

# Digital Libraries: A 5S Approach

Edward A. Fox, Editor

Virginia Tech, Dept. of Computer Science, Blacksburg, VA 24061 USA

## Chapter Authors:

Monika Akbar, Pranav Angara, Yinlin Chen, Lois M. Delcambre, Noha Elsherbiny, Eric Fouh, Marcos André Gonçalves, Nádia P. Kozievitch, Spencer Lee, Jonathan Leidig, Lin Tzy Li, Mohamed Magdy Gharib Farag, Uma Murthy, Sung Hee Park, Rao Shen, Venkat Srinivasan, Ricardo da Silva Torres, and Seungwon Yang

Draft 12/11/11

© 2011 Edward A. Fox  
All rights reserved.

Do not copy for others, disseminate, or use other than for personal purposes.  
This version is for the co-authors; those associated with CS6604, Fall 2011; reviewers asked to comment on the work; and/or Virginia Tech's Digital Library Research Laboratory. All others are asked to destroy any copies they receive.

## ABSTRACT

In 1991, a group of researchers chose the term *digital libraries* to describe an emerging field of research, development, and practice. Over the last 20 years, Virginia Tech has had funded research in this area, largely through its Digital Library Research Laboratory. This volume reports our key findings and current research investigations.

Much of the early work in the digital library field struck a balance between addressing real-world needs, integrating methods from related areas, and advancing an ever-expanding research agenda. Our work has fit in with these trends, but simultaneously has been driven by a desire to provide a firm conceptual and formal basis for the field. Our aim has been to move from engineering to science. We claim that our *5S* (Societies, Scenarios, Spaces, Structures, Streams) framework, discussed in publications dating back to at least 1998, provides a suitable basis, as can be seen in this volume.

While the *5S* framework may be used to describe many types of information systems, and is likely to have even broader utility and appeal, we focus here on digital libraries. Our view of digital libraries is broad, so further generalization should be straightforward. This is proved in part by the recent expansion of interest related to each of the five Ss, e.g., Social networks, Scenario-based design, geoSpatial databases, Structure-based approaches (e.g., metadata, ontologies, XML), and data Stream management systems.

*5S* also has allowed us to cover the three parts of this work: Basic Concepts, Advanced Concepts, and Applications. It has made it possible to describe key issues, and to develop well grounded methods and systems. Thus, we have addressed Exploration, Evaluation, Integration, and Security. We have connected with related fields, including hypertext/hypermedia, information storage and retrieval, knowledge management, machine learning, multimedia, personal information management, and Web 2.0. Applications have included managing not only publications, but also archaeological information, educational resources, fish images, scientific datasets, and scientific experiments/simulations.

Underlying this volume are five completed dissertations (Gonçalves, Kozievitch, Murthy, Shen, Torres), nine dissertations underway, and many masters theses. There are hundreds of related publications, presentations, tutorials, and reports. Yet this is more than a lab report or typical edited volume. It is organized for use as a textbook, suitable for computer science, information science, and library science (e.g., LIS) courses, as well as for use by researchers, developers, and practitioners. While each chapter has a section on formalization, generally those can be skipped by those who prefer to focus on the many case studies. These reflect our experience with a long string of prototype or production systems developed in the lab, such as CITIDEL, CODER, CTRnet, Ensemble, ETANA, ETD-db, MARIAN, and Open Digital Libraries.

Given this rich content, we trust that any interested in digital libraries, or in related systems, will find this volume to be intellectually satisfying, illuminating, and helpful. We

hope it will help move digital libraries forward into a science as well as a practice. We hope it will help address the needs of the next generation of digital librarians.

## KEYWORDS

5S framework, annotation, classification, compound objects, content based image retrieval (CBIR), digital libraries (DLs), digital objects, documents, e-science, education, electronic theses and dissertations (ETDs), evaluation, exploration, formalization, geospatial information, integration, LIS curriculum, metadata, ontologies, personalization, security, simulation, social networks

## 0.1 DEDICATION

This book is dedicated to all those who have worked in, or collaborated with, Virginia Tech's Digital Library Research Laboratory.

## 0.2 ACKNOWLEDGMENTS

As editor, my belief is that our greatest thanks go to our families. Accordingly, I thank my wife, Carol, and our sons, Jeffrey, Gregory, Michael, and Paul, along with their growing families, as well as my parents and many other relatives. Similarly, on behalf of the chapter authors, I thank all of their families.

Teachers and mentors deserve a special note of thanks. My interest in research was stimulated and guided by JCR Licklider, my undergraduate advisor, author of *Libraries of the Future*<sup>1</sup>, who, when at ARPA, funded the start of the Internet. Michael Kessler, who introduced the concept of bibliographic coupling, was my BS thesis advisor; he also directed MIT's Project TIP (technical information project). Gerard Salton was my graduate advisor (1978–1983); he is sometimes called the 'Father of Information Retrieval'.

Likewise, we thank our many students, friends, collaborators, co-authors, and colleagues. In particular, we thank former students who have collaborated, including: Pavel Calado, Yuxin Chen, Fernando Das Neves, Shahrooz Feizabadi, Robert France, Nithiwat Kampanya, Rohit Kelapure, S.H. Kim, Neill Kipp, Aaron Krowne, Bing Liu, Ming Luo, Paul Mather, Unni Ravindranathan, Ryan Richardson, Ohm Sornil, Hussein Suleman, Wensi Xi, Baoping Zhang, and Qinwei Zhu.

Likewise, we thank faculty and staff, at a variety of universities and other institutions, who have collaborated, including: Lillian Cassel, Vinod Chachra, Hsinchun Chen, Debra Dudley, Roger Ehrich, Joanne Eustis, Weiguo Fan, James Flanagan, James French, Richard Furuta, Dan Garcia, C. Lee Giles, Martin Halbert, Eberhard Hilf, Gregory Hislop, John Impagliazzo, Filip Jagodzinski, Douglas Knight, Deborah Knox, Alberto Laender, Carl Lagoze, Susan Marion, Gail McMillan, Claudia Medeiros, Manuel Perez, Naren Ramakrishnan, Frank Shipman, and Layne Watson.

Clearly, however, with regard to this volume, my special thanks go to my co-authors. Each has played a key role in the unfolding of the theory, practice, systems, and usability of what is described herein. Special thanks go to Uma Murthy for helping with the bibliography and to Monika Akbar for assistance with technical aspects of book production. Regarding 5S, Marcos André Gonçalves helped launch our formal framework, and has continued to exercise intellectual leadership in this regard.

At Virginia Tech, there are many in the Department of Computer Science and in Information Systems that have assisted, providing very nice facilities and a creative and supportive environment. The College of Engineering, and before that, of Arts and Sciences, provided an administrative home and intellectual context.

In addition, we acknowledge the support of the many sponsors of the research described in this volume. Our fingerprint work was supported by Award No. 2009-DN-BX-K229 from the National Institute of Justice, Office of Justice Programs, U.S. Department

<sup>1</sup>In this 1965 work, Licklider called for an integrative theory to support future automated libraries, one of the inspirations for this book.

of Justice. The opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect those of the Department of Justice.

Some of this material is based upon work supported by the National Science Foundation (NSF) under Grant Nos. CCF-0722259, DUE-0121679, DUE-0121741, DUE-0136690, DUE-0333601, DUE-0840719, IIS-9986089, IIS-0080748, IIS-0086227, IIS-0325579, IIS-0910183, IIS-0916733, ITR-0325579, OCI-0904844, SES-0729441, .... Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

This work has been partially supported by NIH MIDAS project 2U01GM070694-7, DTRA CNIMS Grant HDTRA1-07-C-0113, and R&D Grant HDTRA1-0901-0017.

We thank corporate and institutional sponsors, including Adobe, AOL, CNI, Google, IBM, Microsoft, NASA, NCR, OCLC, SOLINET, SUN, SURA, UNESCO, US Dept. Ed. (FIPSE), VTLS, .... A variety of institutions have supported tutorials or courses, including AUGM, CETREDE, CLEI, IFLA-LAC, and UFC.

Visitors and collaborators from Brazil, including from FUA, UFMG, and UNICAMP, have been supported by CAPES (4479-09-2), FAPESP, and CNPq. Our collaboration in Mexico had support from CONACyT, while that in Germany was supported by DFG. Students in our VT-MENA program in Egypt have been supported through that program.

### 0.3 PREFACE

Michael Lesk, Michael McGill, and I met at a workshop connected with ACM SIGIR 1991 in Chicago, hoping that information retrieval would lead to broader practical impact. We led an effort to launch the field of digital libraries, that rapidly connected with parallel interests and activities of many other researchers. This also resonated with leaders at all levels, to help advance education, preserve and share cultural heritage, and move society forward into a new approach to creating, sharing, disseminating, discovering, and (re)using knowledge.

