*, when $P(t|v) \geq 0.8$. The lexical hierarchy requires that terms occur within the same adjective-noun compound [102]. [77] proposed a hierarchical monothetic document clustering algorithm for summarization and browsing search results. It tried to maximize the distinctiveness of the monothetic features describing the clusters while at the same time maximizing the number of documents that can be described or covered by the monothetic features.

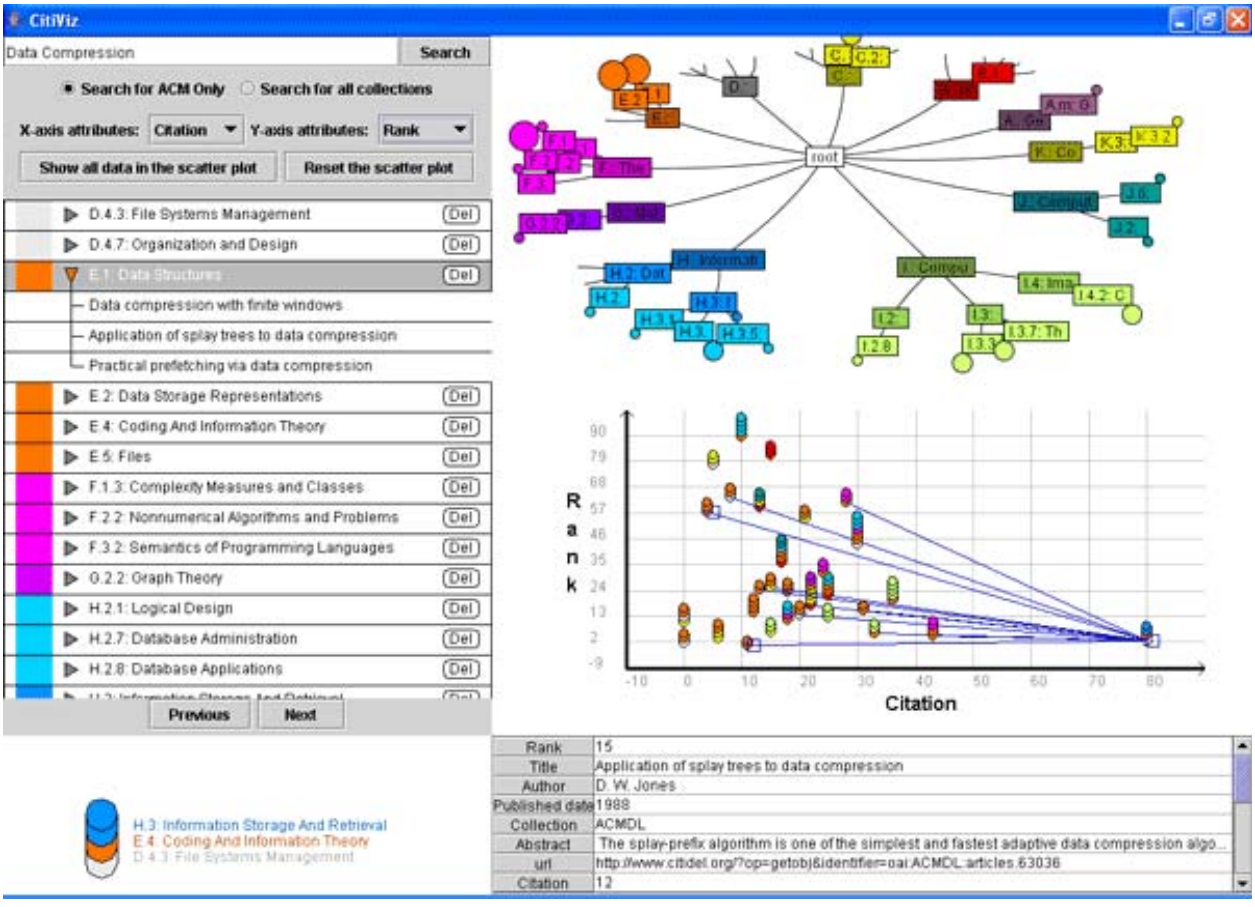


Fig. 2.2.2 A screenshot of CitiViz

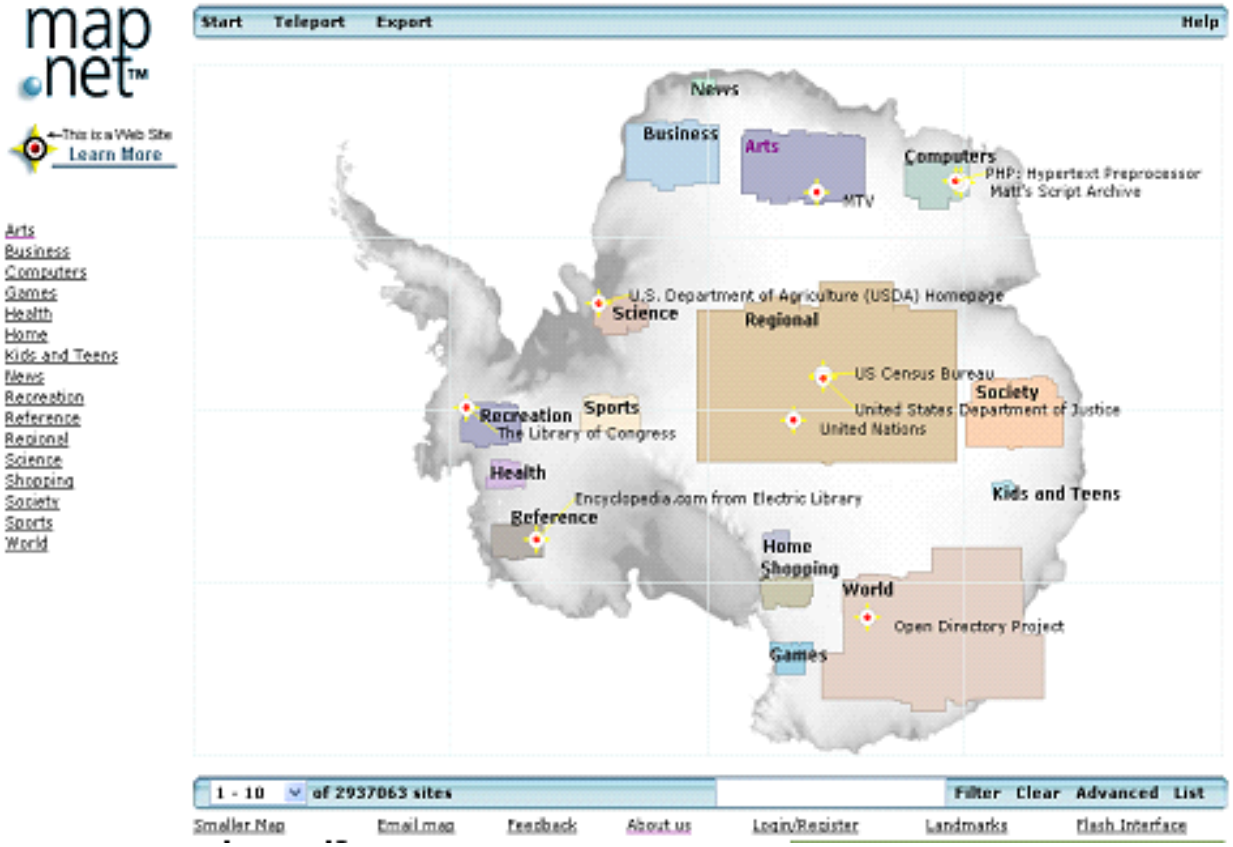


Fig. 2.2.3 map.net's overview of Open Directory Project

Chapter 3. Modeling of Domain Specific Digital Libraries with the 5S Framework

Requirements gathering and conceptual modeling are essential for the customization of digital libraries (DLs), to help attend the needs of target communities. In this chapter, we show how to apply the 5S (Streams, Structures, Spaces, Scenarios, and Societies) formal framework to support both tasks. The intuitive nature of the framework allows for easy and systematic requirements analysis, while its formal nature ensures the precision and correctness required for semi-automatic DL generation. Further, we show how 5S can help us define a domain-specific DL metamodel in the field of archaeology. An archaeological DL case study (from the ETANA project) then yields informal and formal descriptions of two DL models (instances of the metamodel). Finally, we illustrate the use of the 5SGraph tool to specify archaeological DLs.

3.1. Introduction

The construction of any digital library involves a number of decisions covering: 1) which types of multimedia content will be supported by the DL; 2) how the stored information is organized and structured; 3) which are the target communities; and 4) which services and capabilities will be provided [50]. The process of formally assembling such decisions and representing them in a format useful for processing by a DL system involves both requirements gathering and analytical modeling or design.

Modern software engineering has encouraged the use of formal methods, with mathematically defined syntax and semantics, to support such tasks. Formal methods and frameworks can support specification of (most of the parts of) complex systems such as DLs, while also promoting rigor and correctness. This chapter focuses on the application of the 5S formal framework in the support of these tasks. More specifically, we show how 5S can help us document complex requirements and can support the modeling of domain-specific digital libraries, illustrated with a case study from the field of archaeology.

The rest of this chapter is organized as follows. Section 2 informally discusses requirements of archaeological DLs according to 5S. Section 3 builds on the prior section to

present a formal archaeological DL metamodel. Section 4 presents a two-part case study illustrating the methodology and models. Section 5 concludes the chapter and outlines future work.

3.2. Archaeological Digital Libraries: a 5S-Based Informal View

This section shows how 5S can be used to analyze the requirements of domain-specific DLs. More specifically, it informally describes the archaeological domain, and therefore archaeological DLs (ArchDLs), in the light of the 5S framework. Some work presented in this section is derived from part of the requirements analysis for ETANA-DL, i.e., email interviews with five prestigious archaeologists and face to face workplace interviews with eleven archaeologists (including three of the five interviewed by email) conducted by the previous PI of the ETANA-DL project, and the Head of Digital Library Initiatives at Case Western University Reserve University.

1. Societies

Societies can be groups of humans as well as hardware and software components. Examples of human societies in ArchDLs include archaeologists (in academic institutes, fieldwork settings, excavation units, or local / national government bodies), the general public (e.g., educators, learners), and those who lived in historic and prehistoric societies. There also are societies of project directors, field staff (responsible for the work of excavation), technical staff (e.g., photographers, technical illustrators, and their assistants), and camp staff (including camp managers, registrars, and tool stewards). Since archaeology is a multi-disciplinary subject, drawing on a wide range of skills and specialties, from the arts and humanities to the biological and physical sciences, societies of specialists (e.g., in geology, anthropology, lithics, ceramics, faunal and floral remains, remote sensing) are involved in ArchDLs. Societies follow certain rules and their members play particular roles. Members of societies have activities and relationships (e.g., specialists serve to assist and advise the varying field and laboratory staffs regarding field problems and other matters related to their special skills and interests). Because archaeologists in diverse countries follow different laws and customs, a number of ethical and freedom-related issues arise in connection with ArchDLs. Examples include: Who owns the finds? Where should they be preserved? What nationality and ethnicity do they represent? Who

has publication rights? To address these issues, and to support the variety of needs of interested societies, DL designers have planned for numerous scenarios.

2. Scenarios

A scenario is often defined as a description of interactions between a human user and a computerized system. Scenarios also can describe interactions among software modules (as in [53]) or among humans. Further, describing scientific processes (hypothesizing, observing, recording, testing, analyzing, and drawing conclusions – used during any archaeological study) as scenarios can help with comprehending specific ArchDL phenomena, and with requirements elicitation and specification generation.

Digital recording as an archaeological process to facilitate information gathering occurs in two stages, the planning stage and the excavation stage. Remote sensing, fieldwalking, field surveys, building surveys, consulting historical and other documentary sources, and managing the sites and monuments (and related records) maintained by local and national government bodies may be involved in the planning stage. During excavation, detailed information is recorded, including for each layer of soil, and for features such as pole holes, pits, and ditches. Data about each artifact is recorded together with information about its exact find location. Numerous environmental and other samples are taken for laboratory analysis, and the location and purpose of each is carefully recorded. Large numbers of photographs are taken, both general views of the progress of excavation and detailed shots showing the contexts of finds. Since excavation is a destructive process, this makes it imperative that the recording methods are both accurate and reliable. Unlike many other applications of information systems, it simply is not possible to go back and re-check at a later date [127]. Large quantities of archaeological data generated during the above-mentioned two stages can be harvested by ArchDLs, organized, and stored to be available to researchers outside a project (site), without substantial delay. After excavation, information stored in ArchDLs is analyzed, and helps archaeologists to test hypotheses. For example, if archaeologists retrieve records of corn artifacts from an ArchDL, they might hypothesize that the former residents were farmers, and test their hypothesis with soil sample data using statistical analysis tools provided by the ArchDL. This hypothesis is a scenario involving archaeologists, the historical community (farmers), and finds (corn samples). Other hypotheses are scenarios describing relationships among historical communities. For example, if

there are large collections of jars of the same style found in two nearby sites, archaeologists might hypothesize that people in these two sites (cities) used the jars to carry things in commercial trade. Thus, primary archaeological data, managed with powerful tools in ArchDLs, help archaeologists find physical relationships between excavation contexts, develop a structural history of a site, and extend the understanding of past material cultures and environments in the area. Data generated from the sites' interpretation then provide a basis for future work including publication, museum displays, and, in due course, input into future project planning.

Besides supporting archaeologists in their work as described above, ArchDLs provide services for the general public. A student interested in a Near Eastern site can access all the archaeological information about it by browsing or using complex retrieval criteria that take into account both intrinsic attributes of items and their extrinsic spatial and temporal interrelationships. Further, she can view the information organized in a spatial hierarchy / map that facilitates navigation among archaeological items at various spatial scales. She can click on items to show details; to display photographs, maps, diagrams, or textual documents; or to jump to other items.

3. Spaces

One important spatial aspect of ArchDLs is the geographic distribution of found artifacts, which are located in a 4D spatial continua, the fourth dimension being the temporal (as inferred by the archaeologists). Metric or vector spaces are used to support retrieval operations, calculate distances, and constrain searches spatially. Other space-related aspects deal with user interfaces or with 3D models of the past.

4. Structures

Structures represent the way archaeological information is organized along several dimensions. Archaeological information is spatially organized, temporally sequenced, and highly variable. Examples include site organization, temporal order, and taxonomies of specific unearthed artifacts like bones and seeds. The structures of sites present, simply and consistently, the basic spatial containment relationship at every level of detail, from the broadest region of archaeological interest to the smallest aspect of an individual find. Generally, specific regions are subdivided into sites, normally administered and excavated by different groups. Each site is

further subdivided into partitions, sub-partitions, and loci, the latter being the nucleus of the excavation. Materials or artifacts found in different loci are organized in containers for further reference and analysis. The locus is the elementary volume unit used for establishing archaeological relationships. Archaeological relationships between loci are from both the vertical and horizontal points of view. The first is given by reference to loci above and below a given locus, the second by coexisting loci (loci located at the same level). The archaeological relationship is related to the temporal succession of various events of construction, deposition, and destruction. Temporal sequencing of archaeological items involves linking items to form a stratigraphic diagram of the kind developed in the 1970s by Edward Harris (http://www.harrismatrix.com/) and now used by many archaeologists. A "Harris Matrix" is a compact diagram representing the essential stratigraphic relationships among all the items; it shows the chronological relationship between excavated layers and contexts. In general, if two layers are in contact with each other and one lies over the other, then the upper layer is chronologically later. This is the basis on which the structural history of a site is founded. The construction of this diagram and its subsequent use in the interpretation of structural phases is central to both the understanding of the site during excavation and to the post-excavation analysis [36]. Spatial and stratigraphic relationships among archaeological items can be regarded as extrinsic attributes (inter-item relationships) [134]; intrinsic attributes are those describing the items themselves. Finally, since archaeological information is highly variable, items observed in a typical excavation may fall into a wide variety of different classification systems, and may exhibit many idiosyncrasies.

5. Streams

In the archaeological setting, streams represent the enormous amount of dynamic multimedia information generated in the processes of planning, excavating, analyzing, and publishing. Examples include photos and drawings of excavation sites, loci, or unearthed artifacts; audio and video recordings of excavation activities; textual reports; and 3D models used to reconstruct and visualize archaeological ruins.

3.3. A 5S-Based Metamodel for Archaeological Digital Libraries

With key requirements for ArchDL summarized in the previous section, we can proceed to constructively define a minimal ArchDL metamodel. A domain-specific metamodel is a generic model which captures aspects specific to the domain at hand. We build upon the definition of a minimal DL as formally defined in [53] and extend it with concepts specific to the archaeology domain. Following our minimalist approach, we only define essential concepts without which we think a DL cannot be considered an ArchDL. The concepts and definitions are illustrated in Fig. 3.3.1, where each concept is enclosed in a box labeled with the number of its formal definition (A. 1- A.12 are also shown in Fig. 1.3.1 and defined in [53]; 1-10 are defined below). The main extensions concern the fact that: 1) most archaeological digital objects are surrogates of real-world artifacts; and 2) these artifacts are found within a social-temporal-spatial context.

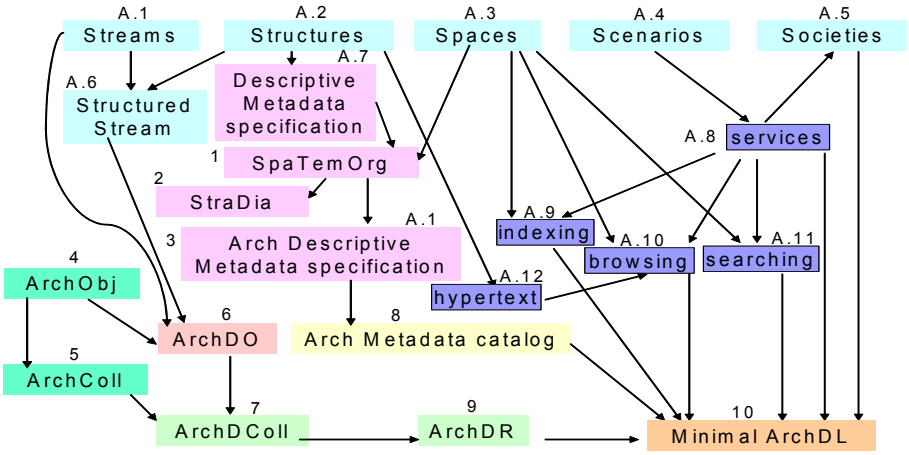


Fig. 3.3.1 Minimal archaeological DL in the 5S framework

Notation: Let $L=\cup D_k$ be a set of literals defined as the union of domains D_k of simple data types (e.g., strings, numbers, dates, etc.). Let also R represent sets of labels for resources. Let *SpaP1* be a tree with a vertex set $V_{SpaP1}= \{v_i \mid i=1,2,...,7\}$; an edge set $E_{SpaP1}=\{(v_i, v_{i+1}) \mid i=1,2,...,6\}$; a labeling function $F_{edge1}: E_{SpaP1} \rightarrow L_{SpaP1}=\{\text{'contains'}\}$; and a labeling function $F_{node1}: V_{SpaP1} \rightarrow L$. Let *SpaP2* be a set: *SpaP2= {'above', 'below', 'coexisting with'}*. Let *TemP* be a tree with a vertex set $\{u_1, u_2\}$; an edge set $E_{TemP}=\{(u_1, u_2)\}$; a labeling function $F_{edge2}: \{u_1, u_2\} \rightarrow L_{TemP}=\{\text{'detailed by'}\}$; a labeling function $F_{node2}: \{u_i \mid i=1,2\} \rightarrow V_{TemP}=\{F_{node2}(u_i)\}$, where $F_{node2}(u_1)=$'*period*', and $F_{node2}(u_2)=$'*chronology*'.

Definition 1: A ***Spatial Temporal Organization (SpaTemOrg)*** is a descriptive metadata specification (see Def. 12 in [53] for details; see also Fig. 3.3.2), ***SpaTemOrg*** = *((V, E), R∪L, F)*, such that $\forall e=(u,v)\in E$, where $u,v\in V$, *F(u)* ∈ *R∪L*, *F(v)* ∈ *R∪L*, *F(e)* ∈ V_{SpaP1} ∪ V_{TemP} ∪ *SpaP2*.

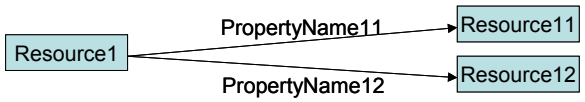


Fig. 3.3.2 Descriptive metadata specification for 'Resource1'

Example 1.1: Given *u*, v_1, v_2 ∈ *V*, *F(u)= 'Bone1'*, $F(v_1)$= *'Jordan Valley'*, $F(v_2)$=*'Nimrin'*, *x=F((u,* v_1*))='region'*, *y=F((u,* v_2*))='site'*, F_{SpaP1}*((x, y))='contains'*, expression *('Bone1', (region:'Jordan valley'), (site:'Nimrin'))* means *'Bone1'* was excavated from the Jordan valley, which contains the Nimrin site (see Fig. 3.3.3).

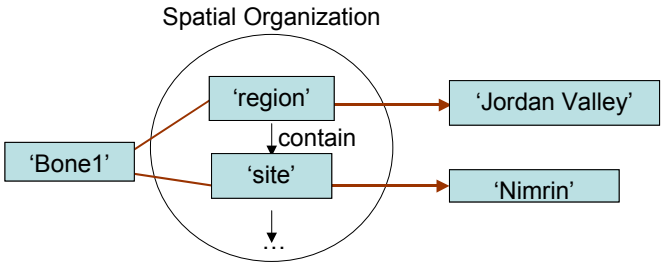


Fig. 3.3.3 Example 1.1 of SpaTemOrg

Example 1.2: Given *u*, v_1, v_2 ∈ *V*, *F(u)= 'Bone1'*, $F(v_1)$= *'Middle Bronze'*, $F(v_2)$= *'2000B.C. – 1500B.C.'*, *x=F((u,* v_1*))= 'period'*, *y=F((u,* v_2*))= 'chronology'*, F_{TemP}*((x, y))= 'detailed by'*, expression *('Bone1', (period: 'Middle Bronze'), (chronology: '2000B.C. – 1500B.C.'))* means *'Bone1'* was excavated from a deposit made in the Middle Bronze age, which has range 2000B.C. – 1500B.C. (see Fig. 3.3.4).

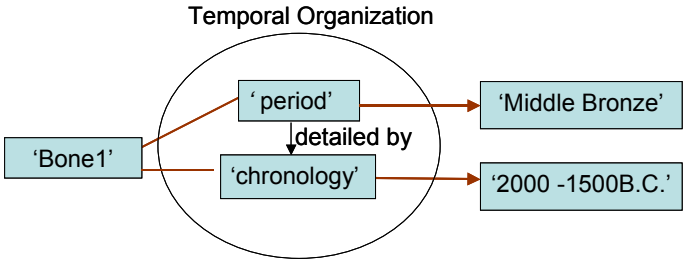


Fig. 3.3.4 Example 1.2 of SpaTemOrg

Definition 2: A ***Stratigraphic Diagram (StraDia)*** is a ***SpaTemOrg*** with a labeling function F_{stra}: $(V \cup E) \rightarrow L_{stra}$={*‘contemporary with’, ‘later than’*}, and two relations, $\cong$ and $\leq$, defined on V:

1) $\cong$ is: a) reflexive, b) symmetric, and c) transitive

2) $\leq$ is: a) reflexive, b) anti-symmetric, and c) transitive

such that $F_{stra}(\cong)$ = *‘contemporary with’*, $F_{stra}(\leq)$ = *‘later than’*, $\forall e = (u,v) \in E$, where $u, v \in V$, $F(u) \in R \cup L$, $F(v) \in R \cup L$, and $F(e) \in L_{stra} \cup SpaP2$.

Example 2.1: Given $u, v \in V$, $F(u)$= *‘locus1’*, and $F(v)$= *‘locus2’*, $F_{stra}((u,v))= F_{stra}(\leq)$ = *‘later than’*, $F((u,v))$= *‘above’* $\in SpaP2$, expression *(‘locus1’ ≤ ‘locus2’)* means that *locus1* was later than *locus2*; and expression *(‘locus1’, above, ‘locus2’)* means that *locus1* was above *locus2* (see Fig. 3.3.5).

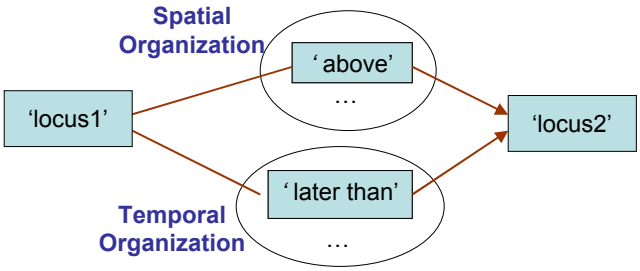


Fig. 3.3.5 Example 2.1 of StraDia

Definition 3: An ***Archaeology Descriptive Metadata specification***: $Arch\_dm \in \{SpaTemOrg\}$.

Definition 4: An ***Archaeology Object in the real world (ArchObj)*** is a unit of observation generated by an archaeological activity (e.g., an archaeological town site, tomb, skeletal material, pottery, etc.).

Definition 5: An ***Archaeology Collection (ArchColl)*** is a tuple: $ArchColl=(h_{ArchColl}, \{ArchObjc\})$, where $h_{ArchColl} \in H$, and H is a set of universally unique handles; $\{ArchObjc\}$ is a set of archaeology objects in the real world.

Definition 6: An ***Archaeology Digital Object (ArchDO)*** is a tuple: $ArchDO=(h, SM, ST, StructuredStreams, SurrogateObj)$, where

1) $h \in H$, where H is a set of universally unique handles;

2) $SM = \{sm_1, sm_2, ..., sm_n\}$ is a set of streams;

3) $ST = \{st_1, st_2, ..., st_m\}$ is a set of structural metadata specifications;

4) $StructuredStreams = \{stsm_1, stsm_2, ..., stsm_p\}$ is a set of *StructuredStream* functions defined from the streams in the *SM* set and the structures in the *ST* set.

5) *SurrogateObj:* a function $\{h\} \rightarrow \{ArchObj_1, ArchObj_2, ..., ArchObj_k\}$ maps a handle h to an archaeology object in the real world, *SurrogateObj(h)*.

Definition 7: An ***Archaeology Digital Collection (ArchDColl)*** is a tuple: $ArchDColl=(h_{ArchDColl}, \{ArchDO\}, SurrogateColl)$, where $h_{ArchDColl} \in H$; H is a set of universally unique handles; $\{ArchDO\}$ is a set of archaeology digital objects with handles in H. Let $Coll \in 2^{\{SurrogateObj(h)\}}$, where h is the handle of *ArchDO*; *SurrogateColl* is a function $\{h_{ArchDColl}\} \rightarrow Coll$ that maps handle $h_{ArchDColl}$ to a real world archaeology collection. Fig. 3.3.6 illustrates functions *SurrogateObj* and *SurrogateColl*.

Definition 8: An ***Archaeology metadata catalog ($ArchDM_{ArchDColl}$)*** for an ArchDL collection *ArchDColl* is a set of pairs $\{(h, \{Arch\_dm_1, Arch\_dm_2, ..., Arch\_dm_i, ...\})\}$, where $h \in H$ and each $Arch\_dm_i$ is an archaeology descriptive metadata specification.

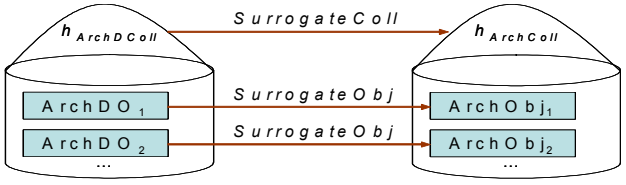


Fig. 3.3.6 Functions *SurrogateObj* and *SurrogateColl*

Definition 9: Let DColl $=\{ArchDO_1, ArchDO_2, ..., ArchDO_k\}$ with k handles in H. An ***Archaeology Digital Repository (ArchDR)*** is a tuple: $ArchDR=(Re, get, store, del)$, where $Re \subseteq 2^{DColl}$, and "get", "store", and "del" are functions over the repository (see Def. 19 in [53] for details on these functions).

Definition 10: An **Archaeological *Digital Library (ArchDL)*** is a tuple: $ArchDL=(ArchDR, ArchDM, Serv, Soc)$, where

1) $ArchDR$ is an archaeology digital repository;

2) $ArchDM=\{ArchDM_{ArchDColl_1}, ArchDM_{ArchDColl_2}, ..., ArchDM_{ArchDColl_k}\}$ is a set of archaeology metadata catalogs for all archaeology digital collections $\{ArchDColl_1, ArchDColl_2, ... ArchDColl_k\}$ in the repository;

3) $Serv$ is a set of services containing at least indexing, searching, and browsing;

4) $Soc= (SM \cup Ac, R)$, where SM is a set of service managers responsible for running DL services, $Ac \subseteq \{Archaeologist, GeneralPublic\}$ is a set of actors that use those services, and R is a set of relationships among $SM \cup Ac$.

3.4. Case Studies: Application of 5S to Archaeological DLs

In the last two sections, 5S was used to provide both an informal and a formal ArchDL model. In this section we use two archaeological information systems of ETANA projects (http://www.etana.org/) as case studies to 1) show the use of 5S as an analytical tool helpful in comprehending specific ArchDL phenomena; and 2) illustrate the use of 5S for requirements gathering and modeling in ArchDL development. Data contributed by ETANA projects to ETANA-DL (http://etana.dlib.vt.edu) are described at http://etana.dlib.vt.edu:8080/etana/htmlPages/etanadl_collections.htm.