Now, 20 years later, the digital library field today is manifest in many ways. There is a LinkedIn group of over 3000, with many joining daily. Almost all publishers and scholarly societies now have their own digital library, or are part of one with suitable partners. Many nations, or national consortia like the European Union, or agencies thereof, run integrated digital libraries. Most universities, and in some cases multiple parts of those organizations, have an institutional repository. Content management systems are widely used in education, and in a variety of other contexts; in addition there are e-portfolio systems, e-print systems, and a variety of personal information management systems.

Thinking expansively, one might connect popular systems used by billions of people with the field of digital libraries. Consider, for example, offerings/services by companies such as Facebook, Flickr, Google, Microsoft, and Yahoo! There also are many specialized systems, like Drupal, DSpace, E-prints, Fedora Commons, Greenstone, and VITAL.

This volume approaches digital libraries through fifteen chapters, spread across three parts. Part 1 covers Basic Concepts, while Part 2 covers Advanced Concepts. To help show how these concepts are applied, Part 3 covers Applications. In Part 1, Chapter 1 provides a broad Introduction, including covering the formal definitions leading to a minimal digital library. With regard to case studies, it briefly introduces many of the systems and applications discussed in-depth later on. Chapter 2 describes Exploration, which includes key services for discovery, search, browsing, and visualization. It uses the ETANA-DL effort to provide context, since browsing is important for archaeological information. Chapter 3 discusses Evaluation, covering the Information Life Cycle, metrics, and software to help evaluate digital libraries. It uses electronic theses and dissertations to provide context, since addressing quality in highly distributed digital libraries is particularly challenging.

Part 2 shifts to more advanced concepts. Chapter 4 explains Complex Objects. While many digital libraries focus on digital objects and/or metadata objects, with support for complex objects they can easily handle aggregation and packaging. Fingerprint matching provides a useful context, since there are complex inter-relationships among crime scenes, latent fingerprints, individuals, hands, fingers, fingerprints, and images. Chapter 5 addresses a key challenge, Integration. It is grounded in years of work to develop ETANA-DL to integrate information from a variety of archaeological digs and projects. Coverage includes schema mapping as well as the 5S suite of tools to aid in such integration. Chapter 6

covers Subdocuments. This builds upon work on superimposed information, closely related to hypertext, hypermedia, and annotation. Multiple case studies cover prior work with fish images, as well as planned work with Flickr. Chapter 7 addresses a key area of knowledge management, also integral to the Semantic Web, namely Ontologies. It uses our work to develop a Crisis, Tragedy, and Recovery Network (CTRnet) as a context; that is quite broad, and so leading to many interesting ontology development problems. Chapter 8 covers a core area of information retrieval and machine learning, as well as Library and Information Science (LIS), namely Classification. Its context is ETDs, since many of these works have no categories that can be found in their catalog or metadata records, and since none are categorized at the level of chapters.

Part 3 shifts to a representative set of important digital library Applications. Chapter 9 moves into the multimedia field, focusing on Content-based Image Retrieval (CBIR). It makes use of the previously discussed work on fish images and on CTRnet, for context. Chapter 10 addresses very popular current issues, both on the Societies side. On one extreme, it covers Social Networks, while on the other end of the spectrum, it focuses on Personalization. By way of context, it discusses systems for collecting, sharing, and providing access to educational resources, namely the AlgoViz and Ensemble systems. That leads nicely to Chapter 11, on Education, using the same systems as context. This is important since there has been considerable investment in digital libraries to help in education, all based on the fact that devising high quality educational resources is expensive, making sharing and reuse highly beneficial. Chapter 12 covers Simulation and Scientific Digital Libraries. Simulation aids many disciplines to test models and predictions on computers, addressing questions not feasible through other approaches to experimentation. More broadly, in keeping with progress toward e-science, where data sets and shared information supports much broader theories and investigations, this chapter covers storing and archiving, as well as access and visualization, dealing not only with metadata, but also with specifications of experiments, experimental results, and derivative versions: summaries, findings, reports, and publications. It uses both the SimDL and CINET projects as context. Chapter 13 approaches Geospatial Information, now readily available in cell phones, cameras, and GPS systems. It connects that with metadata, images, and maps. It uses the CTRnet project as context. Chapter 14 focuses on Security. While many digital libraries support open access, it has been clear since the early 1990s that industrial acceptance of digital library systems and technologies depends on their being trusted, requiring an integrated approach to security. Chapter 15 introduces text extraction, especially in the context of ETDs, where the high level structure should be identified, and where the valuable and voluminous sets of references can be isolated and shifted to canonical representations.

Concluding the work are the appendices, with Mathematical Preliminaries (that supplement the definitions in Chapter 1) and a Glossary. Finally, there is an extensive Bibliography and a helpful Index.

### **0.3. PREFACE ix**

How can computer scientists connect with all this? Though some of the early curricular guidelines advocated coverage of information, and current guidelines refer to the area of Information Management, generally courses in this area have focused instead either on data or knowledge. Over time, some programs began teaching about multimedia. Fortunately, Virginia Tech has had graduate courses on information retrieval since the early 1970s. More recently, programs teach courses with titles including keywords like ‘Web’ or ‘search’. Perhaps parts of this volume will provide a way for computing programs to address all areas of Information Management, building on a firm, formal, integrated approach. Further, computing professionals should feel comfortable with particular Ss, especially Structures (as in data structures) and Spaces (as in vector spaces), and to lesser extents Streams (related to multimedia) and Scenarios (related to human-computer interaction). Today, especially, there is growing interest in Societies (as in social networks).

How can information scientists connect with all this? Clearly, they are at home with ‘information’ as a key construct. Streams (e.g., sequences of characters, or bitstreams) provide a first basis for all types of information. Coupled with Structures, they lead to all types of structured streams, as in documents and multimedia. Spaces may be less clear, but GIS systems are becoming ubiquitous, connecting with GPS, cell phone, Twitter, and other technologies. Scenarios, especially in the form of Services, are at the heart of most information systems. Societies, including users, groups, organizations, and a wide variety of social networks, are central, especially with human-centered design. Thus, information science can easily connect with 5S, and digital libraries are among the most important types of information systems. Accordingly, this volume may be fit nicely into capstone courses in information science or information systems.

How can library scientists connect with all this? One might argue that many of the librarians of the future must be trained as digital librarians. Thus, this work should fit nicely into library science programs. While it could fit into theory or capstone courses, it also might serve well in introductory courses, if the more formal parts are skipped. Alternatively, the first part of the book might work well early in a library school program, the second part could fit midway in the program, and the last part might be covered in specialized courses that connect with an individual chapter.

How can researchers connect with all this? We hope that those interested in formal approaches will help us expand the coverage of concepts reported herein. A wonderful goal would be to have an elegant formal basis, and useful framework, for all types of information systems. We also hope that the theses and dissertations related to this volume, all online (thanks to Virginia Tech’s ETD initiative), will provide an even more in-depth coverage of the key topics covered herein. We hope you can build on this foundation to aid in your own research, as you advance the field further.

How can developers connect with all this? We hope that concepts, ideas, methods, techniques, systems, and approaches described herein will guide you to develop, implement,

x

and deploy even better digital libraries. The result should be less time ‘reinventing the wheel’, and perhaps the emergence of a vibrant software and services industry as more and more digital libraries emerge. Further, if there is agreement on key concepts, then there should be improvements in: interoperability, integration, and understanding. We hope you thus can leverage this work to advance practices and provide better systems and services.

Even if you, the reader, do not fit clearly into the groups discussed above, we hope you nevertheless will find this volume interesting. Please share with us and others what ways you found this work to be useful and helpful!

— Edward A. Fox, Editor, Blacksburg, Virginia, September 2011

# Contents

0.1	Dedication.....	iv
0.2	Acknowledgments .....	v
0.3	Preface .....	vii
	<b>List of Tables .....</b>	<b>xxii</b>
	<b>List of Figures.....</b>	<b>xxiv</b>
<b>1</b>	<b>Introduction.....</b>	<b>1</b>
1.1	Context .....	1
1.2	Background .....	4
1.2.1	Definitions .....	4
1.2.2	Perspectives.....	5
1.3	Motivation .....	15
1.4	Digital Library Curriculum.....	22
1.5	High Level Constructs.....	27
1.6	Digital Library Systems .....	28
1.7	5S Intuition .....	30
1.7.1	Streams.....	30
1.7.2	Structures .....	31
1.7.3	Spaces .....	31
1.7.4	Scenarios .....	32
1.7.5	Societies .....	33
1.8	Formalization of Ss.....	34
1.8.1	5S Formalisms .....	34
1.9	Formalization of Minimal Digital Library .....	38
1.10	Digital Library Taxonomy.....	49

xii CONTENTS

1.11	Summary .....	52
1.12	Exercises and Projects .....	53
<b>2</b>	<b>Exploration.....</b>	<b>54</b>
2.1	Introduction .....	54
2.2	Related Work .....	56
2.3	Exploring Service Formalization .....	57
2.4	Case Study: Exploring Services in ETANA-DL .....	75
2.4.1	Multi-dimensional browsing .....	76
2.4.2	Browsing and searching integration.....	77
2.4.3	Browsing, searching, and visualization integration .....	79
2.4.4	ETANA-DL exploring services formative evaluation .....	83
2.5	Summary .....	86
2.6	Exercises and Projects .....	87
<b>3</b>	<b>Evaluation.....</b>	<b>88</b>
3.1	Introduction .....	88
3.2	Related Work .....	88
3.3	Background and Context .....	88
3.4	Formalization .....	89
3.5	Digital Objects.....	92
3.5.1	Accessibility .....	92
3.5.2	Pertinence .....	93
3.5.3	Preservability .....	94
3.5.4	Relevance .....	98
3.5.5	Significance .....	99
3.5.6	Similarity .....	99
3.5.7	Timeliness .....	100
3.6	Metadata Specifications and Metadata Format .....	101
3.6.1	Accuracy .....	101
3.6.2	Completeness .....	104

**CONTENTS**    xiii

3.6.3	Conformance .....	104
3.7	Collection, Metadata Catalog, and Repository.....	106
3.7.1	Collection Completeness .....	106
3.7.2	Catalog Completeness and Consistency .....	108
3.7.3	Repository Completeness and Consistency .....	109
3.8	DL Services .....	109
3.8.1	Effectiveness and Efficiency .....	109
3.8.2	Extensibility and Reusability.....	110
3.8.3	Reliability .....	111
3.9	Case Study: 5SQual .....	113
3.9.1	5SQual Overview .....	114
3.9.2	DL Evaluations using 5SQual .....	121
3.10	Summary .....	133
3.11	Exercises and Projects .....	134
<b>4</b>	<b>Complex Objects .....</b>	<b>137</b>
4.1	Introduction .....	137
4.1.1	What is a CO.....	138
4.1.2	Kinds of CO .....	139
4.1.3	Comparison of COs (DCC, Buckets, OAI-ORE) .....	143
4.2	Related Work .....	144
4.3	Formalization .....	145
4.3.1	CO .....	145
4.3.2	Minimum CO.....	146
4.3.3	ICO .....	147
4.4	Case Study: Fingerprints.....	147
4.4.1	Introduction .....	148
4.4.2	Approach .....	149
4.4.3	Implementation .....	152
4.4.4	Results .....	156
4.5	Summary .....	158

xiv CONTENTS

4.6 Exercises and Projects .....	159
<b>5 Integration .....</b>	<b>160</b>
5.1 Introduction .....	160
5.1.1 The Digital Library Integration Problem .....	160
5.1.2 Hypothesis and Research Questions .....	162
5.2 Related Work .....	162
5.2.1 Semantic interoperability in digital libraries .....	163
5.2.2 Integrated Services .....	163
5.3 Problem Formalization and Overall Approach .....	164
5.3.1 Background on the 5S framework .....	164
5.3.2 Notation and Definitions .....	165
5.3.3 Architecture of an integrated DL .....	167
5.3.4 Integration toolkit: 5SSuite .....	169
5.4 Case Study: An ETANA-DL Experience .....	169
5.4.1 Modeling of Domain Specific Digital Libraries with the 5S Framework .....	170
5.4.2 Visual Mapping Tool: SchemaMapper .....	174
5.5 Summary .....	181
5.6 Exercises and Projects .....	181
<b>6 Subdocuments .....</b>	<b>183</b>
6.1 Introduction .....	183
6.2 Related Work .....	186
6.2.1 Superimposed information .....	186
6.2.2 Subdocuments and hypertext .....	187
6.2.3 Subdocuments and superimposed information in digital libraries ..	188
6.2.4 Subdocuments and annotations .....	189
6.3 Review of select definitions .....	189
6.3.1 Complex objects .....	193
6.4 Formalization and approach to a digital library with superimposed information (SI-DL) .....	194

## CONTENTS xv

6.4.1	5S extensions .....	196
6.4.2	Collections and catalogs .....	203
6.4.3	Services.....	204
6.4.4	SI-DL .....	205
6.5	Case Studies .....	206
6.5.1	Using the SI-DL metamodel to describe SuperIDR .....	206
6.5.2	Flickr (planned) .....	216
6.5.3	Using the metamodel to map subdocuments/annotations between two systems (planned) .....	217
6.6	Summary .....	217
6.7	Exercises and Projects .....	217
<b>7</b>	<b>Ontologies.....</b>	<b>218</b>
7.1	Introduction .....	218
7.1.1	What is an ontology .....	219
7.1.2	Kinds of ontologies.....	221
7.1.3	Ontology Languages .....	223
7.2	Literature Review.....	225
7.2.1	Ontology Engineering.....	225
7.2.2	Ontology and Digital Libraries.....	227
7.3	Formalization .....	228
7.3.1	The Big Picture .....	228
7.3.2	Ontology Components .....	231
7.4	Ontology Engineering.....	232
7.4.1	Methodologies .....	232
7.4.2	Tools .....	235
7.4.3	Reasoning ontology .....	237
7.5	Ontology Applications .....	237
7.5.1	Digital libraries .....	237
7.5.2	Semantic Web .....	238
7.5.3	Focused Crawling.....	239

xvi CONTENTS

7.6	Ontology Evaluation .....	241
7.7	Case study: Crisis, Tragedy, and Recovery (CTR) ontology .....	243
7.7.1	Approach .....	243
7.8	Exercises and Projects .....	246
<b>8</b>	<b>Chapter 8: Classification.....</b>	<b>247</b>
8.1	Introduction .....	247
8.1.1	Intellectual Merit .....	248
8.1.2	ETDs and NDLTD .....	248
8.1.3	Problem Summary .....	249
8.1.4	Research Questions .....	249
8.1.5	Contributions of this project .....	249
8.2	Related Work .....	250
8.2.1	Definitions .....	250
8.2.2	Hierarchical Text Classification .....	251
8.2.3	Naïve Baye's Classifier .....	251
8.2.4	Neural Networks Classifier .....	252
8.2.5	Search Based Strategy .....	253
8.2.6	Comparative Analysis.....	253
8.2.7	Scalability Analysis .....	253
8.3	5S Formalisms .....	254
8.3.1	Streams.....	254
8.3.2	Structures .....	254
8.3.3	Spaces .....	254
8.3.4	Scenarios .....	255
8.3.5	Societies .....	256
8.3.6	Formal Definition Of Classification .....	256
8.3.7	Hierarchical Classification .....	256
8.4	Case Study: Hierarchical Classification of ETDs .....	256
8.4.1	Building a Taxonomy .....	256
8.4.2	Crawling ETD Metadata.....	258

**CONTENTS xvii**

8.4.3	Categorizing ETDs .....	258
8.5	Summary .....	259
8.6	Exercises and Projects .....	260
<b>9</b>	<b>Content-based Image Retrieval .....</b>	<b>261</b>
9.1	Introduction .....	261
9.2	Content-based Image Retrieval .....	262
9.3	Related Work .....	263
9.3.1	Image Descriptors .....	263
9.3.2	CBIR Systems.....	266
9.3.3	Indexing Structures.....	267
9.3.4	Effectiveness Measures .....	267
9.3.5	User Interaction in CBIR Systems .....	268
9.3.6	Applications .....	270
9.4	Formalization .....	272
9.5	Case Study .....	279
9.6	Research Challenges.....	280
9.7	Summary .....	281
9.8	Exercises and Projects .....	281
<b>10</b>	<b>Social Networks in Digital Libraries .....</b>	<b>284</b>
10.1	Introduction .....	284
10.2	Related Work .....	285
10.2.1	Online Communities .....	285
10.2.2	Social and Behavioral Networks.....	286
10.2.3	Social Navigation .....	286
10.3	The Next Generation of Educational Digital Libraries .....	287
10.3.1	Data Collection and Analysis .....	287
10.3.2	DL 1.0 vs. DL 2.0.....	292
10.4	Finding Communities in Digital Library .....	294
10.4.1	Social graph construction using logging and metrics .....	295

xviii CONTENTS

10.5	Analysis of Passive Social Networks .....	296
10.5.1	Graph partitioning.....	296
10.5.2	Topic modeling .....	298
10.5.3	Analyze a pair of passive networks .....	299
10.6	Case Study: The AlgoViz Portal .....	299
10.6.1	Deduced Social Networks .....	300
10.6.2	Community Detection on DSN .....	301
10.6.3	Community Interests.....	303
10.7	Exercises and Projects .....	304
<b>11</b>	<b>Education .....</b>	<b>305</b>
11.1	Introduction .....	305
11.2	Related Work .....	306
11.3	Formalization: Educational DL.....	307
11.3.1	5S Perspective on educational DL.....	309
11.3.2	Federated Search and Harvesting of metadata .....	313
11.3.3	Classification .....	313
11.4	Case Studies .....	316
11.4.1	AlgoViz.....	316
11.4.2	Ensemble .....	318
11.5	Summary .....	319
11.6	Exercises and Projects .....	319
<b>12</b>	<b>Bioinformatics, Scientific, and Simulation Digital Libraries .....</b>	<b>321</b>
12.1	Introduction .....	321
12.2	Related Work .....	322
12.3	Formalism .....	323
12.3.1	Workflows, Content, and Ontologies Definitions .....	324
12.3.2	User Role and Task Definitions .....	329
12.3.3	Service Definitions .....	332
12.4	Case Studies .....	342