3.4.1. Virtual Nimrin

Tell Nimrin (TN) is an archaeological town site at Shuna South, Jordan, north of the Dead Sea, in the Jordan Valley. The digital presentation of TN, virtual Nimrin (VN, http://www.cwru.edu/affil/nimrin/), at Case Western Reserve University, is supervised by director James W. Flanagan.

1. Societies

VN was designed for the general public as well as research specialists. Other communities addressed include: directors, core field staff (square supervisors, technical archaeologists, disciplinary specialists, assistant staff, and managers), and VN website developers/viewers from a score of museums, research institutions, colleges, and universities in Jordan and the United States.

2. Scenarios

Each of the communities involved in the VN society is engaged in various tasks and activities. Core field staffs were responsible for the actual work of excavation and recording. For example, in the field, unearthed bones were bagged separately, daily, with a feature and field specimen number which could be cross referenced with associated ceramics. These bagged bones were transported to field laboratories to be dry brushed, washed when necessary, and separated into generalized categories such as large, medium, or small mammals; fish; birds; etc. To advance and enhance digital recording, digital photography and additional programming were used. Project directors pursued geological and archaeological research by analyzing the field survey and excavated record, testing hypotheses, and publishing preliminary and final reports. For example, they found there was a reduction in percentages of bones of hogs over time at Tell Nimrin and hypothesize that the reason probably was the introduction of religious taboos against eating pork. VN website developers built systems to allow users to interact with and interpret the site without being constrained by the director's view. General users may be interested in taking a tour of the VN website and in viewing museum quality artifacts and major finds, while specialists may want to interact with or download the databases created from the field records of excavation.

3. Spaces

TN's geographical setting marks the intersection of the N-S and E-W arteries in the Jordan Valley approximately 12km north of the Dead Sea and 8km from the Jordan River due east of Jericho. It is at $35^{o}37'30''$ east longitude and $31^{o}54'00''$ north latitude with a Palestinian grid reference of 2094E/1451N. The mound stands 12.75m high on the south bank of Wadi Nimrin, with base 200m below sea level. GPS was used in a geological survey, such as to document the regression

of the ancient freshwater Lake Lisan that once filled the Jordan Valley, and to determine how the change from a freshwater to a saline body affected the pattern of ancient settlements in the region. Other space aspects of VN are TN's coordinate system (site grid and identification of squares) displayed in the topographical drawing, and VN's user interface.

4. Structures

Structures of VN include its relational database, TN's site organization, and TN's stratigraphic diagram, from which a temporal sequence was derived. Spatial and temporal description of records in the database is specified according to TN's polar point grid site organization and site chronological order. The 00/00 point was set at the highest elevation of the mound which was central to its N/S and E/W expanse as well. From there, the site was divided into quadrants, which were subdivided into 5m × 5m squares, each labeled according to the point closest to 00/00. For instance, N40/W20 identified both the point 40m north and 20m west of 00/00 and the five meter square to the north and west of that point. Stratigraphical relationship analysis has identified eight major strata. They are: Modern (stratum VIII), Mamluk (stratum VII), Late Byzantine/Umayyad (stratum VI), Roman/Byzantine (stratum V), Persian (stratum IV), Iron II (stratum III), Iron I (stratum II), and Middle Bronze (stratum I). They clarified TN's long history as an agricultural town site and indicated TN was a substantial settlement, inhabited continuously for the past 4 millennia, except for a 500 year period.

5. Streams

VN deals with various streams, such as drawings and photos of (parts of) TN, publications of preliminary (final) reports, and tuples of primary data in the database.

Virtual Nimrin (VN) Formal Model:

Virtual Nimrin is a tuple: VN = *(VN_R, VN_DM, VN_Serv, VN_Soc),* where

1) *VN_R* is an archaeological digital repository having Tell Nimrin's digital collections of animal bones, seeds, etc.

2) $VN\_DM=\{VN\_DM_{DCollObj}\}$ is a set of archaeology metadata catalogs for all archaeology digital collections in VN, where $VN\_DM_{DCollObj}$ is a metadata catalog for digital collection

DCollObj. Let $VN\_dm_{DCollObj}$ be a descriptive metadata specification for digital objects in *DCollObj*. $VN\_dm_{DCollObj} \in \{SpaTemOrg\} \cup \{VN\_Dobj\}$, where

a) *{Jordan Valley', 'Nimrin', 'quadrant', 'square', 'locus', 'bag'}* $\cup$ *{'Ottoman-Modern', 'Islamic', 'Byzantine', 'Late Hellenistic-Roman', 'Persian', 'Iron II', 'Iron I', 'Middle Bronze'}* $\subset L$. (See examples in Fig. 3.3.3 and Fig. 3.3.4 for reference.)

b) *VN_Dobj= ((V, E), R, F)* is an archaeological-object-specific descriptive metadata specification. If *DCollObj* is a digital collection of animal bones, then $\forall e = (u, v) \in E$, where $u \in V$, $v \in V$, $F(u) \in R$, $F(v) \in L$, and $F(e) \in$ *{'boneName', 'animalName'}*.

3) *VN_Serv={browsing, searching}*

4) *VN_Soc= ({VN_ServiceManager}* $\cup$ *{VN_Actor}, R)* $\cup$ *{HistoricNimrinResident}* $\cup$ *{PrehistoricNimrinResident}, R)*

where *{director, fieldStaff, specialist, student,...}* $\subset$ *{VN_Actor}*, *{VN_BrowseManager, VN_SearchManager}* $\subset$ *{VN_ServiceManager}*, and *R* is a set of relationships among *{VN_Actor}* and *{VN_ServiceManager}*, e.g., *browse=(student* $\times$ *VN_BrowseManager, browsing)*, *search=(specialist* $\times$ *VN_SearchManager, searching)*, and *guide=(director* $\times$ *fieldStaff,* Φ*)* $\in R$.

3.4.2. Halif DigMaster

The site of Tell Halif, located in southern Israel, is the focus of the Lahav Research Project. Halif DigMaster (HD) is an online archaeological database that offers access to a collection of Persian/Classical (and some Iron II Age) figurines recovered in excavation from Tell Halif (TH).

1. Societies

HD was developed to disseminate archaeological information to the academy and to the public. Societies of HD include the communities who excavated the figurines from Tell Halif, provided HD with a preliminary presentation of the excavated material, collaborated with HD on resource sharing, or developed and/or employed HD.

2. Scenarios

Scenarios of HD can be those related to activities such early publication, collaborative publication, 3D publication, and other usage scenarios of HD. The overlong lag between

discovery and publication is an embarrassment for archaeology of the ancient Near East. To address this embarrassment, staff of the Lahav Research Project made graphic forms of excavated objects available to the scholarly and professional world prior to final publication, while field work was still continuing. Scenarios of inviting excavators with non-public or incompletely published collections to add their materials to HD allow several excavation teams to share resources. Scenarios of using 3D technology such VRML (Virtual Reality Modeling Language) overcome the limitations of "flat" photographs on screen. Usage scenarios of HD describe services such as browsing and searching the HD database.

3. Spaces

Major spaces in HD are the geographical setting of Tell Halif (located at $34^{o}52'$ east longitude and $31^{o}23'$ north latitude, at coordinates 1373/0879 on the Palestinian grid) and HD's user interface. Another space is of the VRML models of artifacts.

4. Structures

Structures in HD include the relational database, Tell Halif's site organization, and TH's strata relationship. TH has been surveyed and plotted in relationship to the standard regional grid. Each of the major sections being worked is called a "field", which is further divided into a number of more or less standard $5\text{m} \times 5\text{m}$ areas. TH consists of seventeen major occupation strata, one built atop another to a depth of more than six meters. Those strata revealed that TH has a history of occupation began in the Chalcolithic era (3500B.C. – 3200B.C.) down to the modern settlement of Kibbutz Lahav (founded in 1963 A.D.).

5. Streams

Streams in HD are full-scale color photographs, ink drawings, QuickTime VR movies, plans and soil profiles, as well as summary reports for excavation units.

Halif DigMaster (HD) Formal Model

Halif DigMaster is a tuple: HD = *(HD_R, HD_DM, HD_Serv, HD_Soc),* where

1) *HD_R* is an archaeological digital repository having Tell Halif's digital collection of figurines, denoted as *DCollFig.*

2) $HD\_DM=\{HD\_DM_{DCollFig}\}$ is a set of archaeology metadata catalogs for the ArchDL collections of figurines, where $HD\_DM_{DCollFig}$ is a metadata catalog for digital collection *DCollFig*. Let $HD\_dm_{DCollFig}$ be a descriptive metadata specification for digital objects in *DCollFig*. $HD\_dm_{DCollFig} \in \{SpaTemOrg\}$, where
$\{$'Southern Israel', 'Halif', 'field', 'area', 'locus', 'basket'$\} \cup \{$'Persian'$\} \subset L$.

3) $HD\_Serv=\{browsing, searching\}$

4) $HD\_Soc= (\{HD\_ServiceManager\} \cup \{HD\_Actor\} \cup \{PersianHalif\}, R)$, where $\{director, fieldStaff, specialist, student\} \subset \{HD\_Actor\}$, $\{HD\_BrowseManager, HD\_SearchManager\} \subset \{HD\_ServiceManager\}$. We denote the community that made the Persian figurines excavated from Tell Halif as *FigMaker*, and denote the persons who those figurines represent (as surrogates) as *FigSurrogate*. Then $\{FigMaker, FigSurrogate\} \subset \{PersianHalif\}$; $\{browse = (student \times HD\_BrowseManager, browsing), search=(director \times HD\_SearchManager, searching), describe=(specialist \times FigSurrogate, \Phi)\} \subset R$.

3.5. Use of the 5SGraph Tool to Specify Archaeological DLs

The key players of DLs are digital librarians, who function at various levels, especially as DL experts and DL designers. These players can be more efficient and effective in helping with the development of DLs if they can work with models instead of code and be assisted with DL development tools.

Fig. 3.5.1 illustrates use of 5SGraph [156, 157] to specify the Nimrin archaeological site, here focusing on Structure, drawing upon a meta-model for archaeology that we have built for ETANA-DL [122-124]. Nimrin has three metadata catalogs, and each has its corresponding metadata format as described in its local schema. The scenario model for the Halif site only consists of a database searching service as shown in Fig. 3.5.2, while ETANA-DL has eight main services (see Fig. 3.5.3). In Chapter 4, we present how to integrate various structure models into the one for the union DL using the visual mapping tool.

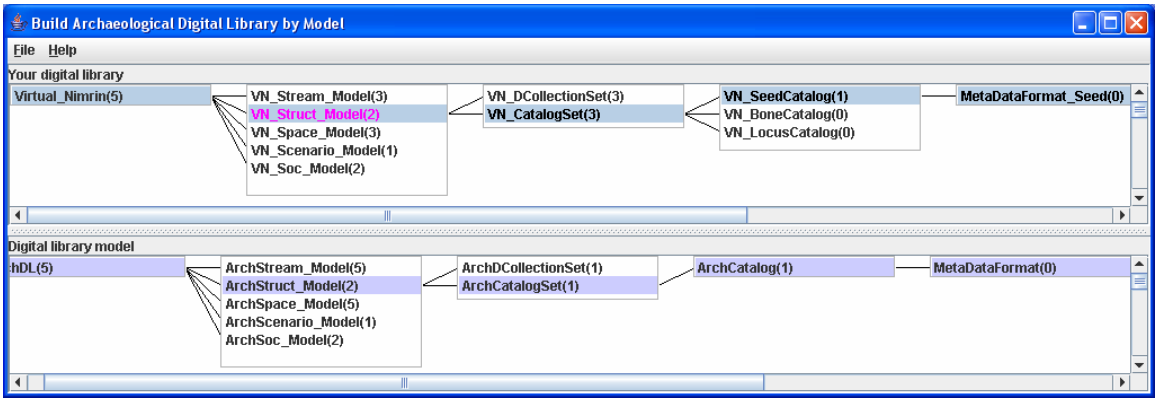


Fig. 3.5.1 Structure model for Nimrin

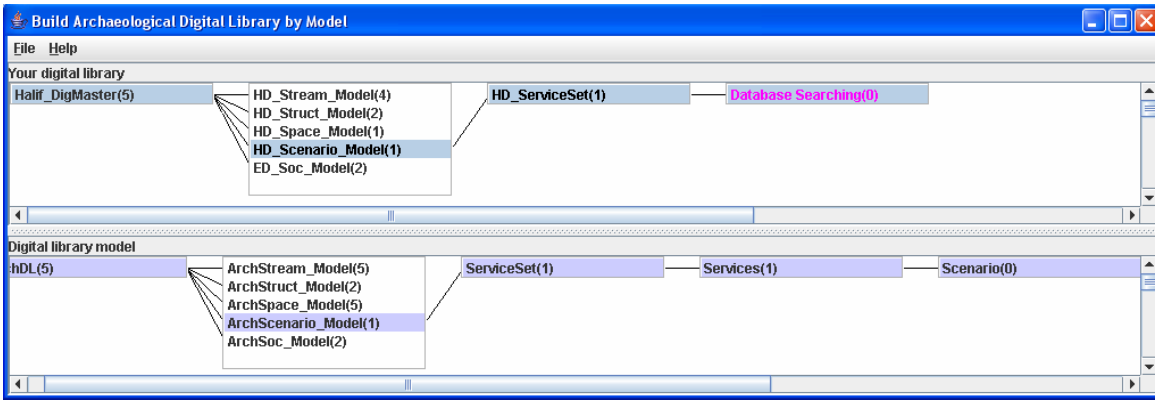


Fig. 3.5.2 Scenario model for Tell Halif

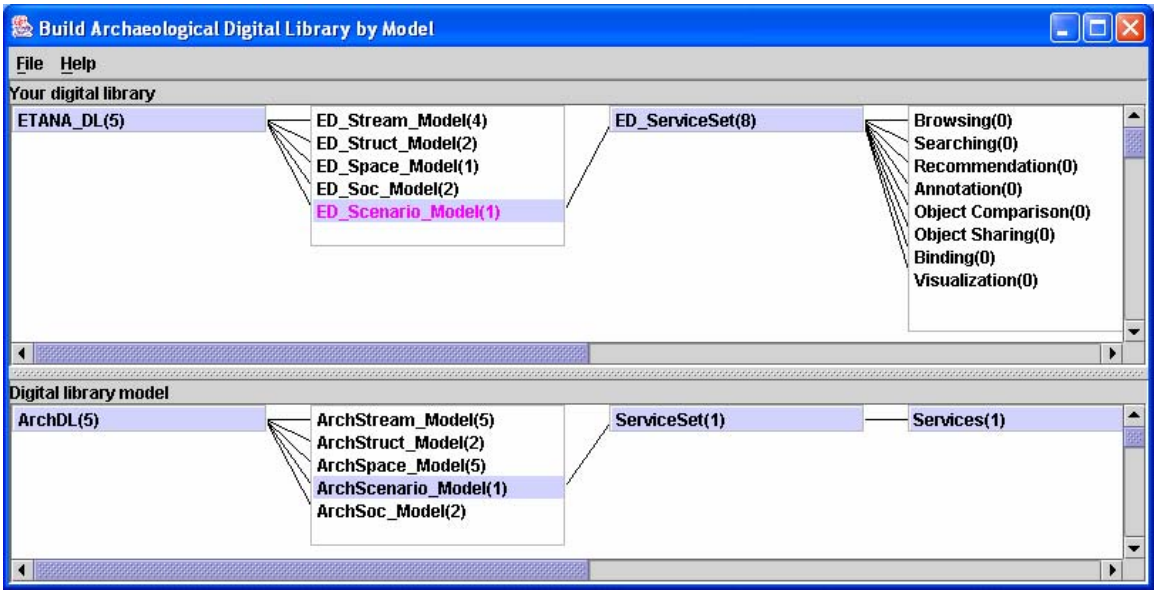


Fig. 3.5.3 Scenario model for ETANA-DL

Chapter 4. Visual Mapping Tool: SchemaMapper

Semantic interoperability is of primary importance in DL integration. Two approaches are interrelated: intermediary-based and mapping-based. The former uses mechanisms like mediators, wrappers, agents, and ontologies. Yet, while many research projects have developed semantic mediators and wrappers to address the interoperability issue, few have tackled the problem of (partial) automatic production of these mediators and wrappers (through a mapping-based approach). The mapping-based approach attempts to construct mappings between semantically related information sources. It is usually accomplished by constructing a global schema and by establishing mappings between local and global schema. In this chapter, we present an incremental approach through intermediary- and mapping-based techniques and a visual mapping tool, SchemaMapper. Section 4.1 describes features of SchemaMapper; its application in archaeological DL integration is illustrated in Section 4.2.

4.1. Features of SchemaMapper

Schema mapping is an interesting problem that so far has been addressed from either an algorithmic point of view or from a visualization point of view. SchemaMapper combines these two perspectives as followings.

1. Algorithmic perspective

Mapping recommendations by SchemaMapper consist of name based (e.g., using edit distance), rule based, and mapping history based strategies.

2. Visualization perspective

SchemaMapper presents local and global schemas using hyperbolic trees [117, 118]. This allows more nodes to be displayed than with linear representation techniques, and avoids the problem of scrolling. Though full node names cannot be displayed (to conserve space), these are available as tool-tip information on individual nodes. Different colors are assigned to differentiate between root level, leaf, non-leaf, recommended, and mapped nodes (with a color legend present on the lower right as in Fig. 4.2.2). A table that contains a list of all the mappings

in the current session is shown at the bottom left of the screen in Fig. 4.2.2. Users may or may not accept recommendations.

SchemaMapper allows global schema editing: deleting nodes, renaming nodes, and adding a local schema sub-tree to the global schema. This has special value for many DLs, e.g., ArchDLs, where it is impossible to predict the final global schema because of its evolutionary nature. SchemaMapper may be superior in this respect to commercial mapping tools like MapForce [63] which lack schema editing capabilities. As a global schema evolves, in order to preserve consistency in the naming of semantically similar nodes, SchemaMapper recommends appropriate name changes to global schema nodes, based on the history stored in a mapping database.

Once the local schema has been mapped to the global schema, an XSLT style sheet containing the mapping is produced by SchemaMapper. This style sheet is essentially the wrapper containing the mappings. When applied to a collection of XML files conforming to the local schema, the style sheet transforms those files to the ones conforming to the global schema. The transform files can be harvested into a union DL. SchemaMapper also saves any changes made to the global schema, and updates the mapping database.

4.2. Case Study: Archaeological DL Application

During the past several decades, archaeology as a discipline and practice has increasingly embraced digital technologies and electronic resources. Vast quantities of heterogeneous data are generated, stored, and processed by customized monolithic information systems. Migration or export of archaeological data from one system to another is a monumental task that is aggravated by peculiar data formats and database schemas. Furthermore, archaeological data classification depends on a number of vaguely defined qualitative characteristics, which are open to personal interpretation. Different branches of archaeology have special methods of classification; progress in digs and new types of excavated finds makes it impossible to foresee an ultimate global schema for the description of all excavation data [36]. Accordingly, an “incremental” and “bottom up” approach is desired for global schema enrichment.

In this section, we explain how all these DL integration requirements can be satisfied, through semi-automatic wrapper generation based on SchemaMapper that simultaneously improves the global schema. Through the integration of artifact data from the Megiddo

excavation site into ETANA-DL, we demonstrate that SchemaMapper allows semi-automatic mapping and incremental global schema enrichment, and supports union catalog generation for a union DL.

4.2.1. Megiddo overview

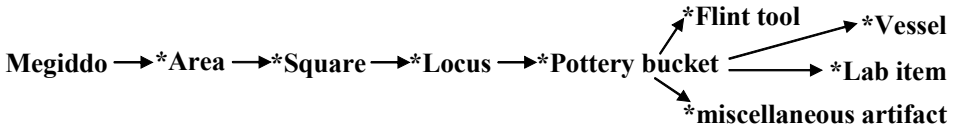


Fig. 4.2.1 Megiddo site organization

Megiddo is widely regarded as the most important archaeological site in Israel from Biblical times and as one of the most significant sites for the study of the ancient Near East. The excavation data collection we received from Megiddo is stored in more than ten database tables containing over 30,000 records with seven different types, namely wall, locus, pottery bucket, flint tool, vessel, lab item, and miscellaneous artifact. The Megiddo schema is described in a structure sub-model (see Fig. 1.3.4 in Chapter 2) within the 5S framework. Structures represent the way archaeological information is organized along several dimensions; it is spatially organized, temporally sequenced, and highly variable. Megiddo site organization is shown in Fig. 4.2.1.

4.2.2. Scenario for mapping Megiddo schema into ETANA-DL global schema

As described earlier, the Megiddo collection consists of seven different types of artifacts. For integrating it into ETANA-DL, we produce one mapping style sheet per artifact. In the following scenarios, we first consider the mapping of "flint tool", and then use the knowledge of this mapping to help map "vessel".

The left hand side of Fig. 4.2.2 shows the Megiddo local schema, while the right hand side shows the ETANA-DL global schema. The ETANA-DL global schema contains the BONE, SEED, FIGURINE, LOCUSSHEET, and POTTERY artifacts already included, apart from the top-level leaf nodes (OWNERID, OBJECTTYPE, COLLECTION, PARTITION, SUBPARTITION, LOCUS, and CONTAINER) that would be presented in all artifacts.

Based on rules and name based matching strategies, SchemaMapper recommends mappings: OWNERID → OWNERID, OBJECTTYPE → OBJECTTYPE, COLLECTION →

COLLECTION, Area → PARTITION, Square1 → SUBPARTITION, Locus →LOCUS, and OriginalBucket → CONTAINER (OWNERID, OBJECTTYPE, and COLLECTION are top-level leaf-nodes whereas Area, Square1, Locus, and OriginalBucket are all elements of the schema of the flint tool collection).

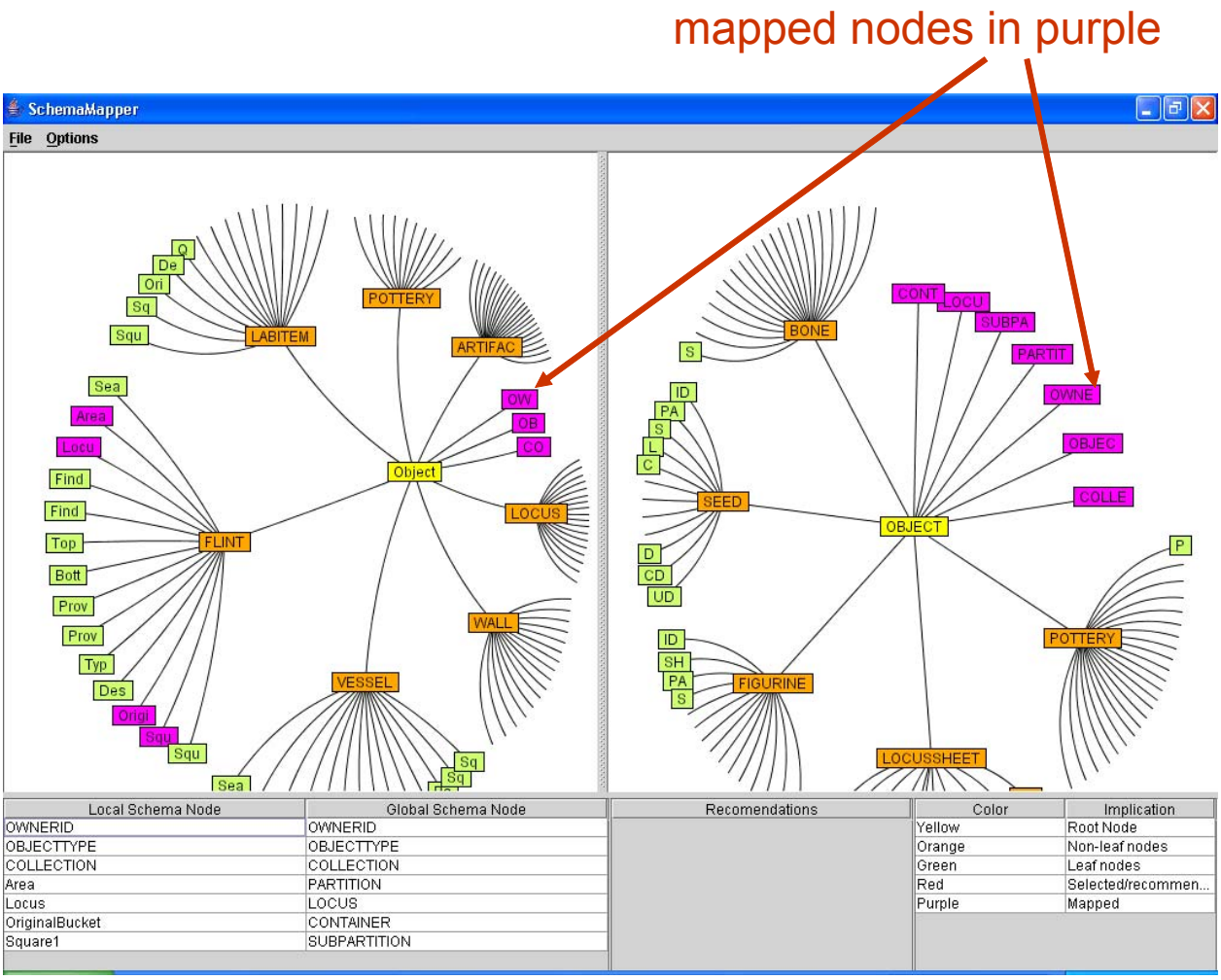


Fig. 4.2.2 Initial set of mappings for flint tool based on rules and name-based matching

The above mapping format has the local schema node on the left hand side and the recommended global schema node on the right hand side. We map the nodes according to the recommendations, indicated by coloring these nodes purple (see Fig. 4.2.2).

As the remaining nodes in the local schema do not have corresponding global schema nodes, we add the flint tool sub-tree as a child of the OBJECT node in the global schema. This ensures that local schema elements and properties are preserved during the mapping

transformation. SchemaMapper determines that some of the nodes (Area, Locus, OriginalBucket, and Square1) are already mapped, deletes these nodes from the global schema sub-tree, and automatically maps the rest with the corresponding elements in the local sub-tree (see Fig. 4.2.3). The user may decide to rename some nodes in the global schema from within this sub-tree to avoid any local connections with the name. Assume the user renames global schema node "Description" to "DESCRIPTION". With this the mapping process is complete (see Fig. 4.2.3). Once the user decides to confirm the mappings, a style sheet is generated, the mappings are stored in the database, and the ETANA-DL global schema is updated with the flint tool schema.

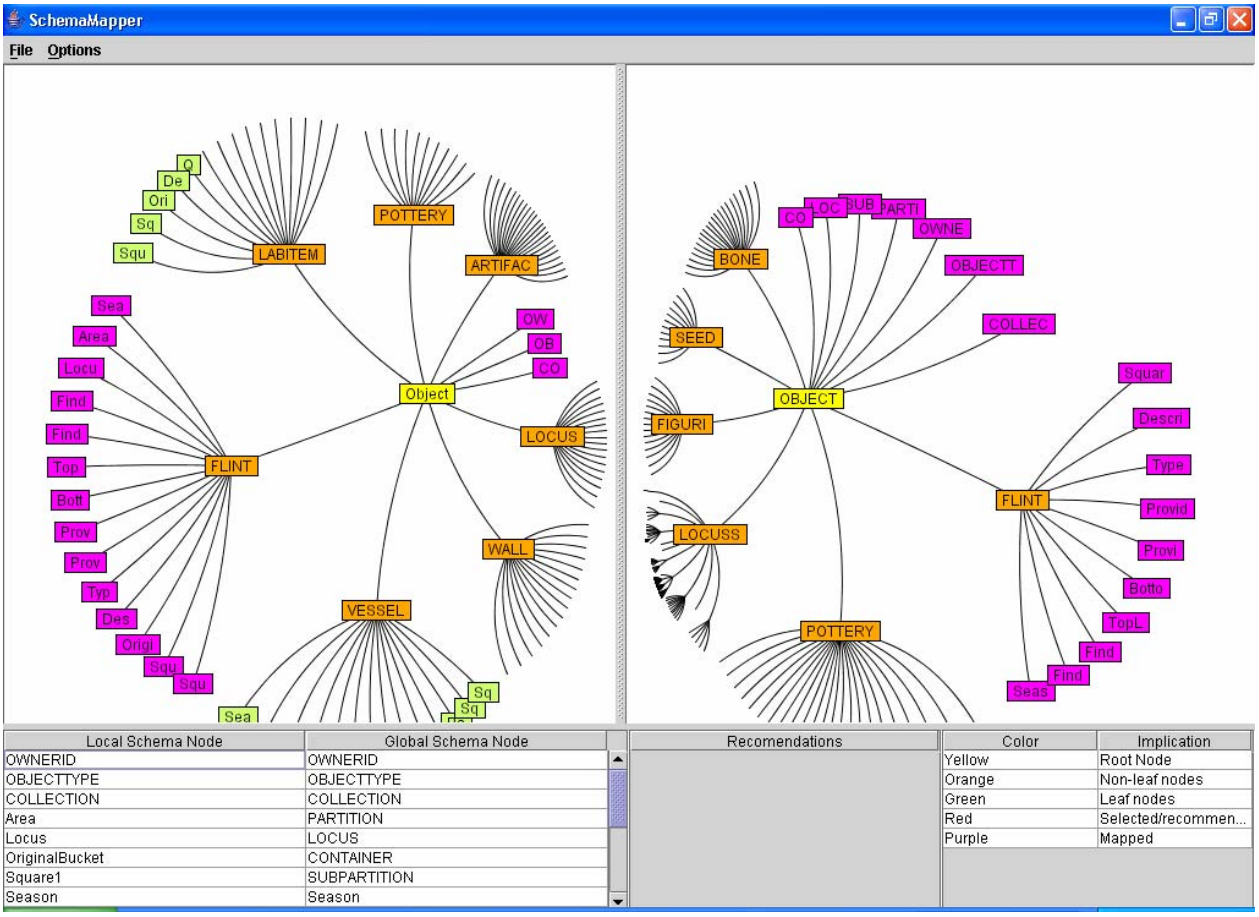


Fig. 4.2.3 Adding FLINT sub-tree as a child of OBJECT in the global schema

We next integrate the schema of VESSEL artifacts of Megiddo into the ETANA-DL global schema. When we open the global schema for mapping, along with the other artifacts, the flint tool, which was integrated in the previous step, also is present (see Fig. 4.2.4). From the mapping of flint tool we realize that mapping of a completely new artifact requires only the top-

level leaf nodes to be displayed in the global schema. For integration of a completely new artifact, the user may choose to view only the top-level leaf nodes in order to avoid erroneous cross mappings from schema nodes of one of the artifacts to similar schema nodes present in other artifacts (see Fig. 4.2.4 and Fig. 4.2.5). This prevents the user from accidentally modifying a node, from say the flint tool sub-tree in the global schema, and rendering the previously generated XML files inconsistent. Also, this avoids confusing the user by presenting him with only the information he needs to see for mapping. Once again recommendations are made to enable the initial set of seven mappings; after this, the user adds the VESSEL sub-tree to the global schema.