**CONTENTS** **xix**

12.4.1 Prototype: Computational Epidemiology .....	342
12.4.2 Prototype: Fingerprint Algorithms .....	343
12.4.3 Prototype: Large-Scale Network Simulations .....	343
12.5 Summary .....	343
12.6 Exercises and Projects .....	346
<b>13 Geospatial Information .....</b>	<b>348</b>
13.1 Introduction .....	348
13.2 Geographic Information .....	349
13.2.1 Raster & Vector Data.....	350
13.2.2 Spatial Relationships and Queries.....	351
13.3 Geographic Information Retrieval .....	353
13.3.1 Geographic information on the Web.....	354
13.3.2 GIR Architecture .....	357
13.4 Multimodal Retrieval for Geographic Information .....	366
13.4.1 Image/Video Retrieval for Geographic Information .....	367
13.5 Related Work .....	370
13.6 Formalization .....	371
13.7 Case Study .....	371
13.8 Summary .....	371
13.9 Exercises and Projects .....	372
<b>14 Security .....</b>	<b>373</b>
14.1 Introduction .....	373
14.1.1 Basic Concepts .....	374
14.2 Related Work .....	375
14.2.1 Content.....	375
14.2.2 Performance .....	378
14.2.3 User .....	378
14.2.4 Functionality .....	381
14.2.5 Architecture .....	381

## **xx CONTENTS**

14.2.6 Quality .....	382
14.2.7 Policy .....	382
14.3 Formalization .....	384
14.3.1 Streams.....	384
14.3.2 Structures .....	385
14.3.3 Spaces .....	386
14.3.4 Scenarios .....	386
14.3.5 Societies .....	387
14.4 Case Study .....	391
14.4.1 Societies .....	391
14.4.2 Streams.....	393
14.4.3 Structures .....	393
14.4.4 Scenarios .....	393
14.4.5 Spaces .....	394
14.5 Summary .....	394
14.6 Exercises and Projects .....	394
<b>15 Text Extraction .....</b>	<b>395</b>
15.1 Introduction .....	395
15.1.1 Rationale and Scope .....	395
15.1.2 Pattern Recognition, Classification, and Structuring.....	395
15.1.3 Problems and applications .....	396
15.2 Related Work .....	396
15.2.1 Algorithms .....	397
15.2.2 Feature Selection .....	399
15.3 Formalization .....	401
15.3.1 Informal Definitions.....	401
15.3.2 Formal Definitions .....	402
15.4 Case Studies .....	403
15.4.1 Document Segmentation .....	403
15.4.2 Results .....	406

**CONTENTS xxi**

15.4.3 Reference Section Extraction.....	408
15.4.4 Evaluation .....	411
15.5 Summary .....	413
15.6 Exercises and Projects .....	413
<b>A Mathematical Preliminaries .....</b>	<b>414</b>
<b>B Glossary.....</b>	<b>418</b>

Editor's Biography

## List of Tables

<b>Table 1.1</b>	List of modules	26
<b>Table 3.1</b>	DL high-level concepts and corresponding DL dimensions of quality with respective metrics	90
<b>Table 3.2</b>	Dimensions of quality and Ss involved in their definitions	91
<b>Table 3.3</b>	Accessibility of VT-ETDs (first column corresponds to the first letter of author's name)	93
<b>Table 3.4</b>	Documents with highest degree of significance	99
<b>Table 3.5</b>	Documents with highest absolute Amsler degree	100
<b>Table 3.6</b>	Completeness of several collections	108
<b>Table 3.7</b>	Analysis of ETANA DL prototype using the metric of Lines of Code	112
<b>Table 3.8</b>	Reliability of CITIDEL services	113
<b>Table 4.1</b>	How standards handle basic CO concepts.	141
<b>Table 4.2</b>	Basic CO concepts from DCC, Buckets, and OAI-ORE perspective.	144
<b>Table 6.1</b>	Examples of the 5 S's in a DL and in an SI-DL.	197
<b>Table 6.2</b>	Digital objects in SuperIDR	212
<b>Table 7.1</b>	Common ontology components and examples	220
<b>Table 8.1</b>	Hierarchical Text Classification Approaches	252
<b>Table 9.1</b>	Coordinates of each image of classes classes 1 and 2 for three different descriptors.	282
<b>Table 10.1</b>	Phases of Data Collection and Analysis	288
<b>Table 10.2</b>	Emerging Themes from the Focus Group Data	290
<b>Table 10.3</b>	Comparison between DL 1.0 and DL 2.0 based on 5S Definitions	292
<b>Table 10.4</b>	Log data for AlgoViz	299

**LIST OF TABLES xxiii**

<b>Table 12.1</b>	Basic Terms and Definitions of 5S formalization [250]	341
<b>Table 14.1</b>	Definition for 5 security services	374
<b>Table 15.1</b>	Comparison of previous extraction approaches	397
<b>Table 15.2</b>	Features for canonical representation extraction	400
<b>Table 15.3</b>	Open source software used	404
<b>Table 15.4</b>	Major reference styles used in ETDs	405
<b>Table 15.5</b>	Drupal modules	406
<b>Table 15.6</b>	Feature sets	410
<b>Table 15.7</b>	Data, used in evaluation, randomly sampled	412
<b>Table 15.8</b>	Result of reference section extraction (P=Precision, R=Recall, F1=F1 score)	413

# List of Figures

Figure 1.1	Information Life Cycle. Adapted from [71].	2
Figure 1.2	CC2001 Information Management Areas.	3
Figure 1.3	Chatham Workshop triangle. Adapted from [369]	6
Figure 1.4	DL Construction Approach.	6
Figure 1.5	Degrees of Structure.	7
Figure 1.6	NSDL Architecture.	8
Figure 1.7	Informal 5S and DL Definitions.	9
Figure 1.8	5S-based Semantics and Relationships of DL Elements.	10
Figure 1.9	Digital Library Content.	12
Figure 1.10	OAI – Repository Perspective.	13
Figure 1.11	OAI — Black Box Perspective.	13
Figure 1.12	Libraries of the Future: JCR Licklider, 1965, MIT Press.	14
Figure 1.13	For More Information (Examples).	15
Figure 1.14	1991 List of Objectives.	16
Figure 1.15	Synchronous Scholarly Communication.	17
Figure 1.16	Asynchronous, Digital Library Mediated Scholarly Communication.	18
Figure 1.17	Digital Libraries Shorten the Chain from.	19
Figure 1.18	Digital Libraries Shorten the Chain to.	20
Figure 1.19	Global Interest.	21
Figure 1.20	Challenges and Benefits - From 2002 Workshop [198]	23
Figure 1.21	DL Curriculum Framework.	25
Figure 1.22	5S map of formal definitions	35
Figure 1.23	Overview of descriptive metadata with example	40
Figure 1.24	A StructuredStream for an ETD (adapted from [458])	42
Figure 1.25	A simple digital object	43
Figure 1.26	Simple indexing service	45
Figure 1.27	A simple hypertext	47

## LIST OF FIGURES xxv

<b>Figure 1.28</b>	Taxonomy of digital libraries terms	50
<b>Figure 2.1</b>	$q$ is a key word named energy.	59
<b>Figure 2.2</b>	$q$ is a structured query named animal bones from the Nimrin site.	59
<b>Figure 2.3</b>	$q$ is an image of 5 spatially related sub-images.	60
<b>Figure 2.4</b>	$q$ is a user's navigation start point.	61
<b>Figure 2.5</b>	Example of $OP_{viz}$	62
<b>Figure 2.6</b>	Example of $cluster_x$ and $cluster_y$ in ETANA-DL	63
<b>Figure 2.7</b>	Example of clustering result	63
<b>Figure 2.8</b>	Example of function $OP_s$ in ETANA-DL	64
<b>Figure 2.9</b>	Example of function $OP_b$ in ETANA-DL	65
<b>Figure 2.10</b>	Constructs for an exploring service	66
<b>Figure 2.11</b>	Sequence of operations	66
<b>Figure 2.12</b>	Relationship among theorems (lemmas) and operations	67
<b>Figure 2.13</b>	An exploring service is a searching service.	67
<b>Figure 2.14</b>	An exploring service is a browsing service.	68
<b>Figure 2.15</b>	An exploring service is a browsing service.	68
<b>Figure 2.16</b>	Example of mapping between navigation path and a structured query	70
<b>Figure 2.17</b>	" $query_i$ " and " $i$ " are associated with the same results.	71
<b>Figure 2.18</b>	Example of Lemma 2	72
<b>Figure 2.19</b>	" $query_{i+1}$ " is refined from " $query_i$ " after browsing.	73
<b>Figure 2.20</b>	Switch from searching to browsing.	74
<b>Figure 2.21</b>	An exploring service is a visualization service.	75
<b>Figure 2.22</b>	Multi-dimensional browsing interface	76
<b>Figure 2.23</b>	Save current navigation path for later use and view records	77
<b>Figure 2.24</b>	Search saucer records	78
<b>Figure 2.25</b>	Equus records are retrieved through basic searching	79
<b>Figure 2.26</b>	Retrieved equus records are organized into 3 dimensions	80
<b>Figure 2.27</b>	Browse the 36 equus records from the Nimrin site after searching	80
<b>Figure 2.28</b>	Initial interface of EtanaViz	81
<b>Figure 2.29</b>	Total number of animal bones across Nimrin culture phrases	82
<b>Figure 2.30</b>	Percentages of animal bones across Nimrin culture phrases	83
<b>Figure 2.31</b>	Impression about ETANA-DL services (mean value)	85