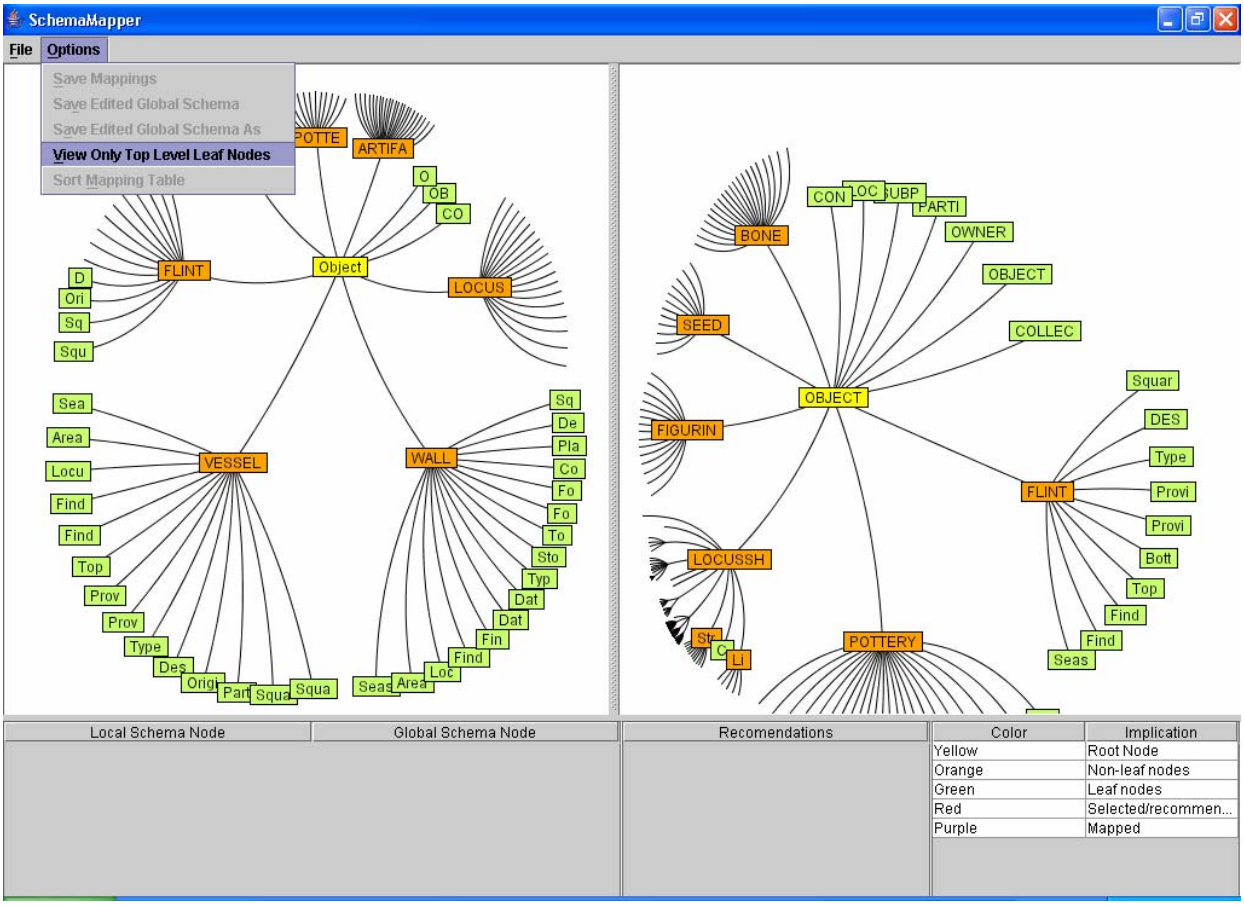


Fig. 4.2.4 Using the View Only Top Level Leaf Nodes option mapping Vessel Collection

As before, SchemaMapper finds that the Area, Locus, Square1, and Original-Bucket nodes are already mapped – and deletes them in the global sub-tree and maps the remaining nodes to corresponding local schema nodes automatically. SchemaMapper also goes through the mappings history and finds that the Description node in the flint tool sub-tree was mapped to the

DESCRIPTION node in the global schema. In order to keep naming consistent, Schema Mapper recommends the user to change the name of the Description node in the VESSEL sub-tree to DESCRIPTION (see Fig. 4.2.5). This is due to the fact that both the DESCRIPTION node in the flint tool sub-branch of the global schema and the Description node in the VESSEL sub-branch of the global schema describe the same artifact type, but as DESCRIPTION has been selected as the global name, all Description elements in the global sub-tree should be renamed as DESCRIPTION. The recommendation, as always, is not mandatory, but if followed will help keep names consistent. When the user confirms the mappings, the database is updated, the style sheet generated, and the global schema updated with the VESSEL schema. It is important to note that the integration of vessel artifacts into the global schema in no way changed the existing flint global entry. This leads us to the observation that, for Megiddo, modification of the global schema is simply appending a new local artifact into the global schema without changing the existing artifacts in the global schema.

The style sheets generated are applied on each sub-collection of Megiddo (like vessel or flint tool collection) to convert local collections to the one conforming to the global schema. Transformed collections are ready for harvest into the union catalog in ETANA-DL, and available for access by services like Searching and Browsing (described in Chapter 5).

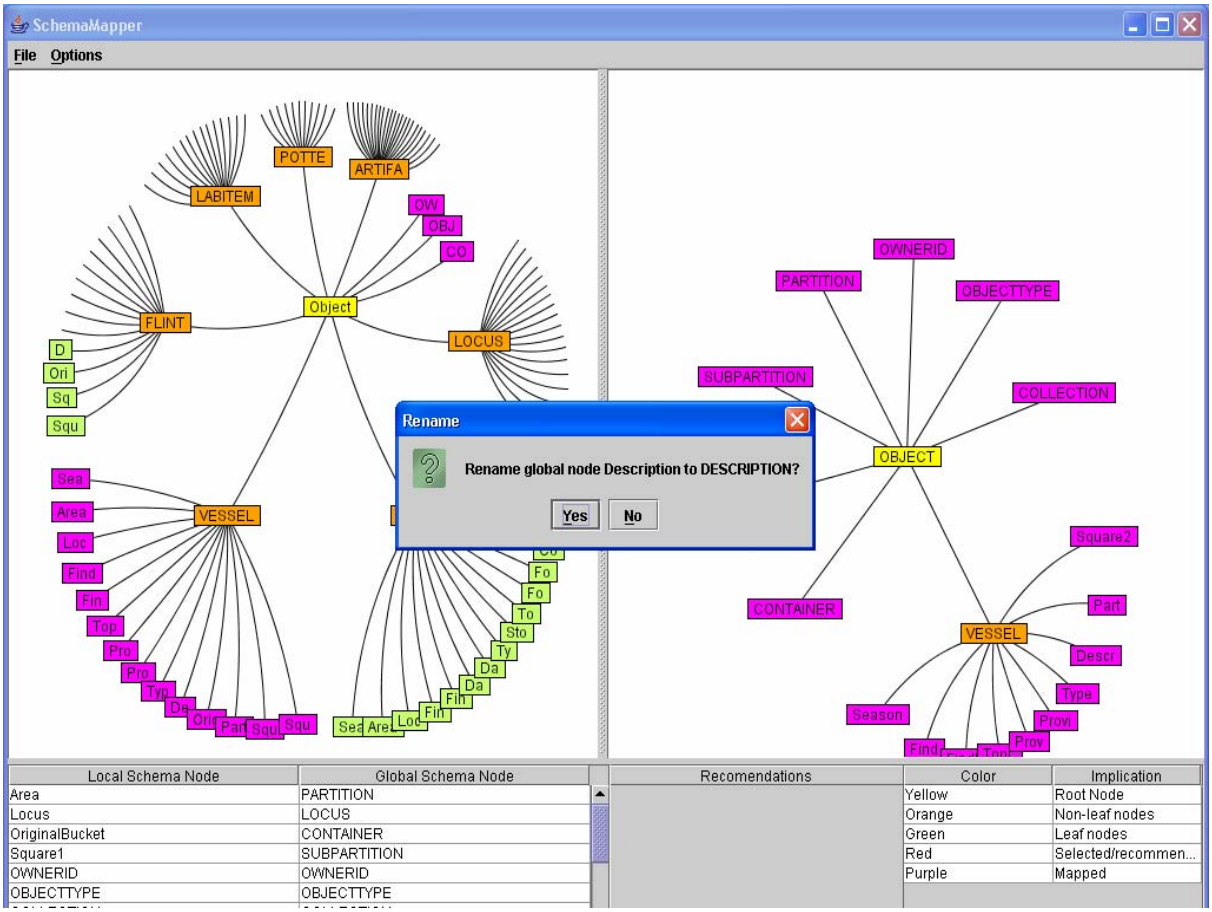


Fig. 4.2.5 Name change recommendation based on mapping history

Chapter 5. Exploring Services

Union services of a union DL include exploring services. By exploring we mean searching, browsing, investigating, studying, or analyzing for the purpose of discovery, e.g., pursuing truth or facts about something. In this chapter, we first formalize and generalize DL exploring services within the 5S framework. Next we develop theorems to indicate that browsing and searching can be converted or mapped to each other under certain conditions. The theorems guide the design and implementation of exploring services for an integrated archaeological DL, ETANA-DL. Its integrated browsing and searching can support users in moving seamlessly between browsing and searching, minimizing context switching, and keeping users focused. It also integrates browsing and searching into a single visual interface for DL exploration. Finally, we present a formative user evaluation of these exploring services in ETANA-DL.

5.1. Introduction

Browsing and searching are two major paradigms for exploring DLs. They are often provided by DLs as separate services. Developers commonly see these functions as having different underlying mechanisms, and they follow a functional, rather than a task-oriented approach to interaction design. While exhibiting complementary advantages, neither paradigm alone is adequate for complex information needs (e.g., that lend themselves partially to browsing and partially to searching [104]). Searching is popular because of its ability to identify information quickly. On the other hand, browsing is useful when appropriate search keywords are unavailable to users (e.g., a user may not be certain of what she is looking for until the available options are presented during browsing; certain criteria do not lend well to keyword search; the exact terminology used by the system may not be known). Browsing also is appropriate when a great deal of contextual information is obtained along the navigation path. Therefore, a synergy between searching and browsing is required to support users' information-seeking goals [10, 11, 46, 48, 91]. Accordingly, a panel at the World Wide Web Conference in 2005 brought together experts to discuss the trends in the integration of searching and browsing,

and in 1995 there was a panel on "Browsing vs. Search: Can We Find a Synergy?" at the Conference on Human Factors in Computing Systems.

Text mining and visualization techniques provide DLs additional powerful exploring services, with possible beneficial effects on browsing and searching. Our study of the CitiViz system [71], which combines browsing, searching, document clustering, and information visualization, showed its advantages, in user performance and preference, relative to traditional interfaces.

Though many research projects have developed different interaction strategies allowing smooth transition between browsing and searching, to the best of our knowledge, none of them generalize these two predominant exploring services in DLs. Reflecting upon the current state of the art and different types of exploring services for DLs has led us to the following research questions:

- Are browsing and searching duals of each other, or can they be converted to each other when certain conditions are met?
- Can we generalize these DL exploring services within a formal DL framework?
- Can the formal generalization guide development of exploring services for domain focused DLs?

To address the above mentioned questions, we:

- Generalize DL exploring services – such as browsing, searching, clustering, and visualization – in the context of the 5S DL theory [50, 53] and develop theorems and lemmas based on the formal generalization.
- Prove that browsing and searching can be converted and mapped to each other under certain conditions based on the theorems and lemmas developed.
- Use an integrated archaeological DL, ETANA-DL (http://etana.dlib.vt.edu) [122, 137], as a case study to illustrate the application of our theoretical approach. We conducted a user study to evaluate ETANA-DL's exploring services. We found that users significantly prefer to integrate browsing and searching.

To the best of our knowledge, we are the first to approach DL exploring services based on a DL theory. Studying DL exploring from this viewpoint has provided several insights. For instance, the formalisms bring a theoretical approach to the subject and the theorems we developed indicate browsing and searching can be converted and switched to each other under

certain conditions. In addition, the theoretical approach provides a systematic and functional method to design and implement DL exploring services.

We think our work has made contributions to aid both users and developers of DLs. For users, fluidity between browsing and searching supports them in achieving their information-seeking goals, thus helps bridge their mental model of an/the information space with the information systems' representation. For DL developers, we suggest some new possibilities for blurring the dividing line between browsing and searching. If these two services are not considered to have different underlying mechanisms, they will not be provided as separate functions in DLs and may be better integrated.

The remainder of this chapter is structured as follows. Section 5.2 formalizes DL exploring services. Section 5.3 describes the exploring services for our archaeological DL, developed based on the theorems and lemmas. Section 5.4 presents the user evaluation of those services.

5.2. Exploring Service Formalization

Notation:

Let C be a collection (a set of digital objects; see Def. 17 in [53] for details), and 2^C be the set of all subsets of C. Let ϕ be the empty set.

Let $HT = (H, Contents, P)$ be a hypertext, where

1) $H=((V_H, E_H), L_H, F_H)$ is a structure (i.e., a directed graph with vertices V_H and edges E_H, along with labels L_H and labeling function F_H on the graph; see Def. 2 in [53] for details)

2) $Contents \subseteq C \cup AllSubStreams \cup AllSubStructuredStreams$ can include digital objects of a collection C, all of their (sub)streams (a stream is a sequence whose codomain is a nonempty set; see Def. 1 in [53]) and all possible restrictions of the *StructuredStream* (see Def. 15. in [53] for details) functions of digital objects.

3) $P: V_H \rightarrow 2^{Contents}$ is a function which associates a node of the hypertext with the node content. Note that the range of P is $2^{Contents}$ instead of $Contents$ as (incorrectly) defined in Def. 22 in [53].

According to the definition of a minimum DL in [53], a DL has hypertext and it is a web accessible information system. Therefore, $\forall$ C, $\exists HT$, i.e., for each collection C in a DL, there exists a hypertext (statically or dynamically created) associated with C.

If $subC \in 2^C$ and $subC \neq \phi$, $subC$ can be partitioned into a set of (non)overlapping clusters (groups) $\{cluster_1, cluster_2, ..., cluster_k\}$, where $cluster_i$ is denoted as a cluster belonging to $subC$, and $\bigcup_{i=1}^{k} cluster_i = subC$.

Contents of $subC$ is denoted $CluCon(subC) = \{cluCon_1, cluCon_2, ..., cluCon_k\}$, where $cluCon_i$ is the contents associated with $cluster_i$.

Let *VSpa* be a vector space (see Def. 13 in the appendix of [53]) and *Base* be a set of basis vectors in *VSpa*. Let *{VisualM}* be a set of visual marks (e.g., points, lines, areas, volumes, and glyphs) and *{VisualMP}* be a set of visual properties (e.g., position, size, length, angle, slope, color, gray scale, texture, shape, animation, blink, and motion) of visual marks.

Definition 1: Let $Q = \{(H_q, Contents_q, P_q)\}$ be a set of conceptual representations for user information needs, where $H_q=((V_q, E_q), L_q, F_q)$ is a structure (i.e., a directed graph with vertices V_q and edges E_q, along with labels L_q and labeling function F_q on the graph; see Def. 2 in [53] for details), $Contents_q$ can include digital objects and all of their streams, and P_q is a mapping function $P_q: V_q \rightarrow Contents_q$.

The notion of conceptual representations for user information needs was used in [53] to define searching service, however, it was not formally defined in [53]. Def. 1 is a formal definition for conceptual representations for user information needs. Based on Def. 1, we can define not only searching, but also browsing services. The examples illustrated below show conceptual representations for user information needs related to textual and image retrieval, and hypertext navigation.

Examples of user information needs: $q = (H_q, Contents_q, P_q) \in Q$

Examples from a) through c) show that conceptual representation for user information needs are materialized into a query specification.

a) Textual retrieval: q is a keyword named "energy".

A user's information need is something about energy; she may explicitly express it as a keyword "energy".

$q = ((V_q, E_q), L_q, F_q), Contents_q, P_q)$, where $V_q = \{v_1\}$, $E_q=\phi$, $L_q = \phi$, $F_q: V_q \rightarrow L_q$, $Contents_q$ is the stream of string "energy", and $P_q: V_q \rightarrow Contents_q$.

In this case, $H_q = ((V_q, E_q), L_q, F_q)$ is a one-node graph (see Fig. 5.2.1), and P_q maps that node to its contents, i.e., string "energy" (indicated by the dashed arrows in Fig. 5.2.1).

Fig. 5.2.1 *q* is a keyword named "energy".

b) Textual retrieval: q is a structured query named "animal bones from the Nimrin site".

A user wants to find records about animal bones from the Nimrin Site from ETANA-DL (an integrated archaeological DL [137]). *q* is a structured query represented as *'+objectType:Bone +site:Nimrin'* based on the query language of ETANA-DL. "+*objectType:Bone*" means that the object type of the desired records should be bone (i.e., the attribute *objectType* should contain value *Bone*); "+*site:Nimrin*" means that the records should be from site Nimrin (i.e., the attribute *site* should contain value *Nimrin*).

$q = ((V_q, E_q), L_q, F_q), Contents_q, P_q)$, where $V_q = \{v_1, v_2\}$, $E_q = \phi$, $L_q = \{$'object type', 'site'$\}$, F_q: $V_q \rightarrow L_q$, $Contents_q$ is the stream of strings "animal bones" and "Nimrin", and P_q: $V_q \rightarrow Contents_q$.

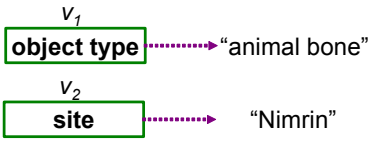


Fig. 5.2.2 q is a structured query named "animal bones from the Nimrin site".

In this case, $H_q = ((V_q, E_q), L_q, F_q)$ is a two-node graph with 'object type' and 'site' as labels for these two nodes (see Fig. 5.2.2), and *P* maps each node to its contents, i.e., string "animal bones" and "Nimrin" respectively (indicated by the dashed arrows in Fig. 5.2.2).

Structured query *q* was defined as a set of attribute-value pairs: $q = \{A_1 : value_{1q}, ..., A_k : value_{kq}, ..., A_n : value_{nq}\}$, where A_k is an attribute or metadata field and each $value_{kq}$ is a value belonging to the domain of A_k [52]. We find that this definition can be derived from Def. 1 (definition of a set of conceptual representations for user information needs). By Def. 1, we get $A_k = F_q(v_k)$ and $value_{kq} = P_q(v_k)$, i.e., A_k is the label of node v_k and $value_{kq}$ is the contents associated with v_k.

c) Image retrieval: *q* itself is an image, which contains five spatially related sub-images (objects).

A user wants to find some images similar to an existing one as shown in Fig. 5.2.3 (a).

$q = ((V_q, E_q), L_q, F_q), Contents_q, P_q)$, where $V_q = \{v_1, v_2, v_3, v_4, v_5\}$, $E_q = \{e_1, e_2, e_3, e_4, e_5\}$, $L_q = \{$'fire', 'earth', 'metal', 'water', 'wood', 'produce'$\}$, F_q: $V_q \cup E_q \rightarrow L_q$, $Contents_q$ is the stream of the five spatially related sub-images with their location information, and P_q: $V_q \rightarrow Contents_q$.

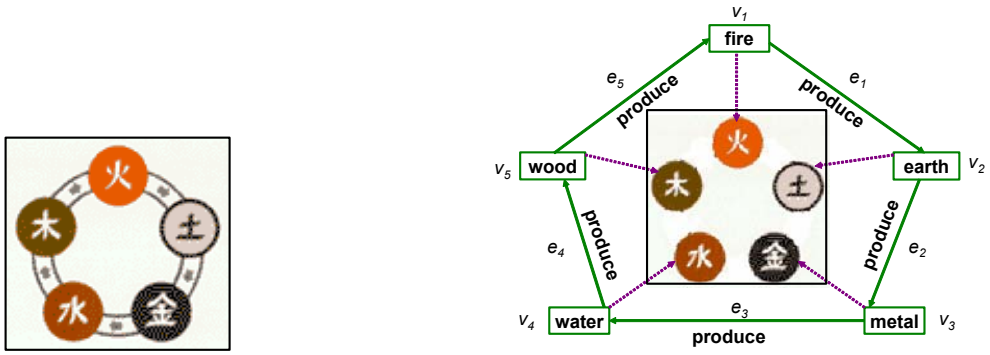


Fig. 5.2.3 q is an image of 5 spatially related sub-images.

In this case, H_q is a graph of five nodes with labels 'fire', 'earth', 'metal', 'water', and 'wood' respectively as illustrated in Fig. 5.2.3 (b). P_q maps each node to its contents, i.e., the associated sub-image with its spatial information (indicated by the dashed arrows in Fig. 5.2.3). This kind of query representation has been used to retrieve images according to spatial relationships of objects or layout representations (e.g., [128, 135]).

d) Navigation starting point

$q = ((V_q, E_q), L_q, F_q), Contents_q, P_q)$, where $V_q = \{v_1\}$, $E_Q = \phi$, $L_q = \{$'ETANA-DL'$\}$, F_q: $V_q \rightarrow L_q$, $Contents_q$ is the homepage of ETANA-DL, and P_q: $V_q \rightarrow Contents_q$.

In this case, $H_q = ((V_q, E_q), L_q, F_q)$ is a one-node graph with 'ETANA-DL' as label for that node (see Fig. 5.2.4), and P_q maps that node to its contents, i.e., the ETANA-DL homepage (indicated by the dashed arrows in Fig. 5.2.4). In this situation, a user does not have an explicit information need like a query though she may have a conceptual information need. She wants to know something about ETANA-DL. She goes to its homepage and her navigation start point represents her initial information need.

Fig. 5.2.4 *q* is a user's navigation start point.

Definition 2: An Exploration Space (***ESpa***) is a tuple, ***ESpa*** = *(Q, Contents, OP_Set)*, where *Q* is a set of conceptual representations for user information needs (see Def. 1); *Contents* can include digital objects of a collection *C* (*C* is a set of digital objects), as well as all of their (sub)streams and all possible restrictions of the *StructuredStream* functions of digital objects; and *OP_Set* is a set of operations on *Q* and *Contents*. $\{ OP_{viz}, OP_{s}, OP_{b}, OP_{clu}\} \subseteq OP\_Set$, where *viz*, *s*, *b*, and *clu,* relate to visualization, search, browse, and cluster operations, respectively, and

1) $OP_{viz} = \{VisualMap_1, VisualMap_2, VisualMap_3\}$, where

$VisualMap_1$: $2^C \rightarrow VSpa$ associates a set of digital objects with a set of vectors;

$VisualMap_2$: $2^C \rightarrow VisualM$ associates a set of digital objects with a visual mark;

$VisualMap_3$: $Base \rightarrow VisualMP$ associates a basis vector with a visual property of a visual mark.

Examples of OP_{viz}:

A special case is that there is only one digital object, a document in the set. Given a vector space *VSpa* of three dimensions, the document is mapped to a vector of three elements, i.e., its length, date published, and number of citations, by function $VisualMap_1$. It is mapped to a visual mark, a point in 2D space, by function $VisualMap_2$. The first two base vectors in *VSpa* are associated with the position of the point in 2D space, while the third base vector may be mapped to another visual property of the point, its gray scale (e.g., a document

represented by a black point has more citations than a document represented by a gray point).

Fig. 5.2.5 shows another example of OP_{viz}. A set of digital objects contains three bone records in the ETANA-DL bone collection. Each of these records is mapped to a vector in a vector space *VSpa* by function $VisualMap_1$ and mapped to a special visual mark: rows of text by function $VisualMap_2$. Two base vectors in *VSpa* are associated with the position of the rows of text in a 2D user interface.

Fig. 5.2.5 Example of OP_{viz}

2) OP_{clu}: $(2^C \times 2^C) \times Sim_{clu} \rightarrow 2^{Contents}$, where

$Sim_{clu} = \{OP_{clu1}(cluster_x, cluster_y) | \ cluster_x \in 2^C, \ cluster_y \in 2^C\}$, where OP_{clu1}: $2^C \times 2^C \rightarrow R$ is a matching function that associates a real number with a pair of subsets of *C*. Sim_{clu} is a set of numerical values measuring the similarity between each pair of subsets of *C*. Similarity measures between clusters are called linkage methods. The three most popular linkage methods (single-link, complete-link, and group-average) were presented in [129]. The range of OP_{clu} is a set of the *Contents* associated with collection *C*. Note that OP_{viz} may be applied to the result of OP_{clu}.

Example of OP_{clu}:

C is a set of all the digital objects in ETANA-DL; $cluster_x$ and $cluster_y$ are subsets of *C*, and they are bone records from the Nimrin site and Umayri site respectively as shown in Fig. 5.2.6. If the similarity between $cluster_x$ and $cluster_y$ is above a predefined threshold, OP_{clu} returns the contents associated with a new cluster, $cluster_x \cup cluster_y$, i.e., a set of all bone

records. $cluster_x$ has 7419 records and $cluster_y$ has 2122 records; while the clustering result, $cluster_x \cup cluster_y$, has 9541 records as shown in Fig. 5.2.7.