## xxvi LIST OF FIGURES

Figure 2.32	Average time on tasks	86
Figure 3.1	Factors in preservability (all links should be assumed to have “depends on” as their labels)	96
Figure 3.2	Timeliness in the ACM Digital Library	102
Figure 3.3	Average completeness of catalogs in NDLTD (as of February 2004)	105
Figure 3.4	Average conformance of catalogs in NDLTD	107
Figure 3.5	5SQual Architecture	116
Figure 3.6	5SQual Interface - Starting Configuration	118
Figure 3.7	5SQual Interface - Evaluation Identification	119
Figure 3.8	5SQual Interface Selection of Dimensions and Indication of Resources	119
Figure 3.9	5SQual Interface Specification of Parameters	120
Figure 3.10	5SQual Interface - Definition of Target for the Outputs	121
Figure 3.11	5SQual Interface Confirmation of the Configuration	122
Figure 3.12	5SQual report excerpt	124
Figure 3.13	VT-ETD - <i>Accessibility Chart</i>	125
Figure 3.14	VT-ETD - Timeliness Chart	127
Figure 3.15	VT-ETD - Completeness Chart	128
Figure 3.16	VT-ETD - Conformance Chart	129
Figure 3.17	BDBComp - Efficiency Chart	130
Figure 3.18	BDBComp - Reliability Chart	131
Figure 3.19	ACM - Significance Chart	132
Figure 3.20	ACM - Similarity Chart - Co-citation	134
Figure 3.21	ACM - Similarity Chart - Bibliographic Coupling	135
Figure 3.22	ACM - <i>Timeliness Chart</i>	136
Figure 4.1	Architecture for a CO-Based Digital Library.	138
Figure 4.2	Digital Content Component Representation.	142
Figure 4.3	Matching the main concepts of the 5S framework and OAI-ORE [351].	143
Figure 4.4	The Complex Image Object.	147
Figure 4.5	The integration of fingerprint digital libraries.	149
Figure 4.6	The main classes representing the Fingerprint DL.	149

## LIST OF FIGURES xxvii

<b>Figure 4.7</b>	An example of compound object using four digital libraries: (A) Recorded Prints, (B) Distorted Images, (C) Crime Scene Images, and (D) Training Material.	150
<b>Figure 4.8</b>	Samples of images from a Recorded Print DL from the Police.	153
<b>Figure 4.9</b>	Samples of fingerprints from a DL which simulates a crime scene.	154
<b>Figure 4.10</b>	CBIR process for Figure 4.8 - part 11.	155
<b>Figure 4.11</b>	CBIR process for Figure 4.9 - part 3.	156
<b>Figure 4.12</b>	Structure for IndividualDCC.	157
<b>Figure 4.13</b>	XML for the individual aggregation.	158
<b>Figure 5.1</b>	5S definitional structure extended for archaeology	165
<b>Figure 5.2</b>	An example of an integrated DL: ETANA-DL	168
<b>Figure 5.3</b>	5S related tools and their use in developing DLs [250]	170
<b>Figure 5.4</b>	5S related integration toolkit and process	171
<b>Figure 5.5</b>	Structure model for Nimrin	175
<b>Figure 5.6</b>	Scenario model for Halif	175
<b>Figure 5.7</b>	Scenario model for ETANA-DL	176
<b>Figure 5.8</b>	Initial set of mappings for flint tool based on rules and name-based matching	177
<b>Figure 5.9</b>	Megiddo site organization	178
<b>Figure 5.10</b>	Adding FLINT sub-tree as a child of OBJECT in the global schema	180
<b>Figure 5.11</b>	Using the View Only Top Level Leaf Nodes option mapping Vessel Collection	181
<b>Figure 5.12</b>	Name change recommendation based on mapping history	182
<b>Figure 6.1</b>	Searching on subimages and associated information	185
<b>Figure 6.2</b>	Working with information selections in situ.	186
<b>Figure 6.3</b>	A concept map for complex object composition	194
<b>Figure 6.4</b>	Temporal relationship among digital objects in an SI-DL.	194
<b>Figure 6.5</b>	Definitional dependencies among concepts in an SI-DL.	196
<b>Figure 6.6</b>	Example of a presentation specification.	199
<b>Figure 6.7</b>	Example of a subdocument and its components.	202
<b>Figure 6.8</b>	An example of the view in context service.	205
<b>Figure 6.9</b>	Software architecture of SuperIDR	207
<b>Figure 6.10</b>	Species description interface in SuperIDR	208

## **xxviii LIST OF FIGURES**

Figure 6.11	Search in SuperIDR	210
Figure 6.12	Definitional dependencies among concepts in SuperIDR	211
Figure 6.13	Superimposed image complex object	213
Figure 7.1	The meaning triangle	219
Figure 7.2	A portion of a Computer Science ontology	220
Figure 7.3	Ontology examples by their formality	223
Figure 7.4	Ontology language examples based on their formality and expressivity	224
Figure 7.5	An example of KIF representation	225
Figure 7.6	An OWL definition of the class ‘Flight’	226
Figure 7.7	From real world to models and to ontologies (adapted from (add citation)).	229
Figure 7.8	Ontology components.	232
Figure 7.9	Ontology development processes	233
Figure 7.10	Ontology tools for building, merging, and annotation (citation)	236
Figure 7.11	Technology stack used in Semantic Web []	239
Figure 7.12	Example of Ontology: Software concepts and their relations []	240
Figure 7.13	Architecture of Ontology-based Focused Crawler	241
Figure 7.14	Components of ontology evaluation, [482]	242
Figure 7.15	Ontology evaluation approaches for different levels	242
Figure 7.16	Highest level concepts from the current CTR ontology	244
Figure 7.17	An ontology concept expansion process	245
Figure 7.18	A conceptual diagram of an expanded ontology	246
Figure 8.1	A Sample ETD Record in the NDLTD Union Catalog	249
Figure 8.2	Positive and Negative Training Sets for a Node	254
Figure 8.3	ETD Structured Stream	255
Figure 8.4	ETD Categorization Pipeline	257
Figure 8.5	ETD Categorization for ETDs from 8 major US universities in Union Catalog	259
Figure 9.1	Typical CBIR system.	262
Figure 9.2	5S extensions to support content-based image description and related services.	272
Figure 9.3	Example of a structured feature vector.	273

## LIST OF FIGURES xxix

<b>Figure 9.4</b>	(a) The use of a simple descriptor $D$ for computing the similarity between images. (b) Composite image descriptor.	274
<b>Figure 9.5</b>	Image digital object elements.	276
<b>Figure 9.6</b>	Use of a descriptor to extract feature vectors.	276
<b>Figure 9.7</b>	(a) $q$ is an image of (b) 5 spatially related sub-images.	277
<b>Figure 9.8</b>	(a) Spiral approach. (b) Concentric rings approach.	278
<b>Figure 9.9</b>	Image ranking in CTRnet using BIC (a) and SASI (b) descriptors.	280
<b>Figure 9.10</b>	(a) Descriptor 1. (b) Descriptor 2. (c) Descriptor 3.	283
<b>Figure 10.1</b>	(Left) Sample codes with number of references. (Right) Distribution of references in three major themes described in Table 10.2	289
<b>Figure 10.2</b>	Relationship between Resource and User	291
<b>Figure 10.3</b>	Social graphs can be created using different log information.	297
<b>Figure 10.4</b>	Using log to find user communities with similar interest (connection threshold $k=10$ )	301
<b>Figure 10.5</b>	Network Statistics of the DSN for September 2010 (connection threshold $k=10$ )	302
<b>Figure 10.6</b>	Network Statistics of the DSN for October 2010 (connection threshold $k=10$ )	302
<b>Figure 10.7</b>	Clusters found in the DSNs of Figure 10.4	303
<b>Figure 10.8</b>	Page titles of clusters found in the November 2010 DSN	303
<b>Figure 11.1</b>	User Model	310
<b>Figure 11.2</b>	Connexions architecture [39]	315
<b>Figure 11.3</b>	TRAKLA2 architecture [411]	316
<b>Figure 11.4</b>	Algoviz's taxonomy organization translated in faceted browsing	317
<b>Figure 11.5</b>	A catalog entry in AlgoViz	318
<b>Figure 11.6</b>	EDL Concept Map	320
<b>Figure 12.1</b>	Context-specific simulation-based digital object provenance as represented by the Open Provenance Model.	326
<b>Figure 12.2</b>	SimDL's automated, generic web UI snapshot displaying a model's schema.	331
<b>Figure 12.3</b>	5S graph of collaboration and related services.	336
<b>Figure 12.4</b>	Ontology-requiring, model-independent services in SimDL.	337