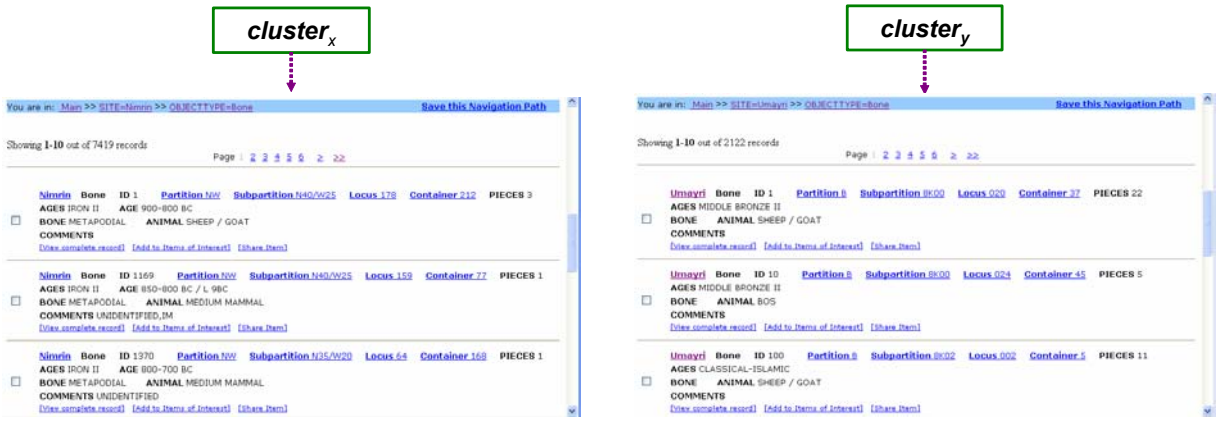


Fig. 5.2.6 Example of $cluster_x$ and $cluster_y$ in ETANA-DL

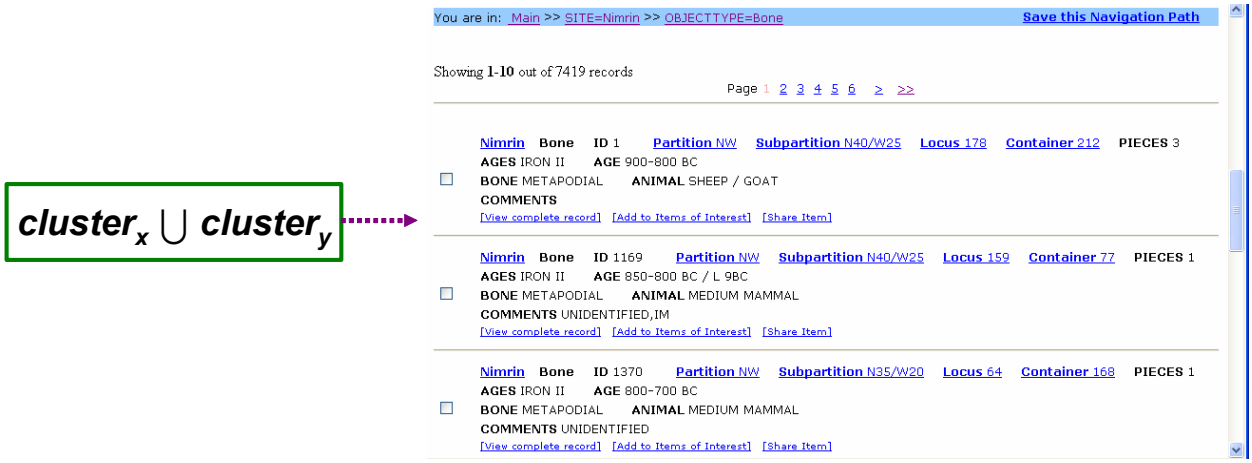


Fig. 5.2.7 Example of clustering result

3) OP_s: $(Q \times C) \times Sim_s \rightarrow 2^{Contents}$, where

$Sim_s = \{OP_q(q, do) | \ q \in Q, \ do \in C \}$, where OP_q: $Q \times C \rightarrow R$ is a matching function that associates a real number with $q \in Q$ and a digital object $do \in C$. The range of function OP_s is the *Contents* associated with collection C. While the similarity function OP_q was defined in Def. 21 in [53], the retrieved results were not defined there. We consider the retrieved results as (a subset of) the *Contents*. OP_{viz} and OP_{clu} may be applied on the result of OP_s. Example of OP_s:

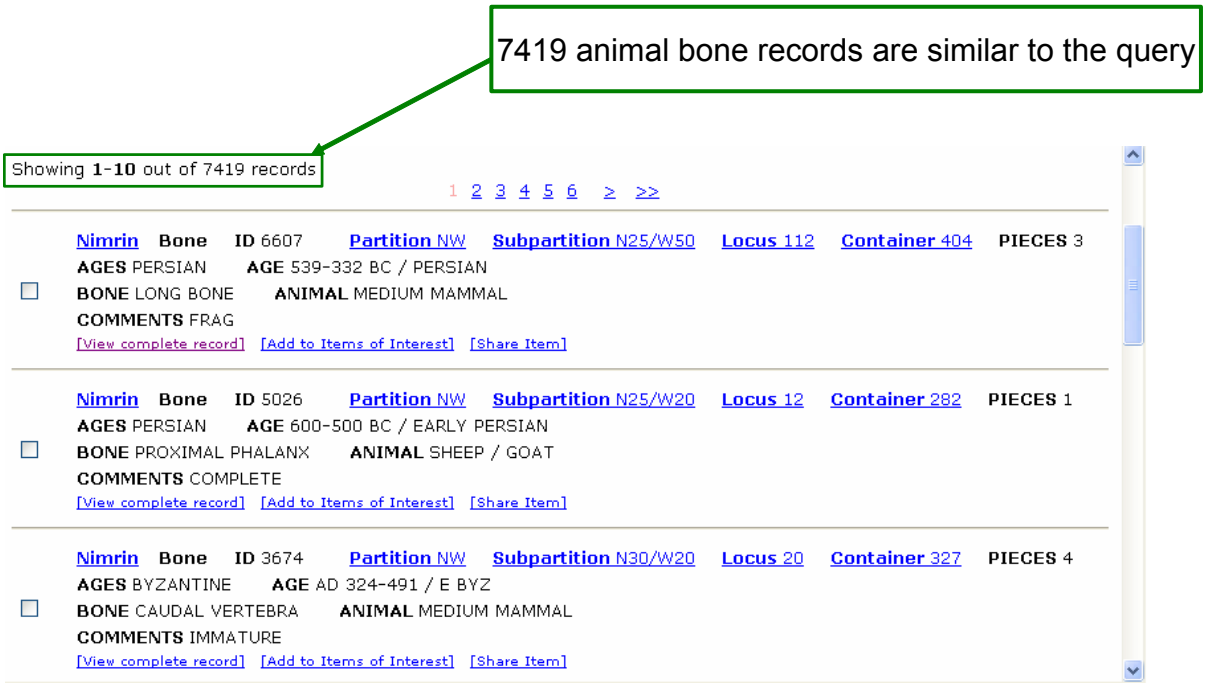


Fig. 5.2.8 Example of function OP_q in ETANA-DL

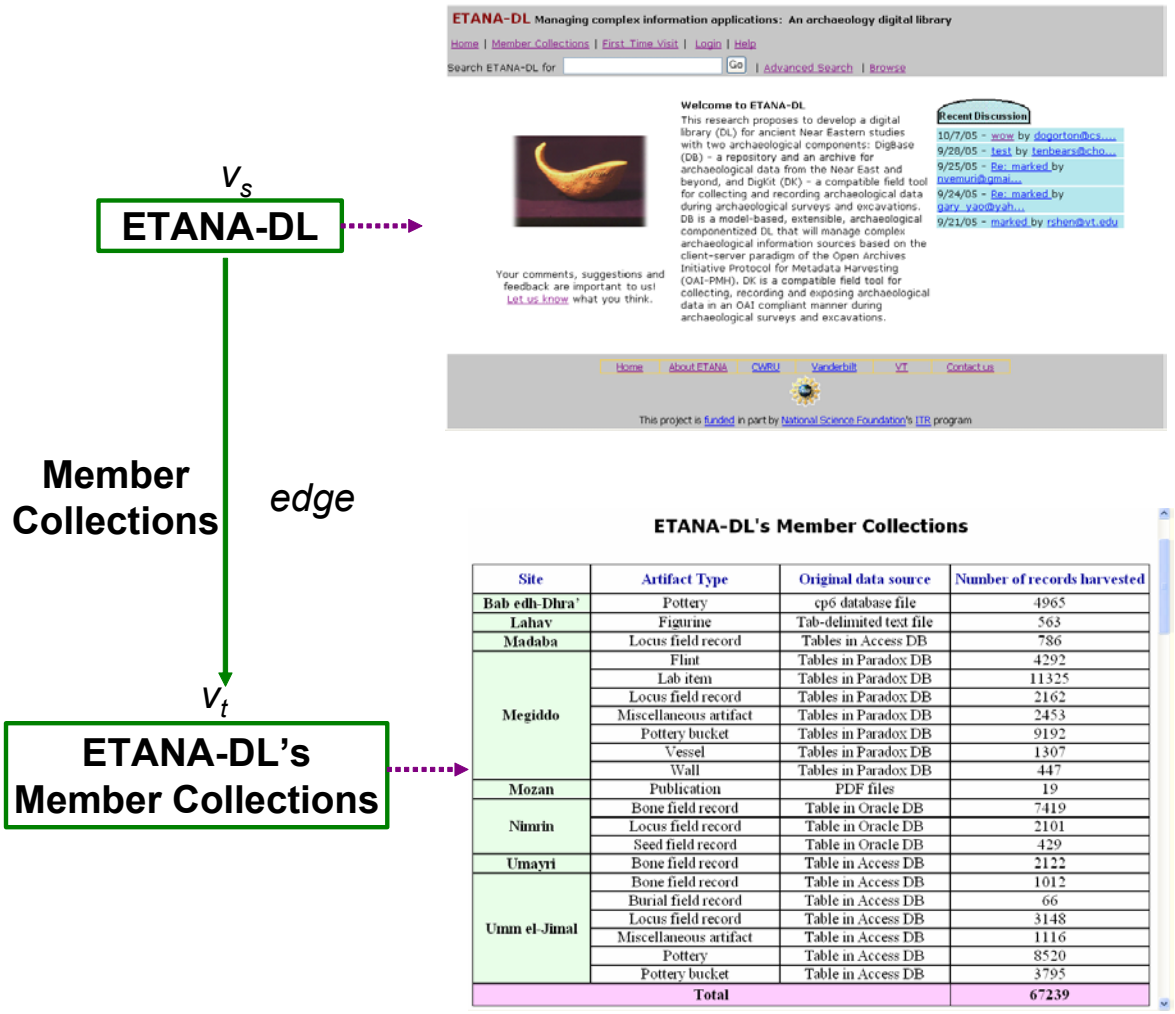
q is a structured query named "animal bones from the Nimrin site" as illustrated in Fig. 5.2.2 before; C is a set of all the digital objects in ETANA-DL; Sim_s is a set of numerical values measuring the similarity between q and each digital object using the vector space model (cosine similarity) [130]. Based on Sim_s, OP_s returns the contents associated with a set of digital objects whose similarity between q is above a predefined threshold. There are 7419 animal bone records similar to the query; OP_{viz} is applied to the result of function OP_s and the retrieved results are shown in Fig. 5.2.8.

4) OP_b: $E_H \rightarrow 2^{Contents}$ is a function which, given a link, retrieves the content of the target node, where E_H is a set of edges of the digraph defined for a hypertext.

The *TraverseLink* function defined in [53] was intended to achieve the same result as OP_b. We think both the domain and range of *TraverseLink* function may need to be refined. The domain of *TraverseLink* function can be generalized and the range of it is not proper. The domain of *TraverseLink* is $V_H \times E_H$, while the domain of OP_b is E_H. Since $\forall\, e=(v_s, v_t) \in E_H$ is a directed edge having a start vertex v_s and an end (target) vertex v_t, the input of OP_b can be simplified as e instead of a pair (v_s, e) as required by *TraverseLink*. The output of OP_b is a set of *Contents*, therefore, the range of OP_b is $2^{Contents}$ (the range of function *TraverseLink* is *Contents*). Note OP_{viz} may be applied to the result of function OP_s as well.

Example of OP_b:

edge $=(v_s, v_t)$ is labeled as "Member Collections", where v_s is labeled as "ETANA-DL", v_t is labeled as "ETANA-DL's Member Collections" , $v_s, v_t \in V_q$, and $v_s, v_t \in V_H$. $OP_b(edge)$ is the content of the target node v_t, i.e. the user's new information need represented by the webpage describing ETANA-DL's member collections (see Fig. 5.2.9).



ETANA-DL's Member Collections

Site	Artifact Type	Original data source	Number of records harvested
Bab edh-Dhra'	Pottery	cp6 database file	4965
Lahav	Figurine	Tab-delimited text file	563
Madaba	Locus field record	Tables in Access DB	786
Megiddo	Flint	Tables in Paradox DB	4292
	Lab item	Tables in Paradox DB	11325
	Locus field record	Tables in Paradox DB	2162
	Miscellaneous artifact	Tables in Paradox DB	2453
	Pottery bucket	Tables in Paradox DB	9192
	Vessel	Tables in Paradox DB	1307
	Wall	Tables in Paradox DB	447
Mozan	Publication	PDF files	19
Nimrin	Bone field record	Table in Oracle DB	7419
	Locus field record	Table in Oracle DB	2101
	Seed field record	Table in Oracle DB	429
Umayri	Bone field record	Table in Access DB	2122
Umm el-Jimal	Bone field record	Table in Access DB	1012
	Burial field record	Table in Access DB	66
	Locus field record	Table in Access DB	3148
	Miscellaneous artifact	Table in Access DB	1116
	Pottery	Table in Access DB	8520
	Pottery bucket	Table in Access DB	3795
Total			67239

Fig. 5.2.9 Example of function OP_b in ETANA-DL

Definition 3: An exploring service (***ESer***) is a set of scenarios $\{sc_1, ..., sc_n\}$ over an exploration space *ESpa*. Each scenario is a sequence of events. An event e_i is associated with one or more of the operations in *ESpa*.

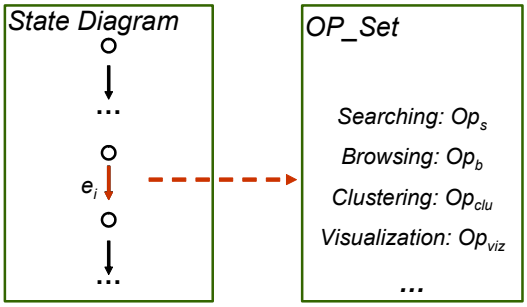


Fig. 5.2.10 Constructs for an exploring service

Fig. 5.2.10 shows two constructs of an exploring service. The left part of Fig. 5.2.10 is a state diagram, which consists of events. The dashed arrow means an event e_i has associated operations(s) in the set of operations, denoted by *OP_Set*. Characterized by its associated operations(s) in *ESpa*, an exploring service can be a searching, browsing, clustering, or visualization service as illustrated in the following theorems and lemmas according to Def. 1 Def. 2, and Def. 3. A sequence of events may be associated with a sequence of operations. e.g., OP_s is followed by OP_{clu}, OP_{viz}, and OP_{clu} as illustrated by the three arrows numbered 1, 2, and 3, respectively (see Fig. 5.2.11).

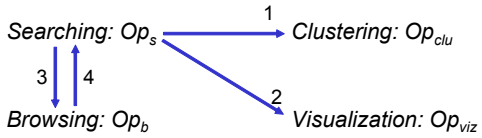


Fig. 5.2.11 Sequence of operations

Table 3 Relationship among theorems (lemmas) and operations

Theorems and Lemmas	Searching Op_s	Browsing Op_b	Clustering Op_{clu}	Visualization Op_{viz}
Theorem 1	√			
Theorem 2		√		
Theorem 3 (Op_s followed by Op_{clu})	√		√	
Theorem 4 (Op_s followed by Op_{viz})	√			√
Lemma 1	√	√		
Lemma 2	√	√		
Lemma 3 (Op_b followed by Op_s)	√	√		
Lemma 4 (Op_s followed by Op_b)	√	√	√	

Table 3 shows the relationships among the theorems (lemmas), operations, and the sequence of these operations. If an operation is used for a theorem (lemma), there will be a check mark in the corresponding cell. Theorem 1 and Theorem 2 discuss searching and browsing services separately; Theorem 3 and Theorem 4 propose post-retrieval clustering and visualization services, respectively; Lemma 1 and Lemma 2 argue that searching and browsing can be mapped to each other under certain conditions; Lemma 3 and Lemma 4 demonstrate switching between searching and browsing.

Theorem 1: If $\forall e_i$, the associated operation with event e_i, is OP_s, then an exploring service is a searching service.

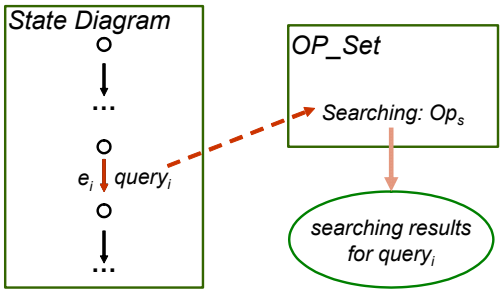


Fig. 5.2.12 An exploring service is a searching service.

The event e_i in Fig. 5.2.12 illustrates that a user issues a query $query_i$. The event then triggers operation OP_s, as indicated by the dashed arrow. The patterned arrow denotes the output of OP_s, i.e., searching results for $query_i$. If the searching result is empty or the user does not think the result is related to her information need, then we consider the user is not satisfied with the searching service.

Proof: $\forall q \in Q$, where Q is a set of conceptual representations for user information needs (see Def.1), there is a searching scenario having a final event of returning the matching function value $sim_s = OP_q(q, do)$ for each digital object $do \in C$ and $\{OP_s((q, do), Sim_s)\}$, the contents of the retrieved digital objects for query q.

Searching services may need indexing services provided by a DL to speed up the performance. We do not discuss indexing services here. Note that the OP_{viz} function may be applied on searching results.

Theorem 2: If $\forall v \in V_q$, $v \in V_H$, and $\forall e_i$, the associated operation with event e_i is OP_b, then an exploring service is a browsing service.

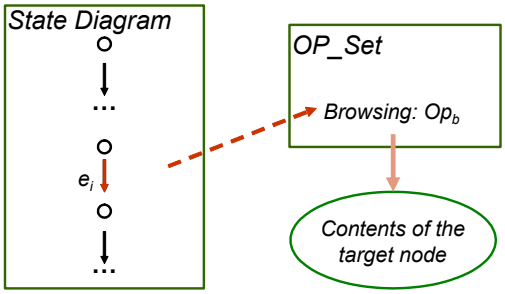


Fig. 5.2.13 An exploring service is a browsing service.

By Def. 23 of [53], a browsing service is associated with an underlying hypertext construct. Event e_i in Fig. 5.2.13 models a path through a website a user follows to access the target node. It invokes operation OP_b defined in Def. 2. The output of OP_b is the contents of the target node. A sequence of target nodes, v_{t_0}, v_{t_1}, …, v_{t_i}, …, v_{t_k}, associated with a sequence of events, e_0, e_1, …, e_i, …, e_k, is denoted as a user's navigation path π.

Since $\forall v \in V_q$, $v \in V_H$, each node v in a user's information need $((V_q, E_q), L_q, F_q)$ is included in the hypertext, therefore, the user's navigation path π is a (sub)structure of the hypertext. If $\exists v \in V_q$, $v \in V_H$, and the contents associated with v are related to the user's information need, then we consider the user is satisfied with the browsing service. Otherwise, either contents in the hypertext or contents associated with nodes in the user's navigation path π are not related to the user's information need. Both lead to an unpleasant browsing experience. In the latter case, there may be a node associated with relevant contents in the hypertext; however, the vertex does not belong to V_q (i.e., the node is not included in the user's navigation path π). Therefore, the user is lost in the hypertext when browsing.

Proof: given a node v_s and a link (v_s, v_t), where v_s, $v_t \in V_q$ and v_s, $v_t \in V_H$, according to Def. 2, each link traversal event e_i is associated with a function OP_b: $E_H \rightarrow 2^{Contents}$, $OP_b(v_s, v_t)=P(v_t)$, and P is a function which associates a node of the hypertext with the node context, i.e., given a node v_s and a link (v_s, v_t) retrieves the contents of target node v_t. Therefore, the exploring service is a browsing service.

Theorem 3: If $\forall e_i$, the associated operations with event e_i are OP_s followed by OP_{clu}, then an exploring service is a post-retrieval clustering service.

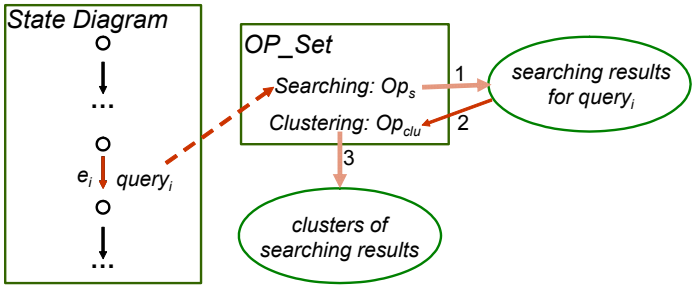


Fig. 5.2.14 An exploring service is post-retrieval clustering.

The event e_i in Fig. 5.2.14 associates operation OP_s, as indicated by the one dashed arrow. The two patterned arrows (numbered 1 and 3, respectively) point to the output of OP_s and OP_{clu}, respectively. Searching results for $query_i$ is the input to OP_{clu}, (shown by the arrow numbered 2).

Proof: $\forall q \in Q$, there is a searching scenario returning C_{retr}, a set of retrieved digital objects, and a post-retrieval clustering scenario having a final event of returning the matching function value $sim_{clu} = OP_{clu1}(cluster_x , cluster_y)$ for each pair of clusters and the contents of the clustering results $\{OP_{clu}((cluster_x , cluster_y), sim_{clu})\}$, where $cluster_x$, $cluster_y \subseteq C_{retr}$.

Note that if $C_{retr}=C$, then the exploration service also is a clustering service on a whole collection C.

Lemma 1: Let $Espa_{browse}= (Q_{browse}, Contents_{browse}, OP\_Set_{browse})$ be the exploration space of a browsing service $Eser_{browse}$, where $OP_b \in OP\_Set_{browse}$; let $Espa_{search} = (Q_{search}, Contents_{search}, OP\_Set_{search})$ be the exploration space of a searching service $Eser_{search}$, where $OP_s \in OP\_Set_{search}$; let π be a user's navigation path, a sequence of target nodes consisting of v_{t_k-1} and v_{t_k} as the last two nodes; let Π be as a set of π, where π is a user's navigation path, a sequence of target nodes, $v_{t\_0}, v_{t\_1}, \ldots, v_{t\_i}, \ldots, v_{t\_k}$, associated with a sequence of events, $e_0, e_1, \ldots, e_i, \ldots, e_k$.

1) $Eser_{browse}$ can be converted to $Eser_{search}$, denoted $Eser_{browse} \Rightarrow Eser_{search}$, if

   $\exists M_1: \Pi \rightarrow Q_{search}$, such that $\forall \pi \in \Pi$, $M_1(\pi) = q \in Q_{search}$, and $OP_b (v_{t\_k-1}, v_{t\_k}) = P(v_{t\_k}) = OP_s(q)$, where $P(v_{t\_k})$ is the contents associated with the last target node $v_{t\_k} \in V_{q\ browse}$ and

   $OP_s (q)$ is the content associated with retrieved digital objects for query $q \in Q_{search}$.

2) $Eser_{search}$ can be converted to $Eser_{browse}$, denoted $Eser_{search} \Rightarrow Eser_{browse}$, if

$\exists M_2$: $Q_{search} \rightarrow \Pi$, such that $\forall q \in Q_{search}$, $M_2(q) = \pi \in \Pi$, and $OP_b\ (v_{t\_k-1}, v_{t\_k}) = P(v_{t\_k}) = OP_s(q)$, where $P(v_{t\_k})$ is the contents associated with the last target node $v_{t\_k} \in V_{q\ browse}$ and $OP_s\ (q)$ is the content associated with retrieved digital objects for query $q \in Q_{search}$.

Proof:

1) $\forall \pi \in \Pi$, $M_1(\pi) = q \in Q_{search}$, and the results of the operations associated with each link traversal event are the contents of retrieved digital objects for query q. Therefore, $Eser_{browse} \Rightarrow Eser_{search}$.

2) $\forall q \in Q_{search}$, $M_2(q) = \pi \in \Pi$, and the results of the operations associated with the event of issuing query q are the contents of the last target node v_{t_k} in the user's navigation path π. Therefore, $Eser_{search} \Rightarrow Eser_{browse}$.

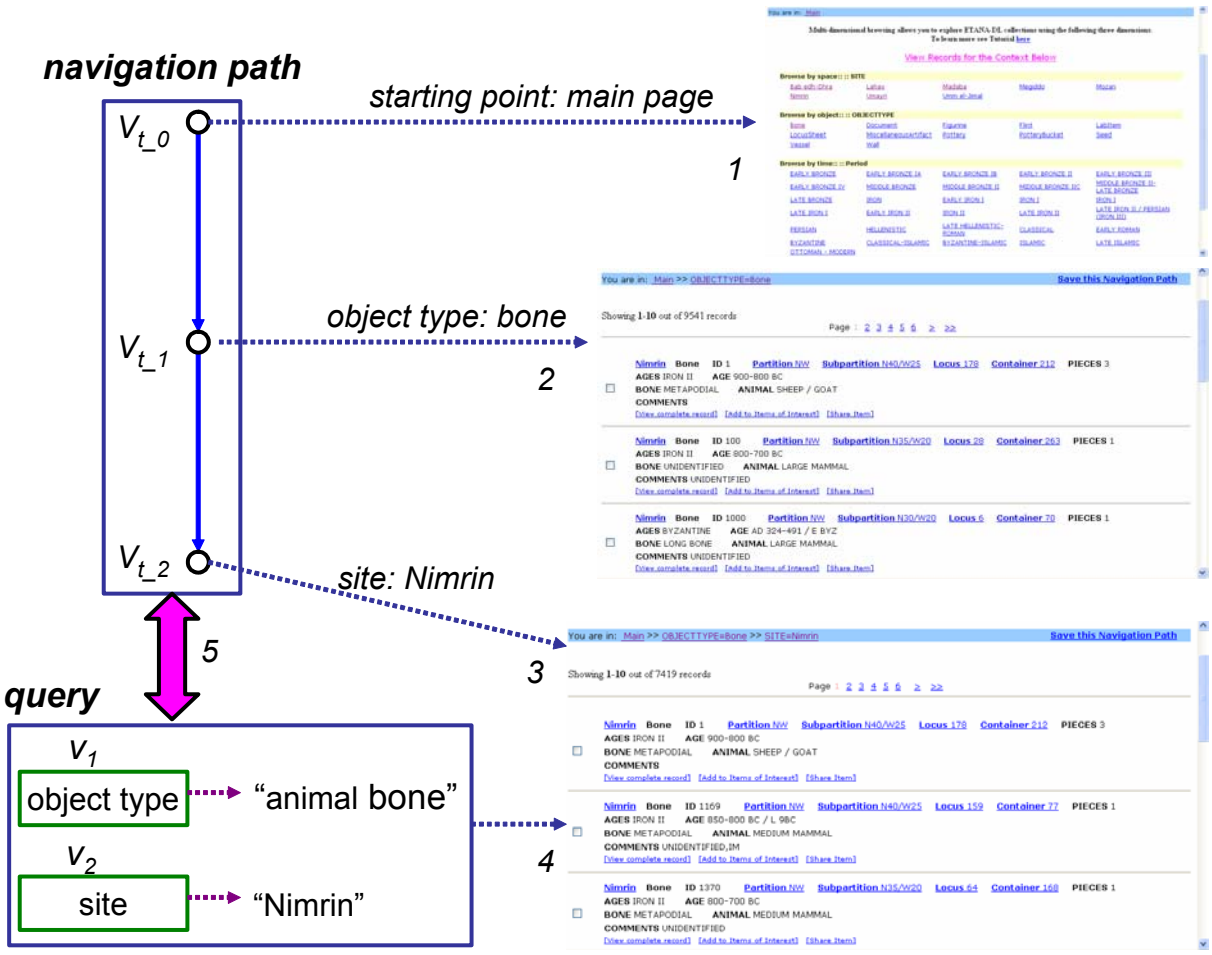


Fig. 5.2.15 Example of mapping between navigation path and a structured query

Example:

The rectangle shown in Fig. 5.2.15 represents a navigation path of a user. It consists of three nodes. The first one v_{t_0} is the starting point, which is associated with the main page of ETANA-DL's multi-dimensional browsing interface (illustrated by an arrow numbered 1 in Fig. 5.2.15); the second one v_{t_1} is related to a page about 9541 bone records (illustrated by an arrow numbered 2 in Fig. 5.2.15); the page about 7419 bone records from the Nimrin site is the contents of the last target node v_{t_1} (illustrated by an arrow numbered 3 in Fig. 5.2.15) and it displays the retrieved results for a structured query (illustrated by an arrow numbered 4 in Fig. 5.2.15). The bidirectional arrow numbered 5 in Fig. 5.2.15 denotes that the navigation path and the structured query can be mapped to each other.

Lemma 2: Given $Q_{search}=\{q_1, q_2, ..., q_n\}$, $\Pi=\{\pi_1, \pi_2, ..., \pi_n\}$, where π_i is a user's navigation path, a sequence of target nodes consisting of $v_{i\_t\_k-1}$ and $v_{i\_t\_k}$ as the last two nodes, $OP_s(q_i) = OP_b(v_{i\_t\_k-1}, v_{i\_t\_k}) = contents_i \in 2^{Contents}$ (see Def. 2), $OP_s^{-1}(contents_i)=q_i$, and $OP_b^{-1}(contents_i)=\pi_i$, then $\exists M_1$, $\exists M_2$, $Eser_{browse} \Rightarrow Eser_{search}$, and $Eser_{search} \Rightarrow Eser_{browse}$.

Proof:

1) $\exists M_1$, $\forall \pi_i \in \Pi$, $M_1(\pi_i) = OP_s^{-1}(OP_b(v_{i\_t\_k-1}, v_{i\_t\_k})) = OP_s^{-1}(contents_i) = q_i$, therefore, according to Lemma 1, $\exists M_1: \Pi \rightarrow Q_{search}$ and $Eser_{browse} \Rightarrow Eser_{search}$.

2) $\exists M_2$, $\forall q \in Q_{search}$, $M_2(q_i) = OP_b^{-1}(OP_s(q_i)) = OP_b^{-1}(contents_i) = \pi_i$, therefore, according to Lemma 1, $\exists M_2: Q_{search} \rightarrow \Pi$ and $Eser_{search} \Rightarrow Eser_{browse}$.

As shown in Fig. 5.2.16, both "$query_i$" and "π_i" are associated with the same results, therefore, $\exists M_1: M_1(query_i) = \pi_i$, $\exists M_2: M_2(\pi_i) = query_i$, $Eser_{browse} \Rightarrow Eser_{search}$ and $Eser_{search} \Rightarrow Eser_{browse}$.