### xxx LIST OF FIGURES

Figure 12.5	Analysis and Experiment Services in DL framework.	340
Figure 12.6	Example analyzing effects of y-axis displacements on matching quality: (a) skin distortion model selected; (b) distorted images; (c) histogram of y displacement versus sum of matching score; (d) plotting of minutiae spatial distribution.	344
Figure 12.7	Number of minutiae in translation distortion.	344
Figure 12.8	Average minutiae reliability: images distorted by translation.	344
Figure 12.9	Spatial distributions of minutia reliability: (a) original image with 28 minutiae; images which increased (b) and decreased (c) the most in minutia points after distortion (56 and 26 minutiae respectively); and (d)-(f) minutia reliability of each image.	345
Figure 13.1	Top view of North Pole: longitude lines (radii) and latitude lines (concentric circles). Bold lines identify some longitude lines. <i>Source: nationalatlas.gov.</i>	349
Figure 13.2	Cutaway view of Earth showing latitude $45^{\circ}N$ , which is the angle measured from the center of the sphere. <i>Source: nationalatlas.gov.</i>	350
Figure 13.3	Cutaway view of Earth sphere: P is located at latitude $\phi^{\circ}N$ and longitude $\lambda^{\circ}E$ .	351
Figure 13.4	Vector and Raster data can be overlaid. Source ESRI.	352
Figure 13.5	Examples of topographic relationship (from [92])	352
Figure 13.6	Google Search result for neighbors of Campinas.	354
Figure 13.7	Campinas neighbourhood and cities within 50 km.	355
Figure 13.8	Architecture of a Geographic Information Retrieval	359
Figure 13.9	Geoparsing example: Place names recognized in this extract of Wikipedia's page about Campinas (as of 11/03/2011).	361
Figure 13.10	True and false references in geoparsing [318]	362
Figure 13.11	Example from Google Maps with a Point of Interest (POI) selected and search for something nearby enabled.	364
Figure 13.12	Example of results returned by Google Place Search.	365
Figure 13.13	Multimodal geocoding architecture proposal.	370
Figure 14.1	CIA triad	375
Figure 14.2	DRM components and protection technologies, adapted from [186]	377
Figure 14.3	Explicit Trust	383

**LIST OF FIGURES xxxi**

Figure 14.4	Intermediary trust model	384
Figure 14.5	The possible security attacks that can occur at each of the 5S, the Ss are color-coded: Red for Scenarios, Blue for Streams, Green for Spaces, Orange for Structures, and Purple for Societies	388
Figure 14.6	Concept map of the issues related to digital library security	390
Figure 14.7	Architecture of CINET	392
Figure 15.1	Text extraction in digital libraries	396
Figure 15.2	Text extraction from the 5S perspective	399
Figure 15.3	Text extraction from the 5S perspective	402
Figure 15.4	Various steps in text and image extraction	404
Figure 15.5	XML metadata for the token ‘name’ occurring in a PDF file	404
Figure 15.6	Web demo	407
Figure 15.7	System architecture	408
Figure 15.8	Dataflow diagram of reference section extraction	408
Figure 15.9	An example of a chapter reference	409
Figure 15.10	An example of an end reference	409
Figure 15.11	An example of a training data set	411
Figure 15.12	VT ETD-db with reference metadata	412

## CHAPTER 1

# Introduction

by Edward A. Fox and Marcos André Gonçalves

*Abstract:* Digital libraries (DLs) are researched, developed, implemented, deployed, and used by millions of people in a wide variety of domains. They include advanced information systems that address the full information life cycle, facilitating asynchronous communication, across time and space, and enabling new methods for scholarly communication in our flat world. Since there is strong motivation to build DLs, they are studied by many of those doing advanced work in computer, information, or library science. Though there are a variety of definitions related to DLs, and varied perspectives to consider, few have adopted a formal approach. The 5S framework provides a theoretical foundation to define key constructs, building upon: Societies, Scenarios, Spaces, Structures, and Streams. Using these 5Ss, 24 definitions of important concepts are provided, leading ultimately to a definition of a minimal digital library. Further, 5S guides us to develop a taxonomy for the DL field.

### 1.1 CONTEXT

Information is a fundamental human need. This need is universal for individuals, leading to current interest in personal information management (PIM, see Chapter 10) [322], and is manifest for groups of people as well. Accordingly, institutions have arisen to help us with this need, including libraries, archives, museums, and a variety of information centers, such as for corporations or governments. Developing and operating these institutions involves information management, which requires planning, acquisition, and a variety of services. Special terms have developed for those who make use of these institutions, including: client, customer, patron, and user.

Today, (computerized) information systems help us meet that need. Digital libraries are perhaps the most advanced, integrated, and comprehensive information systems, supporting work with information, across its entire life cycle [71].

Fig. 1.1 makes clear that each phase of the information life cycle flows into the next; one turn of the wheel leads to another. Thus, decisions made by an author, in a particular social context, effect how easy it will be to index, store, distribute, access, preserve, and reuse a document and its content. One labeling of parts of the cycle emphasizes level of activity, as shown on the outside of the circle. Key high level operations corresponding to

## 2 1. INTRODUCTION

that labeling include creation, searching, and utilization. On the other hand, focusing on more detailed operations, we find in the inside of the circle: authoring, organizing, retrieving, filtering, and modifying. Thus, digital libraries [390, 200, 159, 27, 73, 56, 381, 680, 74] build broadly upon advances in electronic publishing, information retrieval, networking, the World Wide Web, text mining, and other aspects of information management.

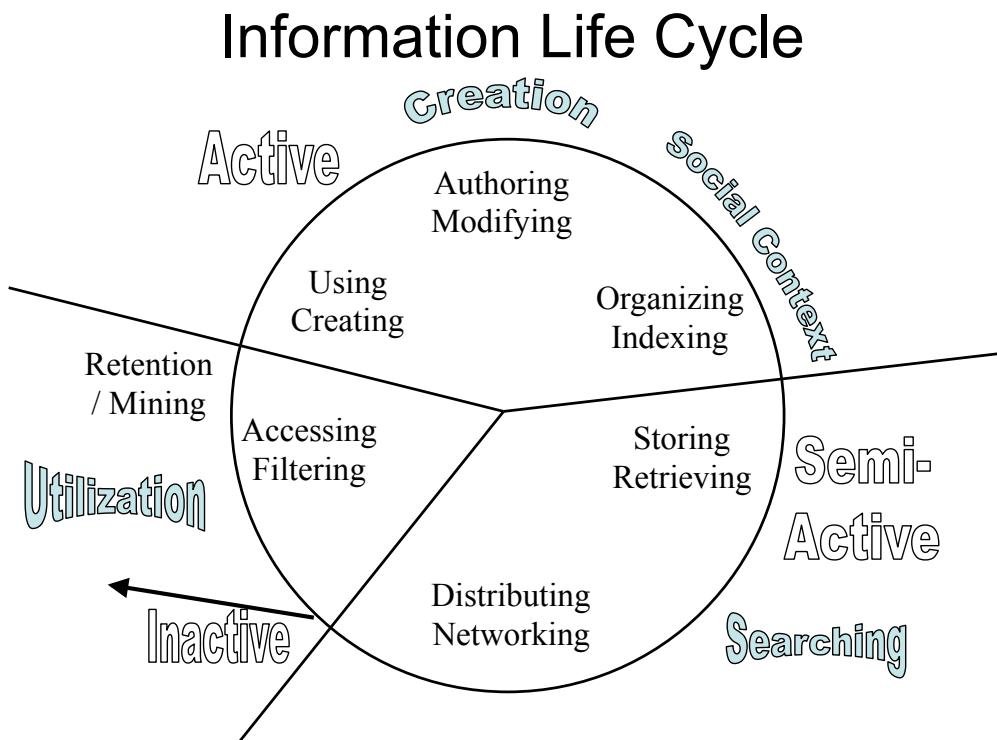


Figure 1.1: Information Life Cycle. Adapted from [71].

Three broad disciplines especially connect with digital libraries. From the first part of the name, there is a clear connection with computing, including computer science. Also from the name, there is a clear connection with libraries, including library science. Further, as is discussed above, information systems, information technology, and information science clearly connect.

These disciplines are closely related. Thus, as can be seen in Fig. 1.2, in the ACM/IEEE-CS Computing Curriculum from 2001, fourteen sub-areas were specified for

### 1.1. CONTEXT 3

the area of information management. The last of those, digital libraries, may be thought of as a capstone for the set of sub-areas.