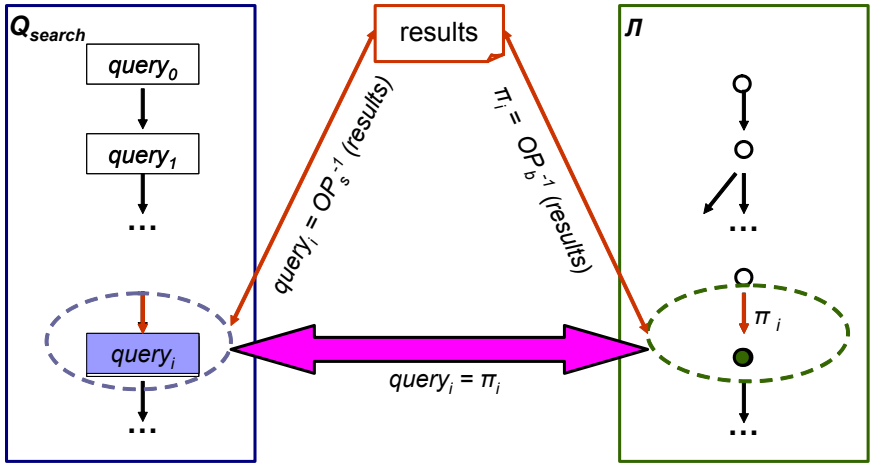


Fig. 5.2.16 "*query*$_i$" and "π_i" are associated with the same results.

Example:

There are 3 records about acacia seed in ETANA-DL. They are associated with the query "acacia seed" (represented as *'+objectType:seed +name:acacia'* based on the query language of ETANA-DL) and with a navigation path (represented as *'Main >> OBJECTTYPE=Seed >> Name=Acacia'*) as shown in Fig. 5.2.17. In this example, searching results are displayed along with the query q and browsing results are displayed along with the corresponding navigation path π. Therefore, there exist function M_1 and M_1, such that $OP_s^{-1}(results) = q$ and $OP_b^{-1}(results) = \pi$, where *results* are represented by the 3 acacia seed records.

Fig. 5.2.17 Example of Lemma 2

PESTO [19], DataWeb [96], and MIX [98] are cases where browsing can be converted to searching. Because of PESTO's "query-in-place" paradigm, DataWeb's hierarchically browsing, and MIX's navigation commands of the standard DOM API, the navigation paths of each of them can be mapped to queries. Therefore, $Eser_{browse} \Rightarrow Eser_{search}$.

Lemma 3: Let $Espa_{postBrowse} = (Q_{postBrowse}, Contents_{postBrowse}, OP\_Set_{postBrowse})$ be the exploration space of an exploring service $Eser_{postBrowse}$ occurring ***after*** $Eser_{browse}$, where $Contents_{postBrowse} = OP_b(v_{t\_i-1}, v_{t\_i})$ is the contents associated with edge (v_{t_i-1}, v_{t_i}), v_{t_i-1} and v_{t_i} are the last two nodes of a user's navigation path $\pi_i \in \Pi$ in $Eser_{browse}$, $C_{postBrowse}$ is a set of digital objects associated with $Contents_{postBrowse}$, and $OP_b \in OP\_Set_{postBrowse}$. According to Theorem 1, $Eser_{postBrowse}$ is a searching service (i.e., browsing service $Eser_{browse}$ leads to searching service $Eser_{postBrowse}$), if $\exists OP_s$: $(Q_{postBrowse} \times C_{postBrowse}) \times Sim_s \rightarrow 2^{Contents\ postBrowse}$, where $Sim_s = \{OP_q(q,$

$do)| q \in Q_{postBrowse}, do \in C_{postBrowse}\}$, where OP_q: $Q_{postBrowse} \times C_{postBrowse} \rightarrow R$ is a matching function that associates a real number with $q \in Q_{postBrowse}$ and a digital object $do \in C_{postBrowse}$.

Proof: $\forall q \in Q_{postBrowse}$, $\{OP_s((q, do), Sim_s)\}$ is the contents of the retrieved digital objects for query q, where $Sim_s = OP_q(q, do)$, therefore, by Theorem 1, $Eser_{postBrowse}$ is a searching service.

The switch from browsing to searching in PESTO [19], DataWeb [96], and MIX [98] can be generalized as shown in Fig. 5.2.18. The arrow numbered 1 points to the browsing results associated with navigation path π_i. Since π_i and $query_i$ can be mapped to each other in these systems as discussed before (indicated by the arrow numbered 3), they are associated with the same results, $Contents_{postBrowse}$. Therefore, the arrow numbered 2 also points to $Contents_{postBrowse}$. After browsing, a user searches $Contents_{postBrowse}$ for a new query $query_{i+1}$. Searching results for $query_{i+1}$ then is a subset of $Contents_{postBrowse}$. It is illustrated as the circle and pointed to by the arrow numbered 4 in Fig. 5.2.18. Therefore, $query_{i+1}$ is a new query refined from $query_i$ as indicated by the arrow numbered 5. So switching from browsing to searching in this situation is a query refining or expansion process.

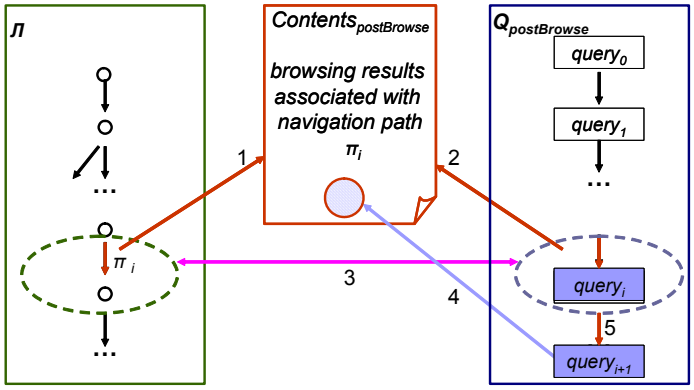


Fig. 5.2.18 "$query_{i+1}$" is refined from "$query_i$" after browsing.

Lemma 4: Let $Espa_{postRetr} = (Q_{postRetr}, Contents_{postRetr}, OP\_Set_{postRetr})$ be the exploration space of an exploring service $Eser_{postRetr}$ occurring **after** $Eser_{search}$, where $Q_{postRetr} = \{((V_{q\ post\ Retr}, E_{q\ post\ Retr}), L_{q\ post\ Retr}, F_{q\ post\ Retr}), Contents_{q\ post\ Retr}, P_{q\ post\ Retr}\}$ (see Def. 1), $Contents_{postRetr}$ is associated with C_{retr}, a set of retrieved digital objects for query $q \in Q_{search}$ in $Eser_{search}$. According to Theorem 2, Lemma 1, and Lemma 2, $Eser_{postRetr}$ is a browsing service (i.e., searching service $Eser_{search}$ leads to browsing service $Eser_{postRetr}$), if $OP\_Set_{postRetr} = \{OP_s,$

$OP_{clu}\}$, $cluCon_{retr} = \{OP_{clu}((cluster_x, cluster_y), sim_{clu}) \mid cluster_x, cluster_y \subseteq C_{retr}\} = \{cluCon_{retr\_1}, cluCon_{retr\_2}, ..., cluCon_{retr\_i}, ..., cluCon_{retr\_z}\}$ is the contents of clustered retrieved results, where $sim_{clu} = OP_{clu1}(cluster_x, cluster_y)$ (see Def. 2), $\Pi = \{\pi_1, \pi_2, ..., \pi_i, ..., \pi_z\}$, where $\pi_i = (v_0, v_i)$ is a navigation path consisting of only two nodes, $v_0, v_i \in V_{q\ post\ Retr}$, and $\exists M_{b\_cluster}: \Pi \rightarrow cluCon_{retr}$.

The event e_i of issuing $query_i$ triggers the operation OP_s, as indicated by the dashed arrow numbered 1 in Fig. 5.2.19. The patterned arrow numbered 2 denotes the output of OP_s, i.e., $Contents_{postRetr}$ (searching results for $query_i$). OP_{clu} takes $Contents_{postRetr}$ as input and yields as output the contents of clusters as shown by the arrows numbered 3 and 4. The arrow numbered 5 represents the mapping from each navigation path to the contents of a cluster. Therefore, the contents of the last target nodes of these navigation paths are the contents of clusters and the mapping function $M_{b\_cluster}$ can be viewed to be OP_b for browsing.

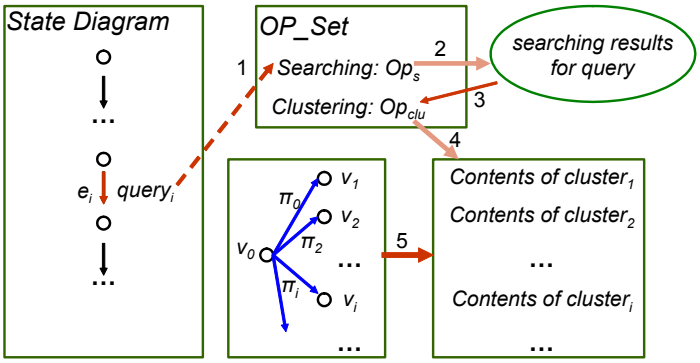


Fig. 5.2.19 Switch from searching to browsing

Proof: $\forall v \in V_{q\ post\ Retr}$, $v \in V_H$, and $\forall e_i$, the associated operation with event e_i is $OP_b((v_0, v_i)) = M_{b\_cluster}(\pi_i) = cluCon_{retr\_i}$, where v_i is the target node of π_i, therefore by Theorem 2, $Eser_{postRetr}$ is a browsing service.

Categorizing or clustering searching results is a case of switching searching to browsing. ScentTrails [104] can be viewed as a special case as $|cluCon_{retr}|=1$, i.e., each cluster is a singleton having one item from the retrieved result list.

Theorem 4: If $\forall e_i$, the associated operations with e_i are OP_s followed by OP_{viz}, then an exploring service is a post-retrieval visualization service.

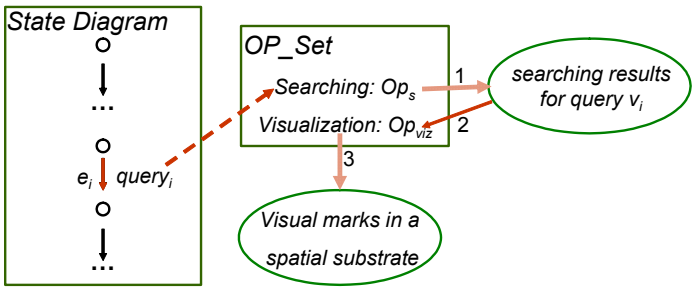


Fig. 5.2.20 An exploring service is a visualization service.

The event e_i in Fig. 5.2.20 associates operation OP_s, as indicated by the dashed arrow. The two patterned arrows (numbered 1 and 3, respectively) point to the output of OP_s and OP_{viz}, respectively. Searching results for $query_i$ is the input to OP_{viz} (shown by the arrow numbered 2).

Proof: $\forall\, q \in Q$, there is a searching scenario returning a set of retrieved digital objects C_{retr} and a post-retrieval visualization scenario having a final event of visually mapping a set of digital objects (or each digital object) of C_{retr} to a visual mark with visual properties in a spatial substrate of n dimensions.

If n=2, it is 2-D visualization; if n=3, it is 3-D visualization. If C_{retr}=C, the exploring service also is a visualization service for a whole collection. If $\exists M_2(q)$, the exploring service is a visualization service for browsing. Vector graphics and raster display are two different types of display used for representation. Virtually all modern current computer video displays translate vector representations to a raster format.

5.3. Case Study: Exploring Services in ETANA-DL

Our theory-based approach to describing DL exploring services allows us to understand browsing and searching in a new way. It guides us to design and implement exploring services for an archaeological DL, ETANA-DL. ETANA-DL is an integrated archaeological DL supporting integration of a number of (ETANA) sites in the Near East. It integrates searching and browsing, allowing users to browse at will and shift between browsing and searching seamlessly. It also provides a visual interface applying data analysis and information visualization techniques to help archaeologists test hypotheses and extend the understanding of past (material) cultures and environments.

In this section, we first introduce a multi-dimensional browsing service, which can actually be considered as a searching service according to Lemma 2. We then illustrate how ETANA-DL

combines browsing and searching in two ways. The first way extends and empowers the multi-dimensional browsing. It can be viewed as query refining and extension based on Lemma 3. Organizing searching results hierarchically is the second way. Both ways allow seamless transition between browsing and searching, as suggested by Lemma 4. We finally describe the visualization service, which integrates browsing and searching into a single visual interface, as suggested by Theorem 4.

5.3.1. Multi-dimensional browsing

Multi-dimensional browsing allows users to move along any of the navigational dimensions, or a combination thereof. By navigational dimension we mean a hierarchical structure used to browse digital objects. Digital objects in ETANA-DL are various archaeological data, e.g., figurine images, bone records, locus sheets, and site plans. They are organized by different hierarchical structures (e.g., animal bone records are organized based on sites where they are excavated, temporal sequence, and animal names). These hierarchical structures contain one or more hierarchically arranged categories that are determined by the elements of the global schema of ETANA-DL. In addition to this, they can be refined based on taxonomies existing in botany and zoology, or from classification and description of artifacts by archaeologists.

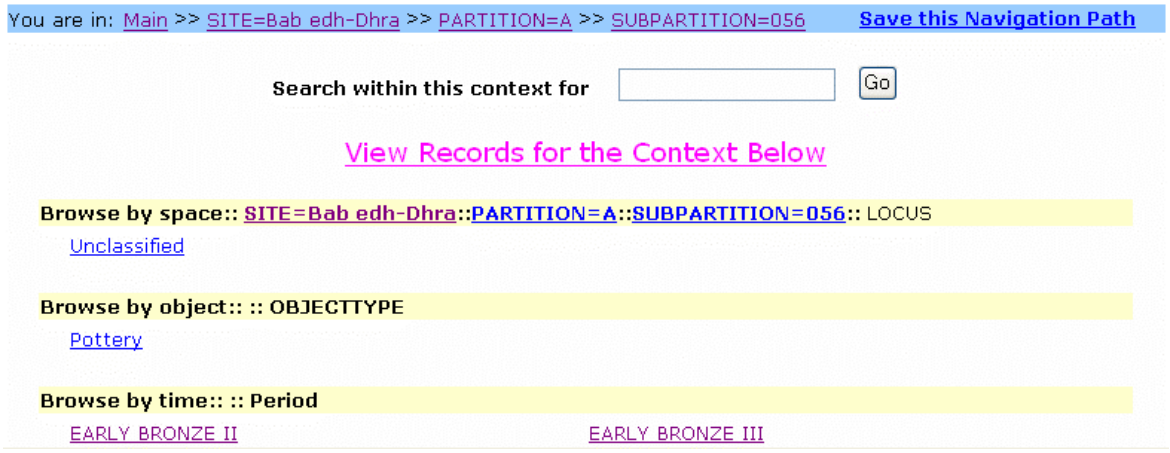


Fig. 5.3.1 Multi-dimensional browsing interface

Typical DLs provide a directory-style browsing interface (as in Yahoo! or Open Directory), with levels in the hierarchy displayed as clickable category names and DL items in that category shown below them. Though some DLs (such as CITIDEL) allow users to browse

through several dimensions, they are limited in that users cannot navigate through all dimensions simultaneously, or across different dimensions.

Fig. 5.3.2 Save current navigation path for later use and view records

In ETANA-DL, a user can browse through three dimensions: space, object, and time. She can start from any of these dimensions and move along by clicking. The scenario shown in Fig. 5.3.1 tells that she is interested in the artifact records from the tomb numbered 056 in area A of the Bab edh-Dhra site. The clickstream representing her navigation path is denoted 'Site=Bab edh-Dhra >> PARTITION=A >> SUBPARTITION=056'. While the navigation path is within the first dimension, it is associated with the other dimensions. The second dimension shows there is only one type of objects, i.e., pottery, from that particular location. The third dimension presents the two time periods associated with those pottery records. Hence, the dynamic coverage and hierarchical structure of those dimensions yields a learning and exploration tool. The user can navigate across dimensions. By clicking "EARLY BRONZE II" in the third dimension, she can view all her interested artifact records from the EARLY BRONZE II period.

Her current navigation path (see the top of Fig. 5.3.2) can be saved for later use. It can be considered as a surrogate for a query for the records in that particular location and time period. Therefore, according to Lemma 2, the multi-dimensional browsing service can be viewed as searching, i.e., browsing $\Rightarrow$ searching.

5.3.2. Browsing and searching integration

1. Search within browsing context

Searching within a browsing context blends querying and browsing and is reminiscent of IBM's PESTO GUI for "in-place querying" [19]. The main idea is that browsing will present a useful starting point for active exploration of an answer space. Subsequent browsing and searching is employed to refine or enhance users' initial, possibly under-specified, information needs.

Browsing context is associated with a user's navigation path. Browsing results within a certain browsing context is defined as a set of records (web pages), e.g., there are 35 pottery records within the browsing context represented by the navigation path 'Site=Bab edh-Dhra >> PARTITION=A >> SUBPARTITION=056'. Assume a user wants to find saucer records in the set of 35 pottery records. She types "saucer" in the search box as shown in Fig. 5.3.3. According to Lemma 3, she switches from browsing to searching, and searching then is a natural extension of browsing. Since the navigation path is a surrogate of a query, searching within a browsing context can be viewed as query refining.

Fig. 5.3.3 Search saucer records

2. Organize searching results hierarchically

Eighty eight equus records are retrieved through the basic searching service (see a query named “equus” in Fig. 5.3.4). They are organized into three dimensions after the user clicks the button “View search results hierarchically” (see Fig. 5.3.5). The user starts browsing and then selects “Nimrin” in the first category to view the records. Thirty six equus records are displayed as shown in Fig. 5.3.6. According to Lemma 4, she switches from searching to browsing. During the next exploring stage of browsing, she can search as illustrated in the previous section. Therefore, she switches seamlessly between browsing and searching, to specify her information needs.

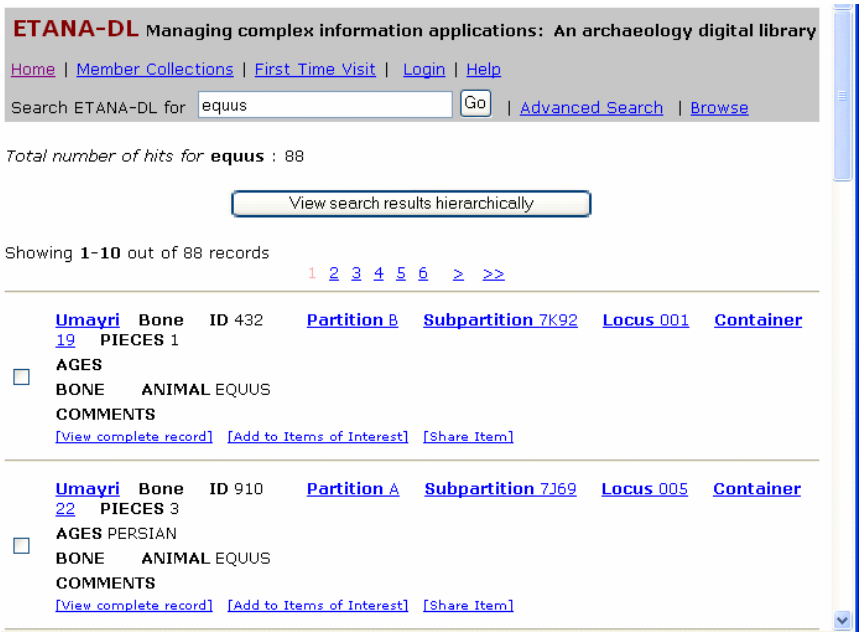


Fig. 5.3.4 Equus records are retrieved through basic searching

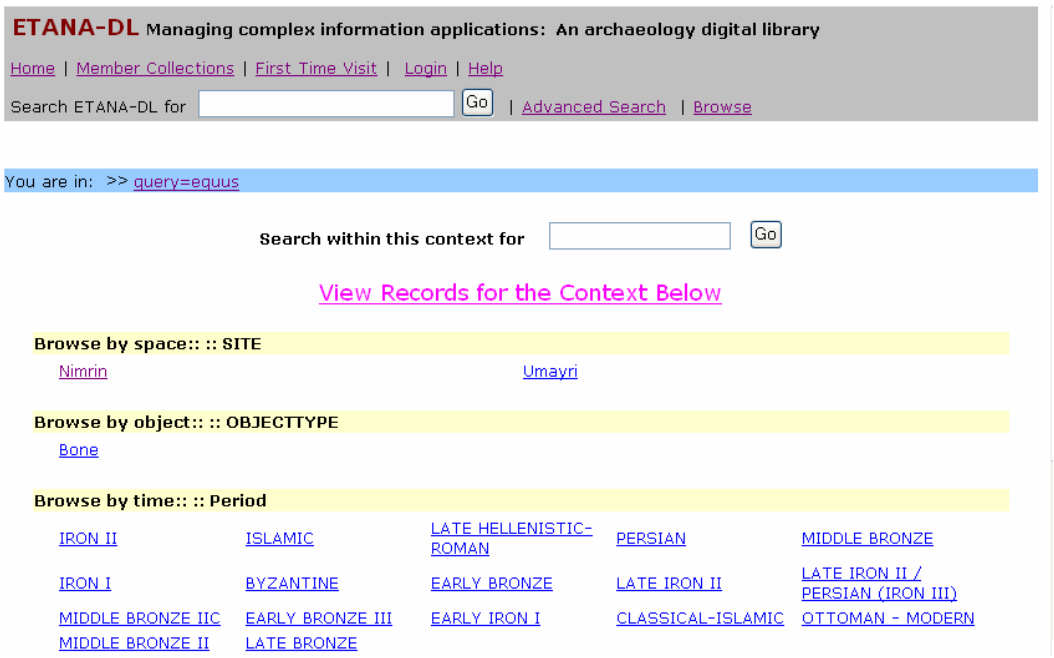


Fig. 5.3.5 Retrieved equus records are organized into 3 dimensions

Fig. 5.3.6 Browse the 36 equus records from the Nimrin site after searching

5.3.3. Browsing, searching, and visualization integration

While the searching and browsing services provided by ETANA-DL allow users to access primary archaeological data, their help with comprehending specific archaeological DL phenomena is limited when vast quantities of data are harvested into ETANA-DL. Fortunately, visual interfaces to DLs enable powerful data analysis and information visualization techniques to help archaeologists test hypotheses and extend the understanding of past (material) cultures and environments. Data generated from the sites' interpretation then provides a basis for future work, including publication, museum displays, and, in due course, input into future project planning. Thus, we developed EtanaGIS and EtanaViz to support visually exploring archaeological DLs. EtanaGIS allows integration of Geographic Information System (GIS) data

for related archaeological sites into ETANA-DL. It provides a web-based GIS portal to allow users to spatially explore ETANA-DL. Details of EtanaGIS can be found at http://etana.dlib.vt.edu/~etana/Viz/EtanaGIS.pdf.

In this chapter, we focus on EtanaViz. It integrates searching, browsing, clustering, and visualization into a single interface. Its initial interface is shown in Fig. 5.3.7. The top left of the screen is a query box. On the top right is a hyperbolic tree showing hierarchical relationships among excavation data based on spatial, temporal, and artifact-related taxonomies. A node name represents a category, and a bubble attached to a node represents a set of archaeological records. The size of a bubble attached to a node reflects the number of records belonging to that category. The hyperbolic tree supports "focus + context" navigation; it also provides an overview of records organized in ETANA-DL. It shows that the records are from seven archaeological sites (the Megiddo site has the most) and are of eight different types.

According to Def. 2, a cluster (group) of records is associated with a vector of two elements, i.e., name and size of the cluster; a cluster is mapped to a visual mark: bubble (circle); the name and size of the cluster are mapped to two visual properties: label and size of the bubble, respectively.

EtanaViz supports exploring to gain insights, as is illustrated in the following example scenarios.

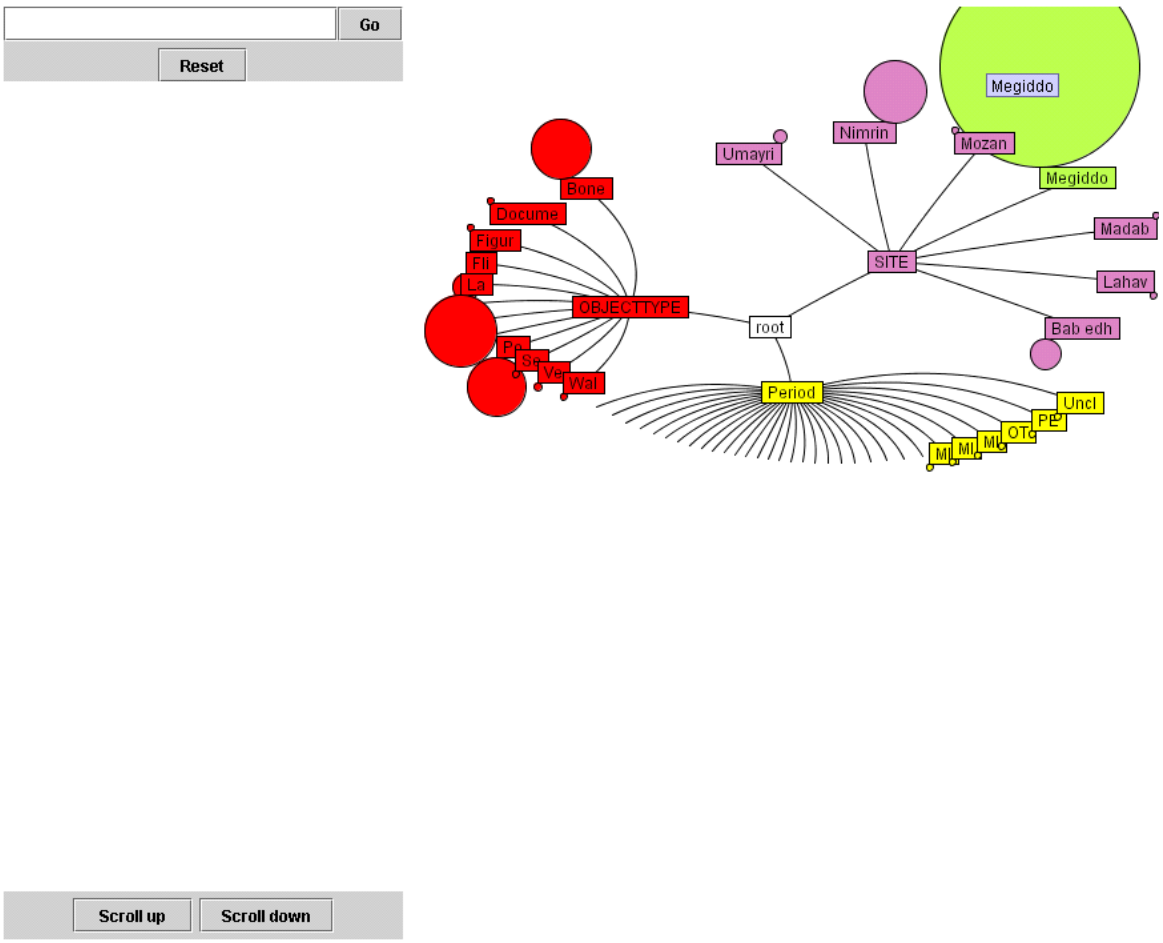


Fig. 5.3.7 Initial interface of EtanaViz

A user is interested in excavated animal bones from site Nimrin, located in the Jordan Valley. She inputs query "SITE=Nimrin&OBJECTTYPE=Bone". The results are displayed as a hyperbolic tree, as illustrated in Fig. 5.3.8. All excavation bone records are grouped into cultural phases (time periods). They are Middle Bronze, Iron I, Iron II, Persian, Late Hellenistic/Roman, Byzantine, Islamic, and Ottoman-Modern. The records also are classified by archaeological site organization and animal categories. The user wants to know the number of bone records for each period. She left clicks a node labeled "MIDDLE BRONZE" in the hyperbolic tree and selects the "add to compare…" option to view total bones throughout the Middle Bronze Age. This causes a bar to be displayed in a chart below the hyperbolic tree and an entry to be listed on the left. She continues to add more bars to view bones throughout the entire time sequence of Tell Nimrin occupation. When she moves the mouse over a bar, a tool tip shows the number of animal bones for the corresponding culture phase.