CC2001 Information Management Areas	
IM1. Information models and systems*	IM8. Distributed DBs
IM2. Database systems*	IM9. Physical DB design
IM3. Data modeling*	IM10. Data mining
IM4. Relational DBs	IM11. Information storage and retrieval
IM5. Database query languages	IM12. Hypertext and hypermedia
IM6. Relational DB design	IM13. Multimedia information & systems
IM7. Transaction processing	IM14. Digital libraries

\* Core components

Figure 1.2: CC2001 Information Management Areas.

Fig. 1.2 can be thought of as one roadmap into this book. Information models and data modeling are at the heart of our emphasis on the 5S framework (introduced later in this chapter); they also are touched upon in every chapter, especially including chapters 1, 3, 4, 6, 10, and 12. Databases are a key aspect of digital libraries, supporting abstractions like metadata record, catalog, and repository—they are discussed in chapters 5, 11, and 12. Transactions undergird all of the digital library services, in some cases related to security issues as in Chapter 14. Distributed concerns are key to work on integration, as in Chapter 5. Data mining closely relates to the extraction work covered in chapters 7, 8, and 15. Information retrieval is discussed in a variety of contexts, especially in chapters 2, 9, and 13. Hypertext and hypermedia also are discussed throughout, in part through their undergirding of the Web, but are given special attention in Chapter 2. Multimedia is considered particularly in chapters 9 and 13.

Digital libraries also connect with artificial intelligence, especially knowledge management, as is considered in chapters 7 and 8. Linguistics, in particular computational linguistics, is at the heart of text extraction, as in Chapter 15, but also underlies indexing, which supports information retrieval, discussed above.

#### **4 1. INTRODUCTION**

Digital libraries emerged at the same time as the global networking infrastructure was proliferating rapidly, including the spread of the commercial Internet and the rise of the World Wide Web. Hence, many standards that are followed in the digital library field come from Internet organizations, including the World Wide Web Consortium (W3C, see <http://www.w3.org>). Chapter 12, focused on e-science, is closely connected with such infrastructure, and describes extending the cyberinfrastructure for science. Chapter 14, contrasting with the focus in Chapter 11 on open access, also ties in with infrastructure, considering how to make that more secure.

Thus, there is a broad context for the field of digital libraries (DLs), providing support for DL emergence and development. The next section provides additional background, from a variety of perspectives, including related definitions.

### **1.2 BACKGROUND**

In the early days of the digital library field, many were concerned with names and definitions. For example, terms like *electronic library* and *virtual library* were considered, but ultimately, around 1991, *digital library* (DL) became the widely accepted term.

#### **1.2.1 DEFINITIONS**

Not as much agreement was achieved with definitions of *digital library*, however. Some of the competing visions cover the following alternative perspectives [72]:

- content, collections, and communities
- institutions or services
- databases

From a 1996 workshop, two complementary views emerged [70], arguing that digital libraries are:

1. “a set of electronic resources and associated technical capabilities for creating, searching and using information. In this sense they are an extension and enhancement of information storage and retrieval systems that manipulate digital data in any medium (text, images, sounds; static or dynamic images) and exist in distributed networks. The content of digital libraries includes data, metadata that describe various aspects of the data (e.g., representation, creator, owner, reproduction rights) and metadata that consist of links or relationships to other data or metadata, whether internal or external to the digital library.”
2. “constructed, collected and organized, by (and for) a community of users, and their functional capabilities support the information needs and uses of that community.”

## 1.2. BACKGROUND 5

They are a component of communities in which individuals and groups interact with each other, using data, information and knowledge resources and systems. In this sense they are an extension, enhancement and integration of a variety of information institutions as physical places where resources are selected, collected, organized, preserved and accessed in support of a user community. These information institutions include, among others, libraries, museums, archives and schools, but digital libraries also extend and serve other community settings, including classrooms, offices, laboratories, homes and public spaces.”

Representative additional definitions include:

- “Digital libraries are organizations that provide the resources, including the specialized staff, to select, structure, offer intellectual access to, interpret, distribute, preserve the integrity of, and ensure the persistence over time of collections of digital works so that they are readily and economically available for use by a defined community or set of communities” [663]
- “A digital library is an organized and focused collection of digital objects, including text, images, video, and audio, along with methods of access and retrieval, and for selection, creation, organization, maintenance, and sharing of the collection.” [677]
- “An organization, which might be virtual, that comprehensively collects, manages and preserves for the long term rich digital content, and offers to its user communities specialized functionality on that content, of measurable quality and according to codified policies.” [97]

### 1.2.2 PERSPECTIVES

Though later in this chapter we give a very precise definition of *digital library*, perhaps the easiest characterization to remember is what is shown in Fig. 1.3.

Clearly, a comprehensive view of DLs must include people and content, as well as applications of technology. The labels on the sides of the triangle coincide nicely with phases of the information life cycle, discussed above. Fig. 1.4 flattens the circle or triangle, putting the DL system in the middle, connecting users and content. From such a systems perspective, many concerns arise, such as how the DL can be built. While the DL may be thought of as a monolithic single entity—convenient regarding naming, purchasing, and operation—in many cases, the DL software is composed of (distributed) components or modules that work together.

The variety of perspectives on digital libraries is a natural consequence of their many aspects or *facets*. Each of those covers a range of possibilities, reflecting the different uses that people desire. For example, one dimension concerns Access vs. Preservation. This is so important as to be institutionalized; traditionally, libraries emphasize access while archives

## 6 1. INTRODUCTION

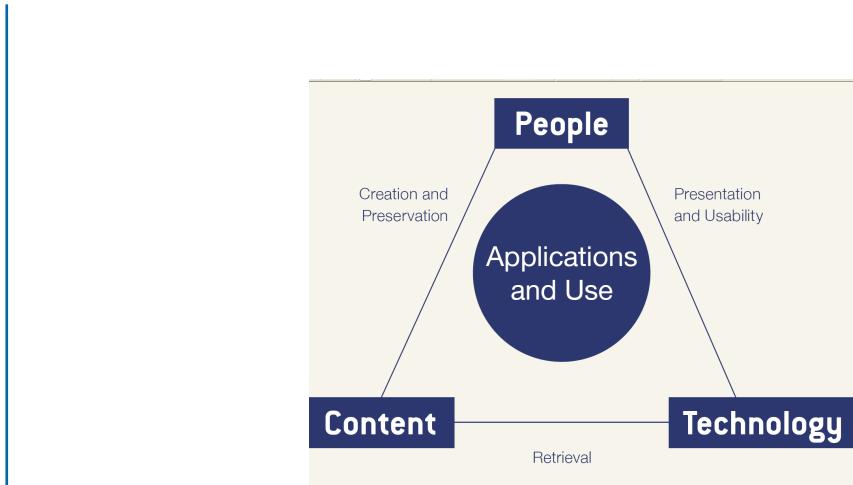


Figure 1.3: Chatham Workshop triangle. Adapted from [369]

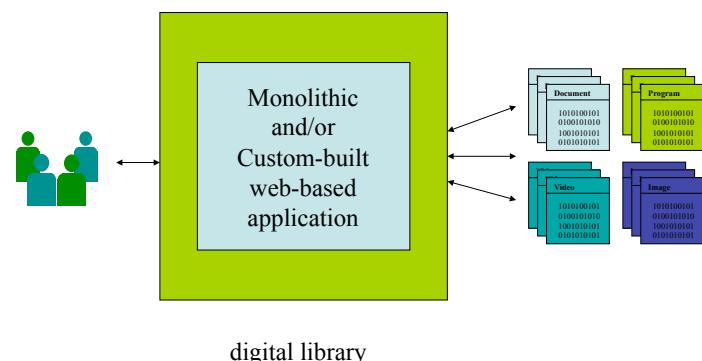


Figure 1.4: DL Construction Approach.

## 1.2. BACKGROUND 7

emphasize preservation. Focusing on the issue of access, another controversy concerns Free vs. High Quality. While in actuality these reflect two different dimensions, namely cost and quality, many assume that what is free will have lower quality than what has a moderate to high cost. Clearly that is not necessarily true.

From a scientific perspective, it is more efficient to choose a set of independent dimensions to describe the range of options and choices related to DLs. Chapter 3 illustrates this in detail, focusing on the quality dimension.

Another important dimension relates to organization. One dichotomy often posed about DLs is Managed vs. Comprehensive. Thus, a library is managed while the WWW is unmanaged (but closer to being comprehensive). Regarding degree of organization, in this book we generally use the term *structure*. Fig. 1.5 gives a high level categorization of structure, naming popular technologies and showing where they fit along that dimension. We argue that DLs must be organized, thus having a moderate degree of structure.