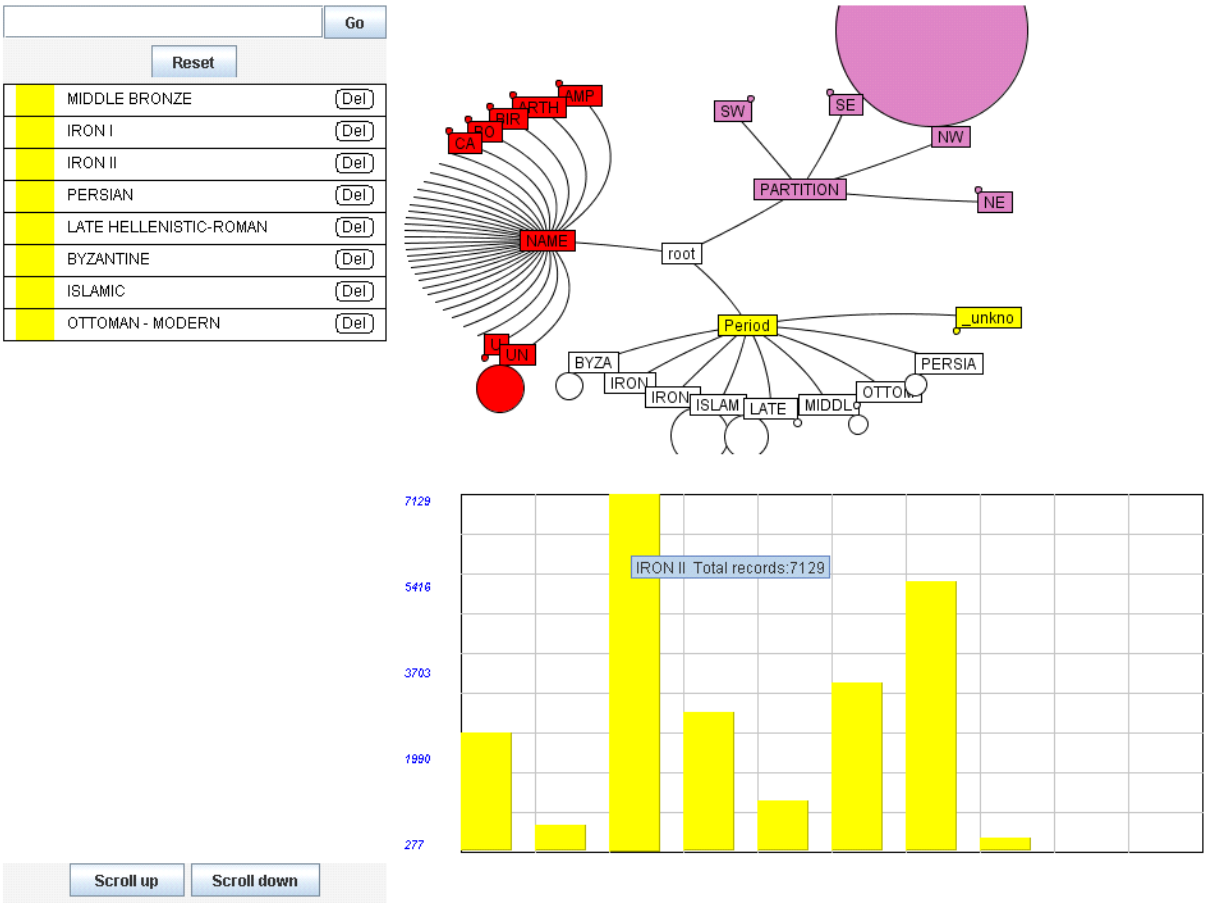


Fig. 5.3.8 Total number of animal bones across Nimrin culture phrases

She continues navigating the hyperbolic tree. She left clicks a node labeled “SUS” and selects the “add to view distribution…” option. She then left clicks the “BOS”, “CAPRA”, and “OVIS” nodes to show how those animal bones constitute the identified bones in each culture phrase. Eight stacked bars representing percentages of those bones are displayed, and four entries with different colors are included in the list on the left of the screen (see Fig. 5.3.9).

The color of the entry can be changed to help distinguish different categories. It is always synchronized with the color in the stacked bars. The red bars (at the bottom of the stacked bars), representing sus (pig) bones, show that sus constitute 4.71% of the Middle Bronze Age faunal assemblage, but less than 1% at the beginning of the Iron Age. The user is wondering why the percentage for pig bones drops dramatically over time at Tell Nimrin. She may hypothesize that the reasons are probably twofold: 1) the introduction of religious taboos against eating pork, and 2) increased demand for clean water sources as human populations grew at Nimrin [146].

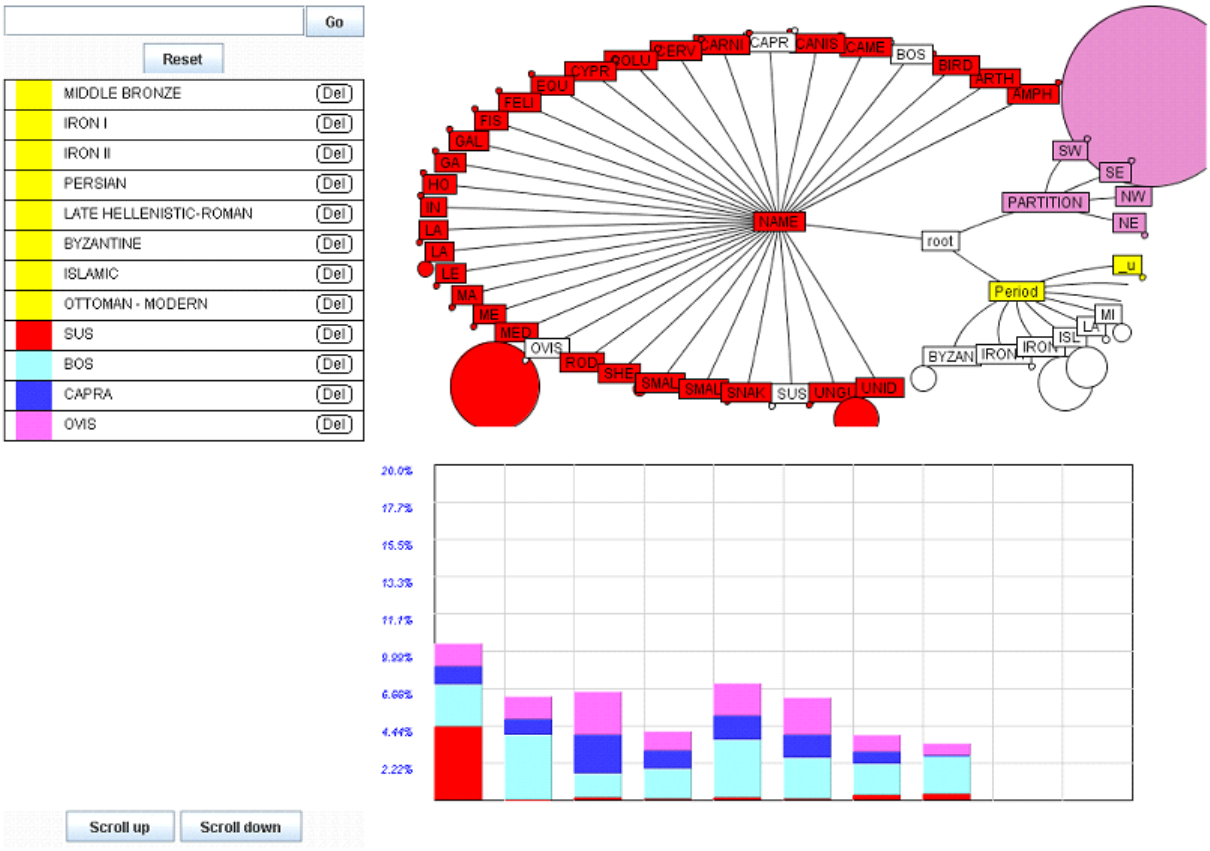


Fig. 5.3.9 Percentages of animal bones across Nimrin culture phrases

Light blue bars (on top of the red sus bars) represent bos (cattle) bones percentages. Two light blue bars are higher than the others. They are corresponding to the Iron II and Late Hellenistic/Roman culture phrases, respectively. The user, considering that cattle figure most prominently during these periods, may suggest improved grazing conditions in the Jordan Valley during that time.

Pink bars and blue bars (the top two of the stacked bars) represent ovis (sheep) bones and capra (goat) bones, respectively. Pink bars are slightly higher than blue bars. This means that ovis bones slightly outnumber capra bones across culture phrases of Tell Nimrin. This would suggest that past environmental conditions in the Jordan Valley provided enhanced forage for sheep while goats would have been employed as browsers on drier vegetation. Relatively stable percentages of slightly higher sheep populations versus those of goats may indicate that favorable environmental conditions and environmental or cultural desertification did not greatly impact the agrarian way of life at Tell Nimrin on the banks of the Jordan, over time [146].

The user may be interested in animal bones excavated from other sites. By repeating the interaction with EtanaViz, as described before, she starts to analyse animal bones excavated from the Umayri site. She also can make inter-site comparisons.

5.3.4. ETANA-DL exploring services formative evaluation

In fall 2005, we conducted a formative user study for ETANA-DL. Many of the findings reported in the usability evaluation are already influencing the iterative design and implementation of ETANA-DL to achieve the usability goals. In this section, instead of listing all the findings, we focus on only the findings that help validate the hypotheses related to browsing, searching, and visualization. Those findings agree with comments found in the related literature as described in section 2 of Chapter 3, i.e., users significantly prefer integrated browsing and searching to browsing (or to searching). When browsing service can be mapped to searching (browsing $\Longrightarrow$ searching, as discussed in section 5.3.1), saved navigation paths can be views as searching history, which keeps track of user's information needs and helps reduce time and effort to achieve information seeking goals.

1. Evaluation methods and procedure

Twenty eight graduate students from the computer science department at Virginia Tech participated in the evaluation experiment, which was posted with instructions online at http://etana.vt.edu:8080/etana/servlet/surveyTasks?submit_start. The experiment was conducted through four sessions. Each user was required to:

1) learn the online tutorial of ETANA-DL;

2) complete a pre-evaluation questionnaire;

3) perform tasks using ETANA-DL. After completion of each task, he (she) was asked to fill out a task-related questionnaire and give comments.

4) provide subjective reactions using post-evaluation survey forms.

Users' interactions with ETANA-DL were logged by ETANA-DL. The time to complete each task and the error rate for each task were measured automatically. At the completion of all the tasks, users were asked to measure the exploring services on a 5-point scale, where 1= poor, and 5=excellent. Our reason for measuring users' impression about ETANA-DL services (five of them are listed in Table 1) stems from the following two pre-experimental hypotheses:

• Users significantly prefer integrated browsing and searching to browsing.

• Users significantly prefer integrated browsing and searching to searching.

2. Results and discussion

The median values for measuring users' impressions regarding five of the ETANA-DL services are shown in Table 4. Browsing, searching, and EtanaViz received four points on a 5-point scale, while searching within browsing context (abbreviated as SWBC) and saving navigation path (abbreviated as SNP) services received 4.5. Users commented that they appreciated SWBC and SNP because "SWBC is simple enough to understand and an excellent way of narrowing down a search…browsing through the different levels can be time consuming, so if we know that we will want to go to a given context a lot, it is useful to just be able to click on a link of SNP to get back to our context of interest…"

Table 4 Impression about ETANA-DL services (mean value)

Browse	**Search**	**EtanaViz**	**Save navigation path (SNP)**	**Search within browsing context (SWBC)**
4.0	4.0	4.0	4.5	4.5

We also did t-tests on the following four hypotheses.

• H1: Impression about SWBC is larger than that for browsing at significance level 0.05.

• H1: Impression about SWBC is larger than that for searching at significance level 0.05.

• H1: Impression about SNP is larger than that for browsing at significance level 0.05.

• H1: Impression about SNP is larger than that for searching at significance level 0.05.

The above four hypotheses were all accepted. The first two accepted hypotheses are associated with the two pre-experimental hypotheses mentioned above, i.e., users significantly prefer integrated browsing and searching to browsing (or to searching). These results agree with comments found in the related literature as described in section 2 of Chapter 3. To probe the last two hypotheses, we analysed four of the seventeen tasks performed by users. For four tasks, users were asked to give the number of records retrieved, for specific information needs. The followings are those four tasks.

1. Use browsing to give the total number of pottery records excavated from tomb 007 in area A of the Bab edh-Dhra site.

2. Use searching to tell how many equus bones are from the Umayri site.

3. Use browsing to tell how many equus bones are from the Nimrin site.

4. Use saved navigation paths to give the total number of pottery records excavated from tomb 056 in area A of the Bab edh-Dhra site.

Fig. 5.3.10 shows the average time for each of the four tasks.

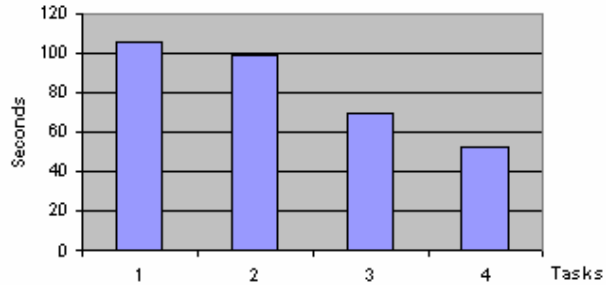


Fig. 5.3.10 Average time on tasks

Task 4 was completed significantly faster than either task 1 or task 2, at significance level 0.05. This showed that reusing saved navigation paths really improves users' performance. It saved users time during exploration. While similar information needs (e.g., task 1, 2, and 4) can be achieved through different ways (browsing, searching, or SNP), SNP keeps track of users' navigation history and helps reduce time and effort to achieve information seeking goals.

We expected that users would complete task 4 significantly faster than task 3. We also thought users would spend about the same time to complete the similar tasks, i.e., task 2 and task 3. However, our experimental results were somewhat surprising in that the average time on task 4 was not significantly less than that on task 3, and the average time on tasks 2 and task 3 was different. We did some follow-up interviews to probe the reasons. Our log file indicated that one user spent more than five minutes to complete task 4. We found that he got disconnected during the online experiment for task 4. Though task 2 and task 3 have similar information needs, users found it was difficult to find appropriate keywords to complete task 2, therefore, more time was needed to try more queries. We believe that since users got experience and developed a searching strategy when doing task 2, they completed task 3 faster than task 2 (task 3 was performed after task 2).

Because our new service to organize searching results hierarchically was not implemented before we conducted the evaluation, we cannot yet report data about its efficiency and effectiveness. However, there is already evidence that information access is improved by posting search hits against an interactive tree structure [31, 32].

Chapter 6. Assessment of Integrated DLs

In this chapter, we seek to answer the question: what qualifies as a successful integrated DL? This leads to another question: what is a successful DL? We synthesize diverse research in the area of DL quality models, information systems (IS) success and adoption models, and information-seeking behavior models, to present a more integrated view of the concept of DL success. Such a multi-theoretical perspective, considering user community participation throughout the DL development cycle, supports understanding of social aspects of DLs and changing needs of users interacting with DLs, and helps determine when and how quality issues can be measured and how possible quality problems can be prevented. Based on our proposed DL success model, we move one step further to define: "what is a successful minimal union DL?" The assessment of a union DL then is illustrated through application to an archaeological DL, ETANA-DL.

6.1. Assessment of DLs: What is a Successful Digital Library

Hundreds of millions of dollars have been invested since the early 1990s in research and development related to digital libraries. Further R&D is needed worldwide [83] if the tremendous potential of DLs is to be achieved. Hence, determining the key characteristics of DL success is of the utmost importance.

What qualifies as a successful DL, and what does not? As this question begins to be analyzed, more questions arise. Who is the intended user of a DL? What is the user's goal for using the DL? What are individual organizations trying to get from their DLs?

For several years, researchers from various disciplines have studied different perspectives of DL success and have generated many interesting yet often isolated findings. For example, [76] presented results from a study of information use by lawyers, and [7] studied journalists' use of electronic information resources. Some findings have provided different although sometime overlapping perspectives on how to evaluate DLs. One of them is the DL quality model developed by Gonçalves [50]. For each key concept of a minimal DL, [50] defined a number of

dimensions of quality and proposed a set of numerical measurements for those quality dimensions.

Though many would consider a DL to be a type of information system, it often is forgotten that there is a long tradition in IS research of evaluating the success of a generic IS. A variety of measures have been used. Two primary research streams, the user satisfaction literature and the technology acceptance literature (i.e., the technology acceptance model, or TAM) have been investigated. User satisfaction is based on users' attitudes toward a system. We define satisfaction as a user's affective state presenting an emotional reaction to an entire DL and the consequence of the user's experiences during various information-seeking stages. Therefore, understanding of the changing needs of users interacting with the DL and users' information-seeking behavior during these stages is needed [2]. Fortunately, the information seeking behavior of academic scholars has been studied for decades, and many models [149] have been generated.

Because attitude leads to action, user satisfaction prompts user acceptance of the system and leads to higher system usage. A system succeeds when its intended users use it as frequently as needed.

This section is organized as follows. Section 6.1.1 presents prior work, which is theoretical background for our proposed research model. Section 6.1.2 proposes a DL success model. Section 6.1.3 presents a case study of our proposed model in a domain specific DL.

6.1.1. Prior work

Library and information science researchers, such as those attending the workshop on "Evaluation of Digital Libraries," have investigated the evaluation of DLs [3, 15, 44, 90, 132, 143]. Saracevic [132] was one of the first to consider the problem. According to his analysis, there are no clear agreements regarding the elements of criteria, measures, and methodologies for DL evaluation. The challenge is made more complex by the various classes of users. DL users have varying levels of expertise, both in information seeking and in domain knowledge [14]. In an attempt to fill some gaps in this area, Fuhr et al. [44] proposed a description scheme for DLs based on four dimensions: data/collection, system/technology, users, and usage. However, work on usability of DLs has lagged, especially regarding the non-user-oriented technical topics in the DL literature [58]. There are a few reported studies: inspection of NCSTRL was described in

[58]; evaluation of ACM, IEEE-CS, NCSTRL, and NDLTD digital libraries was reported in [73]; evaluations of ADL and ADEPT were documented in [62] and [20], respectively.

Theories regarding DLs (e.g., the 5S framework [53]), IS success and adoption, and information-seeking behavior have evolved in parallel. They provide foundations that can be integrated to help answer the question: "what is a successful DL?". The prior research suggests the need for a more comprehensive view of DL success. There have been calls for research to empirically validate and extend IS success and adoptions models into varying contexts [145]. Motivated by these calls for research and the increasing number of DL users with varying skills and from different backgrounds and cultures, we seek to answer the question: what is the appropriate model of DL success from the perspective of end users (DL patrons)?

The most prominent IS success models existing in the literature today are by Venkatesh [145], DeLone [28], and Seddon [136]. They are discussed in the following subsections 2 and 3.

1. System Usage as a Success Measure

System usage has been considered to be an important indicator of IS success in a number of empirical studies, for many systems. However, simply measuring the amount of time a system is used does not fully capture the relationship between usage and the realization of expected results. The nature, extent, quality, and appropriateness of the system use also should be considered [27]. The nature of system use should be addressed by determining whether the full function of a system is being used for the intended purpose. Thus, Doll et al. [30] developed a multidimensional measure of system usage based on the system's nature and purpose. Young et al. [152] suggested that full functional use of an e-commerce system should include informational use, transactional use, and customer service use. With regard to the extent of use, Lassila et al. [84] identified various states of system utilization based on the use or nonuse of basic and advanced system capabilities. Accordingly, we believe that log analysis could be beneficial to the measurement of DL usage.

2. Technology Acceptance Model (TAM): Predict Intention to Use

TAM provides predictions of intention to use by linking behaviors to attitudes that are consistent with system usage, in time, target, and context. Over the past decade, TAM has led to

a large number of empirical tests, comparisons, model variants, and model extensions. It has been widely applied to understand the attitude one holds about the use of IS.

Venkatesh's model [145] predicted behavioral intention to use a system and is a unified model building on the eight most popular behavioral IT acceptance theories in the literature. It consists of four core determinants of intention and usage, as shown in Fig. 6.1.1. They are: performance expectancy (defined as the degree to which individuals believe that using the system will help them improve their job performance), effort expectancy (the degree of ease associated with the use of the system), social influence (the degree to which an individual perceives that society as a whole will view her favorably for using the system), and facilitating conditions (the degree to which an individual believes that an organizational and technical infrastructure exists to support the system).

Despite its predictive ability, TAM provides only limited guidance about how to influence usage through system design and implementation. Venkatesh et al. stressed the need to extend the TAM literature by explicitly considering system and information characteristics and the way in which they might indirectly influence system usage.

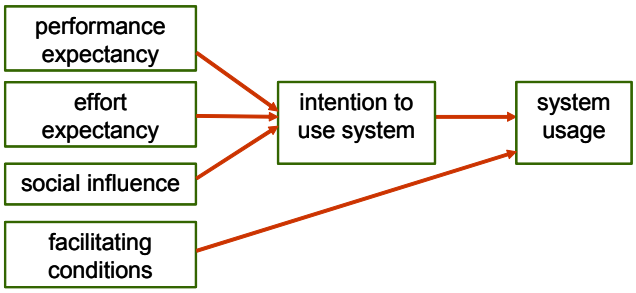


Fig. 6.1.1 Venkatesh's model

3. Satisfaction: Attitude toward the System

In contrast to TAM, system and information characteristics have been core elements in the literature on user satisfaction. The DeLone study [28] is one of the first attempts at a comprehensive review of the literature on IS success. It organized a broad base of diverse research (180 articles) and presented a more integrated view of IS success. DeLone's model consists of six interdependent constructs for IS success: system quality (SQ), information quality (IQ), use, user satisfaction, individual impact, and organization impact (see Fig. 6.1.2). It identified IQ and SQ as antecedents of user satisfaction and use.

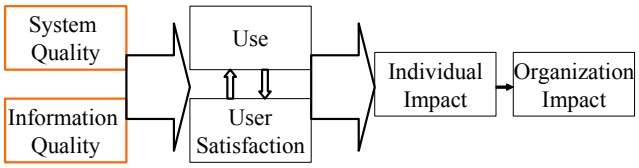


Fig. 6.1.2 DeLone's IS success model

Seddon suggested that DeLone et al. tried to do too much with their model, and as a result, the model is confusing and lacks specificity [136]. Seddon's major contribution is a re-specified model of IS success. Seddon defined success as a measure of the degree to which the person evaluating the system believes that the stakeholder is better off. The model shows that both perceived usefulness and user satisfaction depend on IQ, SQ, and benefits (see Fig. 6.1.3). Both DeLone and Seddon made an explicit distinction between information aspects and system features as determinants of user satisfaction.

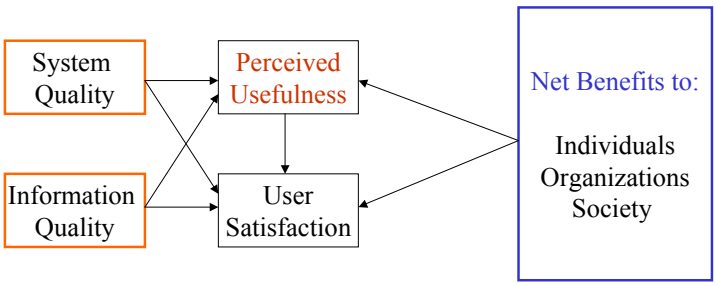


Fig. 6.1.3 Seddon's IS success model

4. Information Seeking Behavior: Identify Temporal Users' Information Needs

Satisfaction is a consequence of the user's experience during various information-seeking stages. The changing needs of users interacting with the DL should be identified. Therefore, understanding of users' information-seeking behavior is needed.

The information seeking behavior of academic scholars has been studied for decades, and many models [149] have been generated. Among them are Ellis's model [33, 34] and Kuhlthau's model [75, 76]. These two models are based on empirical research and have been tested in subsequent studies.

Ellis's model includes six generic features: staring, chaining, browsing, differentiating, monitoring, and extracting [33, 34]. These six features are coded from E1 through E6 as shown

in Fig. 6.1.4. As of 2002, there were more than 150 papers that cite Ellis's information-seeking behavior model of social scientists [93]. Most of the information-seeking behavior features in Ellis's model are now being supported by capabilities available in Web browsers. For example, an individual could begin surfing the Web from one of a few favorite pages or sites (staring); follow links to related resources in both backward and forward linking directions (chaining); scan the Web sites of sources selected (browsing); bookmark useful sources for future reference (differentiating); subscribe to email based services that alert the user to new information or developments (monitoring); and search a particular source or site for all information on that site on a particular topic (extracting) [21, 22].

Kuhlthau's model complements that of Ellis by attaching to stages of the information seeking process the associated feelings, thoughts and actions, and the appropriate information tasks [75, 76]. The stages of Kuhlthau's model are: initiation, selection, exploration, formulation, collection, and presentation. They are coded from K1 through K6 as shown in Fig. 6.1.4. Kuhlthau's model is more general than that of Ellis in drawing attention to the feelings associated with the various stages and activities. It has been applied to support for learning from DLs [92].

5. DL Quality Model within the 5S Framework

Gonçalves [50] proposed formalizations for quality dimensions and indicators for DL deeply grounded in the 5S DL theory [53]. Key concepts of a minimal DL (digital object, metadata specification, collection, catalog, repository, and service) are considered along with corresponding quality dimensions and measurements. Gonçalves discussed connections between his proposed dimensions of DL quality and an expanded version of a workshop's consensus view of the information life cycle model [16].

6.1.2. DL success model

We further connect Gonçalves' DL quality model and the information life cycle model with Ellis' and Kuhlthau's information-seeking behavior models as shown in Fig. 6.1.4. The outer arrows in Fig. 6.1.4 indicate the life cycle stage (active, semi-active, and inactive) for a given type of information. The innermost portion of the cycle has four major phases of

information use or process: information creation, distribution, seeking, and utilization. Each major phase is connected to a number of activities.

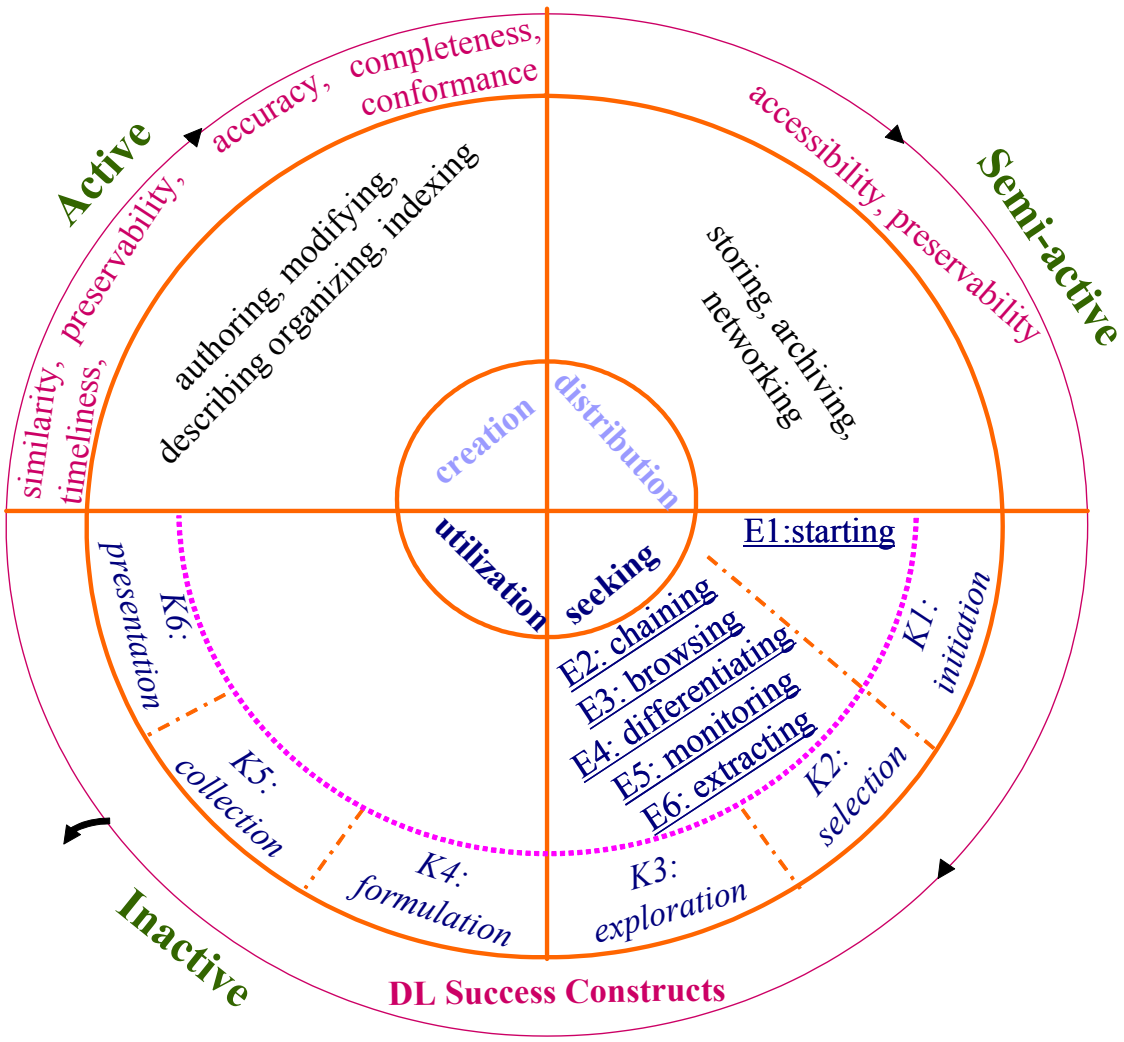


Fig. 6.1.4 Connection of DL quality model with information life cycle and information seeking behavior models

Gonçalves stated that his work took a very system-oriented view of the quality problem and partially neglected its usage dimension. Our goal is to define the success of DL from an end user perspective; hence we focus on the 'seeking' and 'utilization' stages. Behaviors occurring at the 'seeking' phase and 'utilization' phase are elaborated in Fig. 6.1.4 by Ellis' (E1-E6, see words with underlines) and Kuhlthau's models (*K1-K6*, see words in italic font). Each dimension of quality is associated with a corresponding set of activities. Quality dimensions associated with the seeking and utilization phases are related to constructs of the DL success model.

Our proposed DL success model consists of four interrelated and interdependent constructs based on the previously discussed theoretical methods. The general proposition of our model is that DL satisfaction and the intention to (re)use a DL are dependent on four constructs: information quality, system quality, performance expectancy, and social influence (see Fig. 6.1.5). Arrows in Fig. 6.1.5 indicate that a construct is affected by each construct that points to it. IQ and SQ can be found in the IS success literature, while performance expectancy and social influence can be found in the IT adoption literature. Since our model incorporates TAM, it is a predictive model, i.e., it can be used to predict intention to (re)use. We think determinants of success are goal and user specific. A measurement instrument of "overall success" based on items arbitrarily selected from the four constructs is likely to be problematic. Individual measures from the four constructs should therefore be combined systematically to create a comprehensive measurement instrument.

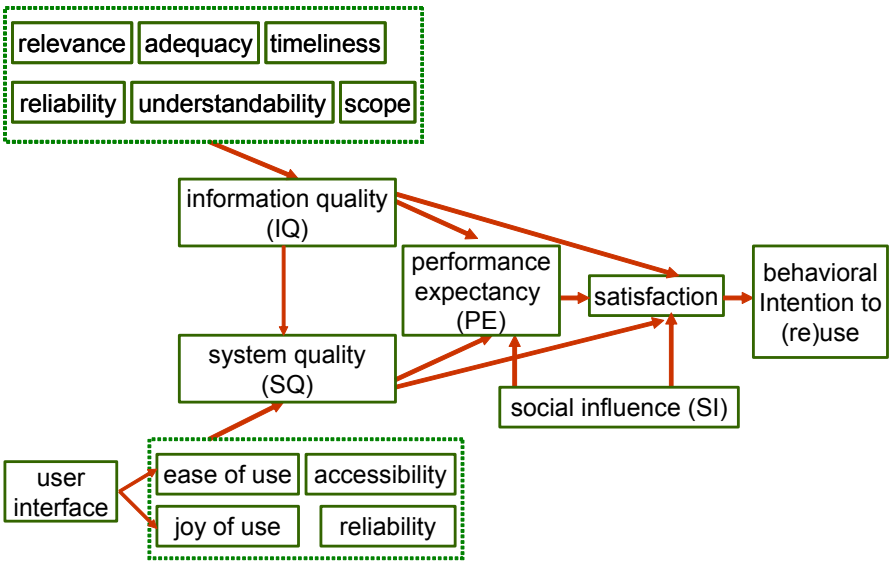


Fig. 6.1.5 DL success model

1. Information Quality (IQ)

Information in DLs can be classified from two different perspectives, the DL developers' view and the DL patrons' (end users') view. Five main concepts related to DL information within the 5S framework [53] are: repository, collection, metadata catalog, digital object, and metadata specification (see Fig. 6.1.6). A DL repository consists of a set of collections, each of which is a set of digital objects. Samples of digital objects can be electronic theses (or

dissertations) and records of artifacts (such as bones, seeds, and figurines) excavated from an archaeological site. Each digital object is assigned associated metadata specification(s), which compose the metadata catalog.

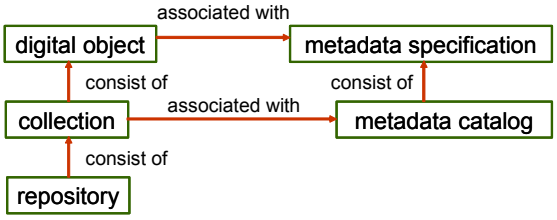


Fig. 6.1.6 Concepts related to DL information

While the dimensions of quality for each of the five concepts are defined in [50] and listed in the left part of Fig. 6.1.7, they do not fully differentiate end users from DL developers. We group the five concepts into three categories and develop six items (factors) to measure the quality for each of the three categories for end users, as shown in the right part of Fig. 6.1.7. The dashed arrows illustrate that parts of the quality dimensions discussed in [50] are associated with the six items measuring DL IQ.

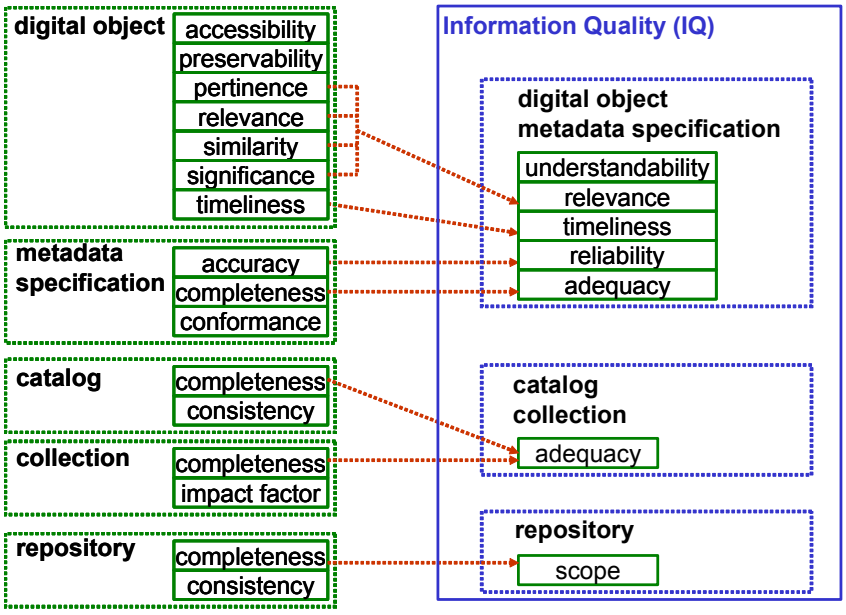


Fig. 6.1.7 DL information quality (IQ) measurement

a) Digital object and metadata specification:

Accuracy and completeness are defined in [50] as quality dimensions for metadata specification, however, they are absent in the quality dimensions list for a digital object. This leads to two other quality measures for digital object and metadata specification: adequacy and reliability. Adequacy indicates the degree of sufficiency and completeness. Reliability indicates the degree of accuracy, credible, and consistency.

Relevance is concerned with such issues as relevancy, pertinence, and the applicability of the information. Pertinence and relevance for digital objects are measured with Boolean values (0 or 1) in [50]. They are a subjective judgment by users in a particular context. We use relevance to measure the quality of both digital object and metadata specification. Significance of a digital object defined in [50] reflects relevance to a user's needs or particular user requirements. Therefore, significance can be partially mapped to relevance. Similarity metrics defined in [50] reflect the relatedness among digital objects. If one of the digital objects is a user's information need, then similarity is associated with the relevance item (factor).

Timeliness is concerned with the currency of the information. Understandability encompasses variables such as being clear in meaning and easy to understand.

Preservability as an important digital object quality property needs to be identified and added by DL developers; however, it may not be visible to DL patrons. The accessibility of a digital object is managed by DL services, so it is used to measure DL services instead of information. Therefore, preservability and accessibility are not included in the six items for DL IQ that are shown in Fig. 6.1.7.

b) Metadata catalog and collection

Adequacy is used to measure the degree of sufficiency and completeness of DL metadata catalogs and collections.

c) Repository

Scope evaluates the extent and range of the repository. These address the breadth of information and the number of different subjects. According to [50], a repository is complete if it contains all collections it should have. Therefore, completeness defined in [50] is associated with scope.

2. System Quality (SQ)

Dimensions of quality for DL services are classified as internal (e.g., top three entries) or external (e.g., bottom three entries) in [50], as shown in the dashed box in Fig. 6.1.8. We focus on the external view, concerned with the use and perceived value of these services from the end users' point of view. They relate to DL system quality (SQ) and performance expectancy (discussed in the next subsection) as indicated by the three dashed arrows in Fig. 6.1.8. We develop four items to measure DL SQ.

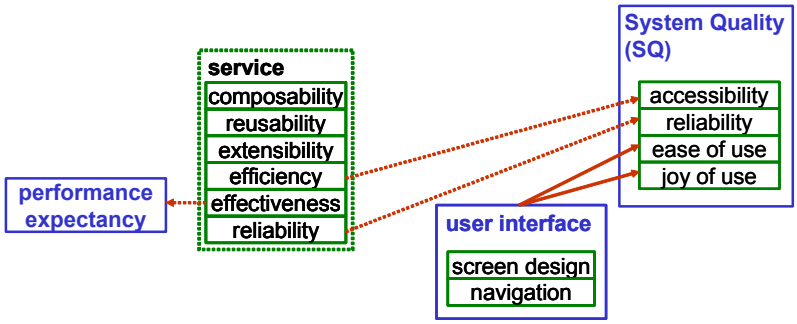


Fig. 6.1.8 DL service quality (SQ) measurement

Prior research subscales for accessibility include system responsiveness and loading time. The accessibility of a DL refers to not only its speed of access and availability but also to its information (e.g., digital objects and metadata accessibility). Efficiency defined in [50] is measured in terms of speed; it is associated with service accessibility. A DL needs to be reliable, which means that it is operationally stable.

Ease of use is concerned with how simple it is for users to (learn to) use DLs. Joy of use is about the degree of user pleasure. These two items are affected by the user interface through navigation and screen design as indicated by the two solid arrows shown in Fig. 8. Navigation is concerned with evaluating the links to needed information that are provided on the various pages of a DL website. Screen design is the way information is presented on the screen. It affects both ease of use and joy of use. Having an organized and well-designed screen aids users in locating relevant information more easily, while an attractive user interface helps increase joy of use. The role that emotion and aesthetics play for people has been pointed out as a central psychological issue by [68]. It had not been applied to usability until [78] noted that the ergonomics quality of a product does not implicitly coincide with the usability perceived by the user. Although we have a

common idea that aesthetic objects should be symmetric, balanced, or well proportioned, there is no general instruction set prescribing how to create aesthetic interfaces [55]. [69] proposes some helpful methods and guidelines for the design of pleasurable products, whereas [60] underlines the importance of the hedonic quality (novelty, originality) of a software product. Work on more formalized approaches is in progress, to refine mathematical concepts [87] and to find mathematical explanations [9] for accessing aesthetic phenomena.

3. Performance Expectancy (PE)

Performance expectancy (see Fig. 6.1.5) is defined as the degree to which users believe that a specific DL will help them gain advantage in accomplishing their desired goal. In [145], it consists of five constructs: perceived usefulness, extrinsic motivation, job-fit, relative advantage, and outcome expectations. Perceived usefulness is the degree to which people believe that using a particular system will enhance their performance. Extrinsic motivation refers to users' desire to perform an activity because it is perceived as being essential for achieving a desired outcome. Job-fit indicates how the capabilities of an information system enhance an individual's performance in accomplishing a specific task. Relative advantage is the degree to which using the system is perceived as better than alternative methods. Outcome expectations relate to the consequences of the behavior.

4. Social Influence (SI)

Social influence (see Fig. 6.1.5) is concerned with a user's perception that other important people favor a particular DL. Many studies have been done in the marketing domain on the role of social influence, e.g., it was found that brand loyalty is positively related to frequency of website usage. It also was found that a positive response to an organization website will lead to a positive view of the brand (organization). Accordingly, it seems appropriate to consider social influence on DL usage. As reported in [143], DL visibility is considered as an important factor that may lead to greater user acceptance of DLs. Potential users may not be aware of the benefits of using the DL, or even its existence. Increasing DL visibility can help users perceive the DL as more useful, although it will not increase the functionality of a DL. This is called an exposure effect, as mere exposure to an object may change an individual's attitude toward the object in a positive way [153].

6.1.3. Case study

As part of the requirements analysis for an archaeological DL, ETANA-DL [137], email interviews with five prestigious archaeologists, and face to face workplace interviews with eleven archaeologists (including three of the five interviewed by email) were conducted. Subsequent formative evaluation studies were carried out to improve system design. In this section, we associate the four constructs of the model discussed in the previous section with the activities occurring in the seeking and utilization phases (see the innermost portion of the cycle in Fig. 6.1.4) by analyzing the results of the interviews and the formative usability studies. These results are shown in Table 5 and may help distinguish issues that are generic across domains, from those that are domain specific.

Table 5 DL success constructs associated with seeking and utilization phases

DL success Construct	*seeking phrase*		*utilization phrase*		
	starting (E1/K1)	*selection exploration (E2-E6)/(K2-K3)*	*formulation (K4)*	*collection (K5)*	*presentation (K6)*
social influence	*DL visibility*				
information quality		*adequacy, scope*	*accuracy*		
system quality		*ease of use joy of use (interface)*	*accessibility*	*accessibility*	*accessibility*
performance expectancy		*usefulness (interface)*			

1. Seeking phase

- E1/K1

"starting" activity in Ellis' model ('initiation' stage in Kuhlthau's model) is usually at the beginning of information seeking. It may help one 'recognize' a need for information. Users' information needs may be initiated by a specific active task or condition, or by requirements identified passively.

Social influence, such as regarding DL visibility, is associated with this stage. Methods to increase DL visibility in the archaeological domain include:

1) Publicize the existence of a DL: One archaeologist said that "… the turning point for the DL will be when someone has demonstrated in a print publication how ETANA-DL helped in their research …". Some recommended more international collaboration. Some

suggested that ETANA-DL may consider collaboration with JADIS (Jordanian Archaeological Data Information System) to increase its visibility. Since JADIS is one of the main Jordanian cultural resource management systems, connecting ETANA-DL with JADIS will allow basic survey and overall information on Jordanian archaeology to be combined with ETANA-DL's more in-depth coverage.

2) Provide a DL alert service (e.g., press alerts): Archaeologists may want alerts when new artifacts from others arise on their subjects of interests.

- (E2-E6) / (K2-K3)

These five feature activities in Ellis's model ('chaining', 'browsing', 'differentiating', 'monitoring', and 'extracting') occur in the 'selection' and 'exploration' stages in Kuhlthau's model. In the 'selection' stage, a general area for investigation is identified (located). The appropriate task at this point is to fix the general topic of exploration. Exploration has many cognitive requirements similar to browsing and search tasks. IQ, SQ, and PE are associated with these stages. Regarding IQ, adequacy (degree of sufficiency and completeness) of DL collections and metadata catalogs and scope of DL repository should be considered. Some archaeologists pointed out that "Ideally, the system would include as many types of data as possible, from text summaries to photos, maps, and other visuals."

Regarding SQ and PE, interface plays a major role in influencing the usefulness, easy of use, and joy of use. The quality of the DL interface makes a significant contribution to a usable DL, and interface problems often are cited by non-users as a major reason for not using electronic information retrieval systems [41]. As a virtual intermediary between users and a DL, the interface is the door through which users access a DL. The interface characteristics (screen design and navigation) that affect DL usability include those commonly found in most web GUIs, as well as the ones specific to archaeological DLs.

1) Screen design: The way that information is arranged on the screen can influence the users' interaction with DLs beyond the effect of the information content. Some archaeologists suggested that "… the interface needs to be more visually stimulating … should allow to browse visual stacks of the digital library…". Another issue to be considered for screen design is the wording for labelling. In the archaeological domain, an example could be the terminology for periodization schemas. There are different

periodization schemas based on political, historical, or cultural events. The archaeologists found it difficult to use a single "standard" periodization schema.

2) Navigation: The navigation should enable archaeologists to explore a DL without having to keep an auxiliary memory aid like a yellow pad at hand.

2. Utilization phase

Information management and utilization was not identified as a category in Ellis's study of social scientists. On the other hand, the last three stages in Kuhlthau's model involve organizing information into a coherent structure.

- K4

The formulation stage is identified as conceptually the most important step in the process [75]. Users focus on a more specific area within the topic and make sense of (or interpret) information in the light of their own needs. A guiding idea or theme emerges which is used to construct a story or narrative, or to test a hypothesis. This formulation also will guide the users in selecting appropriate information.

Research has considered the process of interpreting documents (e.g., reading and annotating them) rather than simply locating them [13]. Within the archaeological domain, archaeologists formulate a personal perspective or sense of meaning from the encountered information. However, they usually undertake interpretation offline. Access to primary data and data analysis services provided by DLs enable archaeologists to make interpretations online, if they change work habits. Alternatively, exporting of results to files or into special formats like spreadsheets may be helpful to support subsequent offline management, processing, visualization, and reporting.

Some sample factors affecting formulation are as follows.

1) Information accuracy: Formulation is associated with verifying the accuracy of the information found. Archaeologists need reputable (trusted) information or information analysis to support interpretation.

2) Information accessibility: It defines how much effort (time) is required to find (locate) the information needed. In the archaeological domain, primary data usually is available to researchers outside a project (site) only after substantial delay. Some archaeologists said

that "… ETANA-DL would be a very efficient way to disseminate and share our research, and in turn, we could utilize the work of others as much as possible."

- K5

In the collection stage, information is gathered to support the chosen focus. Information accessibility is very important as discussed above.

- K6

During this final stage, presentation, ideas, focus, and collected resources are organized for publishing and sharing. Some archaeologists suggested making arrangements with the publishers of obscure journals to include their publications in ETANA-DL. Others found it useful for ETANA-DL to provide a discussion forum to share interpretation of annotated items.

6.2. Assessment of a Minimal Union DL

The quality dimensions and corresponding objective measurements for the two major concepts of a minimal union DLs, i.e., union catalog and union service, are shown in Table 6. They are discussed in Section 6.2.1 and Section 6.2.2 respectively. To measure the success of a union DL from the end user's perspective, we use the DL success model proposed in Section 6.1.2. The mappings between 5S based DL quality dimensions, 5S and DL concepts, and DL success constructs are shown in Table 7.

6.2.1. Union catalog: consistency and completeness

Gonçalves [50] introduced consistency and completeness as two metrics for a metadata catalog. A metadata catalog is consistent if every descriptive metadata specification is connected to only one digital object through a unique identifier. A catalog is complete if it has at least one descriptive metadata instance for every digital object in the collection. According to Definition 2 in 1.3, we get

Lemma 1: A union catalog is complete if all the catalogs to be integrated are complete.

Lemma 2: A union catalog is consistent if it satisfies the following two requirements:

1) All the catalogs to be integrated are consistent;

2) Each descriptive metadata specification in the union catalog describes only one digital object.

We need to resolve the object identification problem and to provide conflict resolution when different metadata records from different DLs represent the same entity in the real world. The resolution function is defined as:

Definition 1: Let $S=\bigcup_{i=1}^{n} S_i$, where S_i is an element set of a local schema of DL_i, let V be a set of the values of the elements in S, a resolution function

$resF : (V \cup ¤) \times (V \cup ¤) \rightarrow (V \cup ¤)$, where ¤ represents the null value.

$resF(x,y) =$

$¤$, *if* $x=¤$ *and* $y=¤$;

x, *if* $x \neq ¤$ *and* $y=¤$;

y, *if* $x=¤$ *and* $y \neq ¤$;

$f(x,y)$, *if* $x \neq ¤$ *and* $y \neq ¤$;

where $f: V \times V \rightarrow V$, is an internal associative resolution function, which may be of various types depending on the type of elements in the schema, the usage of the value, and many other aspects [100]. A simple resolution might concatenate the values and annotate them with the source that provided the value. Especially conflicts in textual elements may be resolved in this way. The resolution is not completely transparent, and users (e.g., data providers and metadata librarians) are given the opportunity to resolve the conflict by their own means [100].

6.2.2. Union service

1. Mapping service: Mapping-completeness

During union catalog generation, a local schema describing metadata of an individual DL is mapped to a global schema in the union DL. Given an element set S_x of a local schema and an element set T of a global schema:

If a mapping is a 1-1 mapping (see Def. 6 in Chapter 1), let $S_x=\{ s_i \}$ and $T=\{ t_i \}$, then we define:

$MapEle_x=\{ s_i \in S_x \mid s_i$ *is such an element that can be mapped to* $t_i \in T\}$.

If a mapping is a complex mapping (see Def. 7 in Chapter 1), then we define:

$MapEle_x=\{ s_i \in S_x \mid s_i$ *is such an element that either itself or one of the formulas derived from it can be mapped to* $t_i \in (T \cup Formu_t) \}$, where $Formu_t$ are defined in Def. 7 in 1.3.

Definition 2: $Mapping\text{-}completeness = \frac{1}{n}\sum_{x=1}^{n}(|MapEle_x|/|S_x|)$, where $1 \le x \le n$, n is the total number of local schema.

With Definition 2, a union catalog generated by lossless 1-1 mapping has *mapping-completeness* 1.

2. Searching service: *Coverage*

Let $Retr(DL_i)$ and $Retr(UnionDL)$ be two sets of searching results retrieved from DL_i and *UnionDL* by query *q*, respectively. *Coverage* related to *q* is:

$$coverage_q = \frac{|\bigcup_{i=1}^{n} Retr(DL_i)|}{|Retr(UnionDL)|}$$

$coverage_q$=1 means that a user can just visit the UnionDL for one-stop shopping for the information she needs from all the DLs to be integrated. The quality dimension, *Relevance*, proposed by [50], is embedded in *coverage*.

3. Browsing service: *NavigationGain*

Let $BrowseS(DL_i)$ be a local schema of DL_i; let $BrowseS(UnionDL)$ be a global schema of *UnionDL*; let $Path(DL_i)$ and $Path(UnionDL)$ be the number of possible navigation paths provided by DL_i and *UnionDL*, respectively.

$$NavigationGain = \frac{Path(Union) - \sum_{i=1}^{n} Path(DL_i)}{\sum_{i=1}^{n} Path(DL_i)}$$

Example: Let ArchDL1 and ArchDL2 be two DLs supporting browsing of representations of excavated animal bones. As shown in Fig. 6.2.1, ArchDL1 provides browsing bones by site organization, whereas ArchDL2 provides browsing by bone names. UnionArchDL, the integrated DL built from these two ArchDLs, supports browsing both by site organization and bone names. The number of nodes in the browsing schema of ArchDL1 is denoted as *x*, and the number of nodes in the browsing schema of ArchDL2 is denoted as *y*.

The possible navigation paths for ArchDL1 are:

- Site;
- Site→Partition;

- Site→Partition→Sub-partition;
- Site→Partition→Sub-partition→Locus;
- Site→Partition→Sub-partition→Locus→Container;
- Site→Partition→Sub-partition→Locus→Container→Artifact;
- Path(ArchDL1) =x=6;
- The possible navigation paths for ArchDL2 are:
- Bone;
- Bone→BoneName;
- Path(ArchDL2) =y=2;

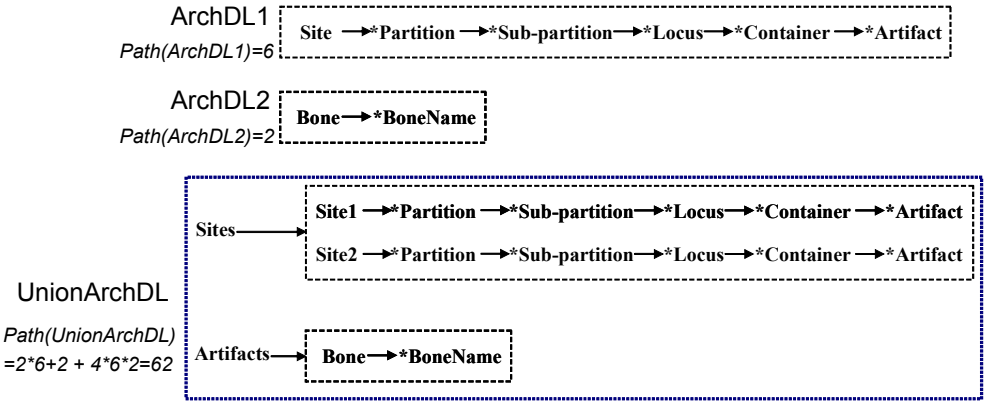


Fig. 6.2.1 Example of calculating NavigationGain

From Fig. 6.2.1, we can conclude that the numbers of total possible within-dimensional navigation paths provided by UnionArchDL is *2x+y*, whereas the numbers of total possible cross-dimensional navigation paths is *4xy*. Therefore,

Path(UnionArchDL) =(2x+y) + 4xy

$$NavigationGain = \frac{Path(UnionArchDL)}{Path(ArchDL1) + Path(ArchDL2)} - 1 = \frac{2x + y + 4x}{x + y} - 1 = \frac{2 \times 6 + 2 + 4 \times 6 \times 2}{6 + 2} - 1 = 6.75$$

Table 6 Quality dimension and measurement for minimal union DLs

Union DL Concept within the 5S Framework	Quality dimension	Measurement
Union Catalog	Consistency	# of sets of metadata specifications per digital object
	Completeness	# of digital objects without a set of metadata specifications; size of the described collection
Union Service (mapping, searching, browsing)	Mapping-completeness	# of local schema elements which have been mapped to global schema
	Coverage	# of results retrieved from a union DL; # of total results retrieved from the DLs to be integrated
	NavigationGain	# of navigation paths available in a union DL; # of navigation paths available in the DLs to be integrated

Table 7 Mappings between 5S based DL quality dimensions and IS success and adoption constructs

DL quality dimension	*DL success manifest variable*	*5S and DL concept*	*DL success construct*
accessibility *accuracy* *completeness* *consistence* *conformance* *pertinence* *preservability* *relevance* *significance* *similarity* *timeliness*	*adequacy* *relevance* *reliability* *scope* *timeliness* *understandability*	*stream, structure* *digital object* *metadata* *collection* *catalog* *repository*	*information quality (IQ)*
composability *efficiency* *effectiveness* *extensibility* *reusability* *reliability*	*accessibility* *reliability* *ease of use* *joy of use*	*society, scenario, space* *service*	*system quality (SQ)* *performance expectancy (PE)*
	DL visibility	*society*	*social influence (SI)*

6.2.3. Summary

In this chapter, we have developed a DL success model from an end user perspective by integrating various research areas. We specified factors that affect DL inform quality and system quality. Relationship between those factors and the quality indicators developed by [50] were identified. We further discussed assessment for a minimal union DL and developed indicators and measurements for quality of union catalog and union services (mapping, searching, and browsing).

Chapter 7. Conclusion

7.1. Contributions

This dissertation makes three major contributions, which address the three main research questions presented in Chapter 1 from the perspectives of modeling, implementation, and assessment.

1. We formalize the DL integration problem and propose an overall approach based on the 5S framework.
2. We apply our framework to integrate domain-specific (archaeological) DLs, illustrating our solutions for key problems in DL integration.
   2.1 We formally model archaeological DLs (ArchDLs) in the 5S framework.
   2.2 We develop the 5SSuite integration tool kit to integrate archaeological DLs, and use ETANA-DL as a case study to justify and evaluate our DL integration approach. SchemaMapper, a component of 5SSuite, is used to map and integrate local schema to global schemas, and to help create a union catalog.
   2.3 We generalize exploring services for integrated DLs and implement them based on the theorems developed within the 5S framework.
3. We propose and study some quality dimensions and indicators specific to integrated DLs and an integrated view of the concept of DL success. The integrated DL success synthesizes diverse research in the area of digital library quality models, information systems success and adoption models, and information-seeking behavior models.

7.2. Future Work

Modeling, implementation, and assessment, undertaken in connection with this dissertation, could be further studied and developed through future work.

7.2.1. Modeling

To integrate domain specific DLs, we may need to model these DLs. Future work can include modeling distributed DLs, possibly including P2P approaches as in OCKHAM (http://www.ockham.org/) and practical (traditional) DLs. Therefore our integration approach can be applied broadly, not just to archaeological DLs.

Our proposed DL success model may be extended and specified to specific DLs, e.g., MetaScholar (http://www.metascholar.org/quality_metrics/).

7.2.2. Implementation

The 5SSuite tool consists of tree components (5SGraph, SchemaMapper, and 5SGen). Each component could be further developed. Complex (one to many and many to one) mappings can be explored and the mapping component can be enhanced accordingly for SchemaMapper.

Exploring services for integrated DLs (e.g., searching, browsing, and visualization) could be improved to better support users' information seeking goals.

7.2.3. Assessment

Our proposed DL success model could be applied in various domain specific DLs, and needs empirical validation.

References

1. Adam, N.R., Atluri, V. and Adiwijaya, I. SI in digital libraries. Commun. ACM 43(6): 64-72, 2000.
2. Adams, A. and Blandford, A. Digital libraries' support for the user's 'information journey'. In Proc. JCDL 2005, June 7-11, 2005, Denver, 160-169.
3. Agosti, M., Nunzio, G.M.D. and Ferro, N. Evaluation of a digital library system. In Proc. of the DELOS WP7 Workshop on the Evaluation of Digital Libraries, Padova, Italy, October 4-5, 2004, 73–76.
4. Ahmedi, L. and Lausen, G. Ontology-Based Querying of Linked XML Documents. Semantic Web Workshop 2002.
5. Amann, B., C. Beeri, I.F. and Scholl, M. Ontology-based integration of XML web resources. In Proc. Int'l Semantic Web Conference '02, pp. 117-131, 2002.
6. Arms, W.Y., Hillmann, D., Lagoze, C., Krafft, D., Marisa, R., Saylor, J. and Terrizzi, C. A Spectrum of Interoperability: The Site for Science Prototype for the NSDL. D-Lib Magazine 8(1), http://www.dlib.org/dlib/january02/arms/01arms.html, 2002.
7. Attfield, S.J. and Dowell, J. Information seeking and use by newspaper journalists. Journal of Documentation, 59(2): 187-240, 2003.
8. Baldonado, M.Q.W. A User-Centered Interface for Information Exploration in a Heterogeneous Digital Library. *JASIS*, *51* (3). 297-310, 2000.
9. Balek, M. and Nesetril, J. Towards Mathematical Aesthetics, http://iti.mff.cuni.cz/series/files/iti231.pdf, 2004.
10. Belkin, N.J. Anomalous states of knowledge as the basis for information retrieval. *Canadian Journal of Inf. Sci.*, *5*. 133–143, 1980.
11. Belkin, N.J., Marchetti, P.G. and Cool, C. BRAQUE: Design of an interface to support user interaction in information retrieval. *Information Processing & Management*, *29* (3). 325–344, 1993.
12. Bergamaschi, S., Castano, S. and Vincini, M. Semantic integration of semistructured and structured data sources. *SIGMOD*, *28* (1). 54-59, 1999.
13. Bishop, A.P. Making Digital Libraries Go: Comparing Use Across Genres. ACM DL 1999, Berkeley, 94-103.
14. Blandford, A. and Buchanan, G. Usability of digital libraries: a source of creative tensions with technical developments. In IEEE-CS Technical Committee on Digital Libraries' on-line newsletter, 2003. http://www.ieee-tcdl.org/Bulletin/v1n1/blandford/blandford.html.
15. Borgman, C.L. Final report to the National Science Foundation. Fourth DELOS Workshop. Evaluation of Digital Libraries: Testbeds, Measurements, and Metrics. Hungarian Academy of Sciences, Computer and Automation Research Institute (MTA SZTAKI), Budapest, Hungary, 6-7 June 2002. Grant IIS-0225626.
16. Borgman, C.L. Social aspects of digital libraries. In DL'96: Proceedings of the 1st ACM International Conference on Digital Libraries, D-Lib Working Session, http://is.gseis.ucla.edu/research/dl/UCLA_DL_Report.html, 1996.
17. Bowman, C.M., Danzig, P.B., Hardy, D.R., Manber, U. and Schwartz, M.F. The Harvest information discovery and access system. *Computer Networks and ISDN Systems*, *28* (1-2). 119 - 125.
18. Card, S.K., G.G., R. and York, W. The WebBook and the WebForager: an information workspace for the World Wide Web. In Proc. of ACM Human Factors in Computing Systems Conference (CHI'96). 111-117, 1996.
19. Carey, M., Haas, L., Maganty, V. and J.Williams PESTO:An integrated query/browser for object databases. In Proc. VLDB, 1996.
20. Champeny, L., Borgman, C.L., Leazer, G.H., Gilliland-Swetland, A.J., Millwood, K.A., D'Avolio, L., Finley, J.R., Smart, L.J., Mautone, P.D., Mayer, R.E. and Johnson, R.A. Developing a digital learning environment: an evaluation of design and implementation processes. In Proc. JCDL 2004, Tucson, 37-46.
21. Choo, C.W., Detlor, B. and Turnbull, D. A Behavioral Model of Information Seeking on the Web -- Preliminary Results of a Study of How Managers and IT Specialists use the Web. In Proceedings of the 61st Annual Meeting of the ASIS, pp.290-302, 1998.
22. Choo, C.W., Detlor, B. and Turnbull, D. Information Seeking on the Web -- An Integrated Model of Browsing and Searching. In Proceedings of the 62nd Annual Meeting of the ASIS, pp. 3-16, 1999.
23. Croft, W.B. and Thompson, R.H. I3R: A New Approach to the Design of Document Retrieval Systems. *JASIS*, *38* (6). 389–404, 1987.
24. Cutting, D., Karger, D., Pedersen, J. and Tukey, J. Scatter/Gather: A clusterbased approach to browsing large document collections. In Proc. of the 15th Annual International ACM/SIGIR Conference. 318-329, 1992.
25. Daniel, J.R. and Lagoze, C. Extending the Warwick Framework: From Metadata Containers to Active Digital Objects. D-Lib Magazine 3(11), http://www.dlib.org/dlib/november97/daniel/11daniel.html, 1997.
26. Davis, J.R. and Lagoze, C. NCSTRL: Design and deployment of a globally distributed digital library. *JASIS*, *51* (3). 273-280.

27. DeLone, W.D. and McLean, E.R. The DeLone and McLean Model of Information Systems Success: A Ten-Year Update. *Journal of Management Information Systems*, *19* (4). 9 - 30, 2003.
28. DeLone, W.H. and McLean, E.R. Information systems success: The quest for the dependent variable. *Information Systems Research*, *3* (1). 60-95, 1992.
29. Doan, A., Domingos, P. and Halevy, A.Y. Reconciling Schemas of Disparate Data Sources: A Machine-Learning Approach. In Proc. SIGMOD 2001. 509-820.
30. Doll, W.J. and Torkzadeh, G. Developing a multidimensional measure of systems use in an organizational context. *Information & Management*, *33* (4). 171-185, 1998.
31. Egan, D., Remde, J., Landauer, T., Lochbaum, C. and Gomez, L. Behavioral Evaluation and Analysis of a Hypertext Browser. In Proc. CHI'89: 205-210.
32. Egan, D.E., Remde, J.R., Gomez, L.M., Landauer, T.K., Eberhardt, J. and Lochbaum, C.C. Formative design-evaluation of SuperBook. *ACM Trans. Inf. Syst*, *7* (1). 30-57, 1989.
33. Ellis, D. Modeling the information-seeking patterns of academic researchers: A grounded theory approach. The Library Quarterly, 63(4): 469-486, 1993.
34. Ellis, D. and Haugan, M. Modeling the information seeking patterns of engineers and research scientists in an industrial environment. Journal of Documentation, 53(4): 384-403, 1997.
35. Fensel, D., Hendler, J.A., Lieberman, H. and Wahlster, W. (eds.). *Spinning the Semantic Web*. MIT Press, 2003.
36. Finkelstein, s., Ussishkin, D. and Halpern, B. Monograph Series of the Institute of Archaeology, Tel Aviv University, 2000.
37. Fonseca, F., Egenhofer, M., Agouris, P. and Camara, G. Using ontologies for integrated geographic information systems. *Transactions in Geographic Information Systems*, *6* (3). 13-19, 2002.
38. Fox, E.A. Development of the CODER System: A Testbed for Artificial Intelligence Methods in Information Retrieval. IP&M, 1987, 23(4): 341-366.
39. Fox, E.A., France, R., Sahle, E., Daoud, A. and Cline, B. Development of a Modern OPAC: From REVTOLC to MARIAN. Proc. 16th Annual Intern'l ACM SIGIR Conf. on R & D in Information Retrieval, SIGIR '93, Pittsburgh, PA, June 27 - July 1, 1993, 248-259.
40. Fox, E.A., France, R.K., Gonçalves, M.A. and Suleman:, H. Building Interoperable Digital Library Services: MARIAN, Open Archives and NDLTD. In Proc. SIGIR 2001. 451-451.
41. Fox, E.A., Hix, D., Nowell, L.T., Brueni, D.J., Wake, W.C., Heath, L.S. and Rao, D. Users, User Interfaces, and Objects: Envision, a Digital Library. JASIS 44(8): 480-491 (1993).
42. Fox, E.A. and Marchionini, G. Toward a Worldwide Digital Library - Introduction. Commun. ACM 41(4): 28-32, 1998.
43. Fox, E.A., Neves, F.D., Yu, X., Shen, R., Kim, S. and Fan, W. Exploring the computing literature with visualization and stepping stones & pathways. *CACM*, *49* (4). 52-58, 2006.
44. Fuhr, N., Hansen, P., Mabe, M., Micsik, A. and Sølvberg, I. Digital libraries: A generic classification and evaluation scheme. Lecture Notes in Computer Science, 2163:187–199, 2001.
45. Fulker, D.W. Interviews with NSDL Grantees on Core Values and Service Perspectives. In Proc. JCDL2005, Denver, 170-171.
46. Furnas, G.W. and Rauch, S.J. Considerations for Information Environments and the NaviQue Workspace. In Proc. DL 98: 79-88.
47. Garcia-Molina, H., Papakonstantinou, Y., Quass, D., Rajaraman, A., Sagiv, Y., Ullman, J.D., Vassalos, V. and Widom, J. The TSIMMIS Approach to Mediation: Data Models and Languages, Journal of Intelligent Information Systems. *8* (2). 117-132, 1997.
48. Golovchinsky, G. Queries? Links? Is there a difference? In Proc. CHI'97: 407-417.
49. Gómez-Pérez, A. and Oscar Corcho Ontology Specification Languages for the Semantic Web. *IEEE Intelligent Systems*, *17* (1). 54-60, 2002.
50. Gonçalves, M.A. Streams, Structures, Spaces, Scenarios, and Societies (5S): A Formal Digital Library Framework and Its Applications, Ph.D. Dissertation, Virginia Tech, http://scholar.lib.vt.edu/theses/available/etd-12052004-135923, 2004.
51. Gonçalves, M.A. and Fox, E.A. Toward a Digital Library Theory: A Formal Digital Library Ontology. SIGIR, Mathematical Formal Methods workshop, 2004.
52. Goncalves, M.A., Fox, E.A., Krowne, A., Calado, P., Laender, A., Silva, A.S.d. and Ribeiro-Neto, B. The Effectiveness of Automatically Structured Queries in Digital Libraries. In Proc. JCDL2004, Tucson, 98-107.
53. Gonçalves, M.A., Fox, E.A., Watson, L.T. and Kipp, N.A. Streams, structures, spaces, scenarios, societies (5s): A formal model for digital libraries. ACM Trans. Inf. Syst. 22(2): 270-312 (2004).
54. Gonçalves, 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. In Proc. SPIRE 2002. 194-209.
55. Grün, C., Gerken, J., Jetter, H.-C., König, W. and Reiterer, H. MedioVis - A User-Centred Library Metadata Browser. In Proc. ECDL 2005, Vienna, 174-185.
56. Gruninger, M. and (editors), J.L. SPECIAL ISSUE: Ontology applications and design. CACM 45(2): 39-65, 2002.
57. Guarino, N. Formal Ontology and Information Systems. In Proc. FOIS 1998, Italy, June, 3-15.
58. Hartson, H.R., Shivakumar, P. and Pérez-Quiñones, M.A. Usability inspection of digital libraries: a case study. Int. J. on Digital Libraries 4(2): 108-123 (2004).
59. Hasselbring, W. Information System Integration: Introduction. Commun. ACM 43(6): 32-38, 2000.

60. Hassenzahl, M., Platz, A., Burmester, M. and Lehner, K. Hedonic and ergonomic quality aspects determine a software's appeal. CHI 2000: 201-208.
61. Hearst, M.A. and Pedersen, J.O. Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results. In Proc. of the 19th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'96). 76-84, 1996.
62. Hill, L.L., Carver, L., Larsgaard, M., Dolin, R., Smith, T.R., Frew, J. and Rae, M.-A. Alexandria digital library: user evaluation studies and system design. JASIS 51(3): 246-259 (2000).
63. http://www.altova.com/products_mapforce.html (Altova MapForce).
64. http://www.altova.com/products_semanticworks.html (Altova SemanticWorks).
65. http://www.microsoft.com/biztalk/ (Microsoft BizTalk).
66. http://www.ndltd.org/standards/metadata/etd-ms-v1.01.html.
67. http://www.w3.org/RDF/.
68. James, W. What is an Emotion? Mind, 9 , p.188-205, 1884, http://psychclassics.yorku.ca/James/emotion.htm.
69. Jordan, P.W. Patrick W. Jordan: Designing Pleasurable Products: Timeless Laws and Current Trends. IWIPS 2005: 197-200.
70. Kahn, R. and Wilensky, R. A Framework for Distributed Digital Object Services, http://www.cnri.reston.va.us/k-w.html, 1995.
71. Kampanya, N., Rao Shen, S.K., North, C. and Fox, E.A. Citiviz: A Visual User Interface to the CITIDEL System. In Proc. ECDL2004, Bath, UK, 122-133, 2004.
72. Kelapure, R.D. Scenario-based Design of Digital Library Service, Masters Thesis, Virginia Tech, http://scholar.lib.vt.edu/theses/available/etd-06182003-055012/, 2003.
73. Kengeri, R., Seals, C.D., Harley, H.D., Reddy, H.P. and Fox, E.A. Usability Study of Digital Libraries: ACM, IEEE-CS, NCSTRL, NDLTD. Int. J. on Digital Libraries 2(2-3): 157-169 (1999).
74. Kohonen, T. Exploration of very large databases by self-organizing maps. In Proc. of the IEEE International Conference on Neural Networks. pp.1-6, 1997.
75. Kuhlthau, C.C. Learning in digital libraries: an information search process approach. Library Trends 45(4): 708-724, 1997.
76. Kuhlthau, C.C. and Tama, S.L. Information search process of lawyers, a call for 'just for me' information services. Journal of Documentation, 57(1): 25-43, 2001.
77. Kummamuru, K., Lotlikar, R., Roy, S., Singal, K. and Krishnapuram, R. A hierarchical monothetic document clustering algorithm for summarization and browsing search results. In Proc. of the 13th international conference on World Wide Web. 658-665, 2004.
78. Kurosu, M. and Kashimura, K. Apparent usability vs. inherent usability: experimental analysis on the determinants of the apparent usability. CHI 95 Conference Companion 1995: 292-293.
79. Lagoze, C., Arms, W.Y., Gan, S., Hillmann, D., Ingram, C., Krafft, D.B., Marisa, R.J., Phipps, J., Saylor, J., Terrizzi, C., Hoehn, W., Millman, D., Allan, J., Guzman-Lara, S. and Kalt:, T. Core services in the architecture of the National Science Digital Library (NSDL). In Proc. JCDL 2002, Portland, Oregon, 201-209.
80. Lagoze, C., Lynch, C.A. and Daniel, J.R. The Warwick Framework: A Container Architecture for Aggregating Sets of Metadata. Cornell University Computer Science, Technical Report TR96-1593, June 1996.
81. Lagoze, C. and Sompel, H.V. The Open Archives Initiative: building a low-barrier interoperability framework. In Proc. JCDL 2001, Roanoke, VA, 54-62.
82. Lamping, J. and Rao, R. Laying Out and Visualizing Large Trees Using a Hyperbolic Space. In Proc. of the ACM Symposium on User Interface Software and Technology. 13-14, 1994.
83. Larsen, R.L. and Wactlar, H.D. Knowledge Lost in Information: Report of the NSF Workshop on Research Directions for Digital Libraries, June 15-17, 2003, Chatham, MA, National Science Foundation Award No. IIS-0331314. http://www.sis.pitt.edu/~dlwkshop/.
84. Lassila, K.S. and Brancheau, J.C. Adoption and utilization of commercial software packages: Exploring utilization equilibria, transitions, triggers, and tracks. *Journal of Management Information Systems*, *16* (2). 63-90, 1999.
85. Lawrie, D.J. and Croft, W.B. Generating Hierarchical Summaries for Web Searches. In Proc. SIGIR 2003. 457-458, 2003.
86. Li, W.-S. and Clifton, C. SEMINT: a tool for identifying attribute correspondences in heterogeneous databases using neural networks. *Data & Knowledge Engineering*, *33* (1). 49 - 84.
87. Ling, D.N.C., Teo, L.S. and Byrne, J.G. Modelling interface aesthetics. Inf. Sci. 152: 25-46 (2003).
88. Lynch, C.A. The Z39.50 Information Retrieval Standard. D-Lib Magazine, 3(4), http://www.dlib.org/dlib/april97/04lynch.html, 1997.
89. Manber, U., Smith, M. and Gopal, B. Webglimpse: Combining browsing and searching. In Proc. Usenix Technical Conference, 1997: 195-206.
90. Marchionini, G. Evaluating Digital Libraries: A Longitudinal and Multifaceted View, Library Trends, 49(2): 304-333, 2000.
91. Marchionini, G. Information seeking in electronic environments. Cambridge, England: Cambridge University Press. 1995.
92. Marshall, B., Zhang, Y., Chen, H., Lally, A.M., Shen, R., Fox, E.A. and Cassel, L.N. Convergence of Knowledge Management and E-Learning: The GetSmart Experience. In Proc. JCDL2003, Houston, 135-146, 2003.
93. Meho, L.I. and Tibbo, H.R. Modeling the information-seeking behavior of social scientists: Ellis's study revisited. JASIST 54(6): 570-587, 2003.

94. Mena, E., Illarramendi, A., Kashyap, V. and Sheth, A.P. OBSERVER: An Approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies. *International Journal on Distributed and Parallel Databases (DAPD)*, *8* (2). 223-272, 2000.
95. Mena, E., Kashyap, V., Illarramendi, A. and Sheth, A.P. Managing Multiple Information Sources through Ontologies: Relationship between Vocabulary Heterogeneity and Loss of Information. In Proc. of the workshop on Knowledge Representation meets Databases in conjunction with European Conference on Artificial Intelligence (ECAI'96), Budapest, Hungary, 50-52, 1996.
96. Miller, R.J., Tsatalos, O.G. and Williams, J.H. DataWeb: Customizable Database Publishing for the Web. *Multimedia, IEEE*, *4* (4). 14-21, 1997.
97. Morgan, E.L. An Introduction to the Search/Retrieve URL Service (SRU), Ariadne Magazine, Issue 40, http://www.ariadne.ac.uk/issue40/morgan/, July, 2004.
98. Mukhopadhyay, P. and Papakonstantinou, Y. Mixing Querying and Navigation in MIX. In Proc. ICDE 2002: 245-254.
99. Munroe, K.D. and Papakonstantinou, Y. BBQ: A Visual Interface for Integrated Browsing and Querying of XML. In Proc. of Visual Database Systems 2000: 277-296.
100. Naumann, F. and Häussler, M. Declarative Data Merging with Conflict Resolution. In Proceedings of the 2002 International Conference on Information Quality, 2002, 212-224.
101. Nelson, M.L., Maly, K. and Zubair, M. Interoperable Heterogeneous Digital Libraries, Technical Report, TR_98_07, http://www.cs.odu.edu/~techrep/techreports/TR_98_07.ps.Z, Old Dominion University Computer Science Department, 1998.
102. Nevill-Manning, C.G., Witten, I.H. and Paynter, G. Lexically-generated subject hierarchies for browsing large collections. *International Journal on Digital Libraries*, *2* (1432-5012). 111-123.
103. Noy, N.F., Sintek, M., Decker, S., Crubezy, M., Fergerson, R.W. and Musen, M.A. Creating Semantic Web Contents with Protege-2000. *IEEE Intelligent Systems*, *16* (2). 60-71, 2002.
104. Olston, C. and Chi, E.H. ScentTrails: Integrating browsing and searching on the Web. *ACM Trans. Comput.-Hum. Interact*, *10* (3). 177-197, 2003.
105. Paepcke, A., Baldonado, M., Chang, C.-C.K., Cousins, S. and Garcia-Molina, H. Building the InfoBus: A Review of Technical Choices in the Stanford Digital Library Project. *IEEE Computer*, *32* (2). 80-87.
106. Paepcke, A., Chang, K.C.-C., Garcia-Molina, H. and Winograd, T. Interoperability for Digital Libraries Worldwide. Commun. ACM 41(4): 33-43, 1998.
107. Paepcke, A., Chang, K.C.-C., Garcia-Molina, H. and Winograd, T. Search Middleware and the Simple Digital Library Interoperability Protocol. D-Lib Magazine 6(3), http://www.dlib.org/dlib/march00/paepcke/03paepcke.html, 2000.
108. Park, J. and Ram, S. Information systems interoperability: What lies beneath? ACM Trans. Inf. Syst. 22(4): 595-632 (2004).
109. Patel, M., Koch, T., Doerr, M. and Tsinaraki, C. Semantic Interoperability in Digital Library Systems, report of DELOS2 Network of Excellence in Digital Libraries, 2005.
110. Patel-Schneider, P. and Laboratories, B. The Yin/Yang Web: XML Syntax and RDF Semantics. In Proc. WWW 2002. 443-453, 2002.
111. Patel-Schneider, P.F. and Siméon, J. Building the Semantic Web on XML. In Proc. of the first International Semantic Web Conference (ISWC 2002), Sardinia, Italia. 147-161, 2002.
112. Payette, S., Blanchi, C., Lagoze, C. and Overly, E.A. for Digital Objects and Repositories: The Cornell/CNRI Experiments. D-Lib Magazine 5(5), http://www.dlib.org/dlib/may99/payette/05payette.html, 1999.
113. Payette, S., Blanchi, C., Lagoze, C. and Overly, E.A. Interoperability for Digital Objects and Repositories: The Cornell/CNRI Experiments. D-Lib Magazine 5(5), http://www.dlib.org/dlib/may99/payette/05payette.html, 1999.
114. Perugini, S., McDevitt, K., Richardson, R., Perez-Quinones, M., Shen, R., Ramakrishnan, N., Williams, C. and Fox, E.A. Enhancing Usability in CITIDEL: Multimodal, Multilingual, and Interactive Visualization Interfaces. In Proc. JCDL2004, Tucson, 315-324, 2004.
115. Peuquet, D., Smith, B. and Brogaard, B. The Ontology of Fields. Report of a Specialist Meeting Held under the Auspices of the Varenius Project. http://www.ncgia.ucsb.edu/Publications/Varenius_Reports/Ontology_of_Fields.pdf, 1998.
116. Plaisant, C., Grosjean, J. and Bederson, B.B. SpaceTree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation. In Proc. INFOVIS 2002. IEEE Symposium on Information Visualization. 57 -64, 2002.
117. Raghavan, A., Rangarajan, D., Shen, R., Goncalves, M.A., Vemuri, N.S., Fan, W. and Fox, E.A. Schema Mapper: A Visualization Tool for DL Integration. In Proceedings JCDL2005, Denver, p. 414, 2005.
118. Raghavan, A., Vemuri, N.S., Shen, R., Gonçalves, M.A., Fan, W. and Fox, E.A. Incremental, Semi-automatic, Mapping-Based Integration of Heterogeneous Collections into Archaeological Digital Libraries: Megiddo Case Study. In Proc. ECDL2005, Vienna, 139-150, 2005.
119. Rahm, E. and Bernstein, P.A. A Survey of Approaches to Automatic Schema Matching. *VLDB Journal*, *10* (4). 334-350.
120. Ram, S., Park, J. and Lee, D. Digital Libraries for the Next Millennium: Challenges and Research Directions. Information Systems Frontiers 1(1): 75-94 (1999).
121. Ram, S. and Ramesh, V. Information Sharing among Multiple Heterogeneous Data Sources Distributed across the Internet. *HICSS 1998*, *4*. 504.
122. Ravindranathan, U. Prototyping Digital Libraries Handling Heterogeneous Data Sources - An ETANA-DL Case Study, Masters Thesis, Virginia Tech, http://scholar.lib.vt.edu/theses/available/etd-04262004-153555/, 2004.

123. Ravindranathan, U., Shen, R., Gonçalves, M.A., Weiguo Fan, E.A.F. and Flanagan, J.W. ETANA-DL: a digital library for integrated handling of heterogeneous archaeological data. In Proc. JCDL 2004, Tucson, 76-77, 2004.
124. Ravindranathan, U., Shen, R., Gonçalves, M.A., Weiguo Fan, E.A.F. and Flanagan, J.W. Prototyping Digital Libraries Handling Heterogeneous Data Sources - The ETANA-DL Case Study. In Proc. ECDL2004, Bath, UK, 186-197, 2004.
125. Rennison, E. Galaxy of news: An approach to visualizing and understanding expansive news landscapes. In Proc. of UIST'94, ACM Symposium on User Interface Software and Technology. 3-12, 1994.
126. Robertson, G.G., Card, S.K. and Mackinlay, J.D. Information Visualization Using 3D Interactive Animation. *Communications of the ACM*, *36(4)*. 57-71.
127. Ryan, D.N. Managing Complexity: Archaeological Information Systems Past, Present and Future, http://www.cs.kent.ac.uk/people/staff/nsr/arch/baas.html.
128. S. Berretti, A.D.B. and Vicario, E. Spatial arrangement of color in retrieval by visual similarity, Pattern Recognition, 35(8): 1661-1674.
129. Salton, G. Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer, Addison-Wesley, 1989.
130. Salton, G. and McGill, M.J. Introduction to Modern Information Retrieval. McGraw Hill Book Co., New York, 1983.
131. Sanderson, M. and Croft, W.B. Deriving concept hierarchies from text. In Proc. of the 22nd Annual International ACM/SIGIR Conference. 206-213, 1999.
132. Saracevic, T. Digital library evaluation: Toward evolution of concepts. Library Trends, 49(2): 350–369, 2000.
133. Savaresi, S.M., Boley, D., Bittanti, S. and Gazzaniga, G. Cluster Selection in Divisive Clustering Algorithms. In Proc. of the Second SIAM International Conference on Data Mining,. 299-314, 2002.
134. Schloen, J.D. Archaeological Data Models and Web Publication Using XML. Computers and the Humanities 35(2): 123-152 (2001).
135. Sciascio, E.D., Donini, F.M. and Mongiello, M. Spatial layout representation for query-by-sketch content-based image retrieval, Pattern Recognition Letters, 23(13): 1599-1612.
136. Seddon, P.B. A respecification and extension of the DeLone and McLean model of IS success. *Information Systems Research*, *8* (3). 240-253, 1997.
137. Shen, R., Gonçalves, M.A., Fan, W. and Fox, E.A. Requirements Gathering and Modeling of Domain-Specific Digital Libraries with the 5S Framework: An Archaeological Case Study with ETANA. In Proc. ECDL2005, Vienna, Sept. 18-23.
138. Shneiderman, B., Feldman, D., Rose, A. and Grau, X.F. Visualizing digital library search results with categorical and hierarchical axes. In Proc. ACM DL 2000: 57-66.
139. Staples, T., Wayland, R. and Payette, S. Fedora Project: An Open- source Digital Object Repository System. D-Lib Magazine 9(4), http://www.dlib.org/dlib/april03/staples/04staples.html, 2003.
140. Suleman, H. Open Digital Libraries, Ph.D. Dissertataion, Department of Computer Science, Virginia Polytechnic Institute and State University, http://scholar.lib.vt.edu/theses/available/etd-11222002-155624/unrestricted/odl.pdf, 2002.
141. Suleman, H. and Fox, E.A. The Open Archives Initiative: Realizing Simple and Effective Digital Library Interoperability. In special issue on "Libraries and Electronic Resources: New Partnerships, New Practices, New Perspectives" of J. Library Automation. *35* (No. 1/2). 125-145, 2002.
142. Sure, Y., Erdmann, M., Angele, J., Staab, S., Studer, R. and Wenke, D. OntoEdit: Collaborative Ontology Development for the Semantic Web. In Proc. of the first International Semantic Web Conference (ISWC 2002), Sardinia, Italia, 348-363, 2002.
143. Thong, J.Y.L., Hong, W. and Tam, K.Y. What leads to acceptance of digital libraries? Commun. ACM 47(11): 78-83 (2004).
144. Uschold, M. and Gruninger, M. Ontologies: principles, methods, and applications. Knowledge Engineering Review. *11* (2). 93-155, 1996.
145. Venkatesh, V., Morris, M., Davis, G. and Davis, F. User acceptance of information technology: Toward a unified view. *MIS Quarterly*, *27* (3). 425-478, 2003.
146. West, D., Finnegan, M., Lane, R.W. and Kysar, D.A. Analysis of Faunal Remains Recovered from Tell Nimrin, Dead Sea Valley, Jordan, final report, 1996.
147. Wiederhold, G. Mediators in the Architecture of Future Information Systems. IEEE Computer. *25* (3). 38-49, 1992.
148. Williams, M.D. What makes RABBIT run? *International Journal of Man-Machine Studies*, *21* (4). 333-352, 1984.
149. Wilson, T.D. Models in information behaviour research. Journal of Documentation, 55(3): 249-270, 1999.
150. Wise, J.A., Thomas, J.J., Pennock, K., Lantrip, D., Pottier, M. and Schur, A. Visualizing the non-visual: Spatial analysis and interaction with information from text documents. In Proc. of the Information Visualization Symposium 1995. 51-58.
151. Wittenburg, K. and Sigman, E. Integration of browsing, searching, and filtering in an applet for Web information access. In Proc. CHI 1997: 293-294.
152. Young, D. and Benamati, J. Differences in public Web sites: The current state of large U.S. firms. *Journal of Electronic Commerce Research*, *1* (3). 94-105, 2000.
153. Zajonc, R.B. Attitudinal Effects of Mere Exposure. Journal of Personality and Social Psychology, 9(2): 1-27, 1968.
154. Zamir, O. and Etzioni, O. Grouper: A Dynamic Clustering Interface to Web Search Results, Computer Networks 31(11-16): 1361-1374, http://www8.org/w8-papers/3a-search-query/dynamic/dynamic.html, 1999.

155. Zhang, J. and Marchionini, G. Evaluation and evolution of a browse and search interface: relation browser. In Proc. of the 2005 national conference on digital government research, 179-188.
156. Zhu, Q. 5SGraph: A Modeling Tool for Digital Libraries, Masters Thesis, Virginia Tech, http://scholar.lib.vt.edu/theses/available/etd-11272002-210531/, 2002.
157. Zhu, Q., Gonçalves, M.A., Shen, R., Cassel, L. and Fox:, E.A. Visual Semantic Modeling of Digital Libraries. In Proc. ECDL2003, Trondheim, Norway, 325-337, 2003.
158. Zubair, M., Maly, K., Ameerally, I. and Nelson, M.L. Dynamic Construction of Federated Digital Libraries. Poster in WWW9 Conference, http://www9.org/final-posters/poster17.html, 2000.