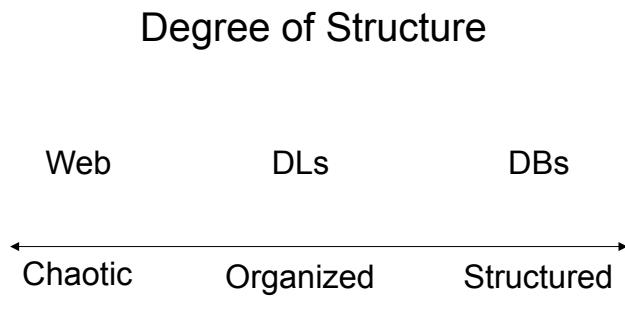


Figure 1.5: Degrees of Structure.

Focusing more on technology, a key dimension regarding DLs is Centralized vs. Distributed. For a personal or other small DL, centralization is common, though with the rapid proliferation of mobile and cloud technologies, even for those applications, centralization is diminishing. From a logical perspective, however, many DLs are centralized in an institution, even though, from a physical or device perspective, they are distributed. Thus, as can be seen in Fig. 1.6, the National Science Digital Library (NSDL), which is discussed in Chapter 11, while viewed from outside as a single entity, in actuality has a distributed

## 8 1. INTRODUCTION

architecture, with: multiple portals, multiple servers, and a broad variety of databases and content collections.

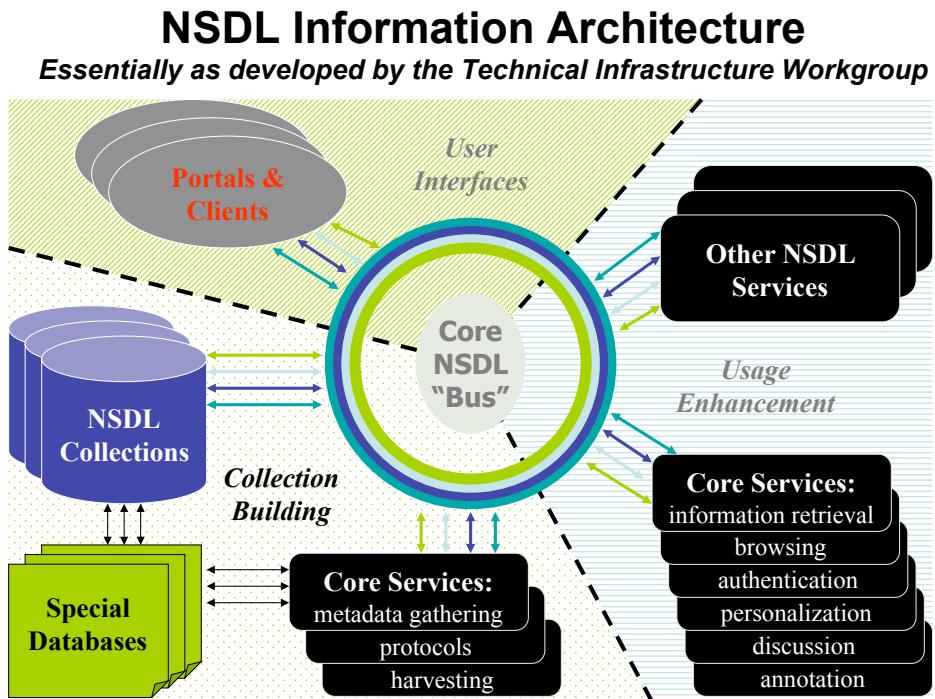


Figure 1.6: NSDL Architecture.

Returning to the challenge of defining DLs, some ask what would be considered a DL, and what would not. Clearly, a DL can be a *digitized library*, but need not be; frequently, DLs go well beyond just being an electronic version of a paper-based library. More generally, while they *can* be a deconstruction of existing systems and institutions, which simply moves those systems to an electronic box in a library, DLs, at the least, usually add both value and functionality. Indeed, often they introduce a better solution to particular human needs. In the broadest sense, then, DLs give us new ways to deal with data, information, and knowledge.

To capture this breadth, and as a lead-in to the discussion later in this chapter, we give our simplified working definition of DL in Fig. 1.7. Further, as can be seen in Fig. 1.8, the five constructs identified (each starting with 'S', hence 5S) can in turn be used to

## 1.2. BACKGROUND 9

describe all of the key aspects of DLs, including their semantics and inter-relationships. For example, we can read off from Fig. 1.8 many important facts about DLs:

- Video contains images and audio.
- A metadata specification describes a digital object.
- Metadata specifications are included in a catalog.
- A catalog describes a collection.
- A repository stores a collection and a catalog.
- A service manager runs a service.
- A service is included in a scenario.
- An actor participates in a scenario.
- A scenario includes events.
- An event executes an operation.

### Informal 5S & DL Definitions

DLS are complex systems that

- help satisfy info needs of users (**societies**)
- provide info services (**scenarios**)
- organize info in usable ways (**structures**)
- present info in usable ways (**spaces**)
- communicate info with users (**streams**)

Figure 1.7: Informal 5S and DL Definitions.

Some additional observations are clear from Fig. 1.8. First, there is a natural split between the left and right parts. Three of the Ss are closely connected on the left, and relate especially to content. The other Ss, closely connected with people and services, also have strong cohesion. Second, some key elements of a DL, such as a *digital object*, are defined in terms of two or more of the Ss. Thus, while many DL elements can be defined solely or mostly using one of the Ss, others may have aspects drawn from multiple Ss; generally they are more complicated abstractions, e.g., an index. Third, the 5S approach, introduced

## 10 1. INTRODUCTION

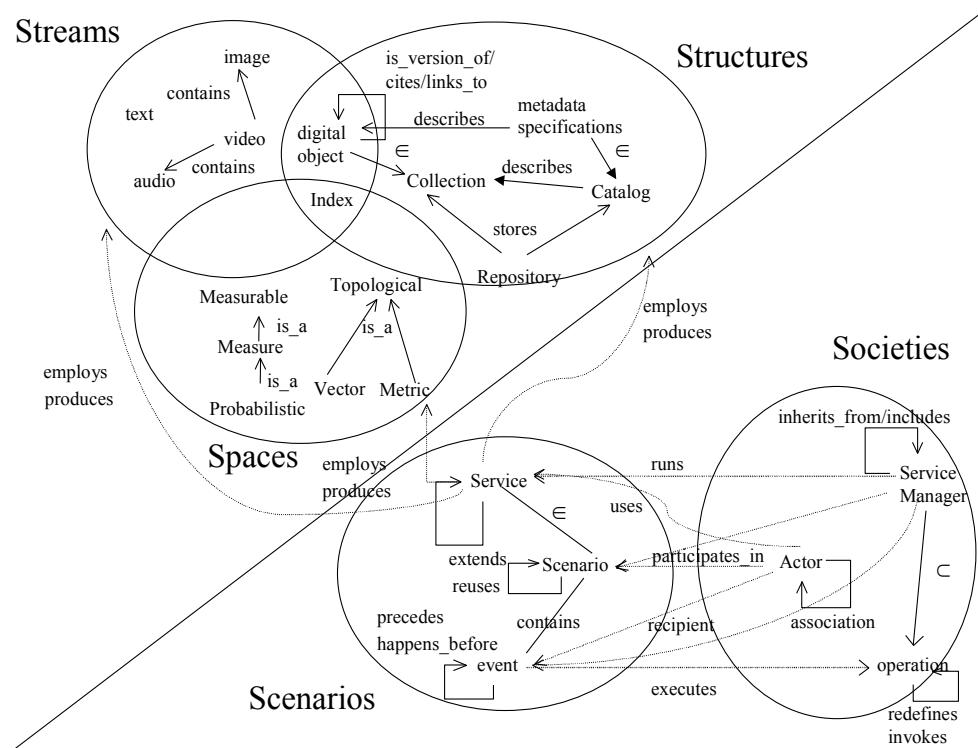


Figure 1.8: 5S-based Semantics and Relationships of DL Elements.

superficially above, and discussed in depth below, is definitional, not programming-oriented. Thus, each concept and construct has a mathematical as well as an intuitive definition. While a particular digital library is made up of programs, and they are based on some approach, e.g., object-oriented, the 5S framework can lead to implementations using any programming paradigm.

While definitions are important, they become really meaningful when viewed from a variety of perspectives. One key perspective on DLs regards content. This is important; recall the left portion of Fig. 1.8, which addresses content representation. From the perspective of the information life cycle, content creation is of interest. Nowadays, with the widespread deployment of electronic publishing and other tools, much of the content in a DL is *born digital*. On the other hand, there are many other objects in the world that cannot be *in* DLs. Nevertheless, they can be described, using some metadata specification or format, leading to a metadata record. Sometimes, too, one or more surrogate representations of real-world objects also are created, typically through a digitization process. Thus, one might have a number of digital image files from photographing or scanning an art object, at varying levels of detail, serving as surrogates for the real-world object. DLs that have both born digital and digitized content often are referred to as hybrid DLs.

While the traditional view of a library is of books, in the digital age it is common to include many other types of content. Fig. 1.9 illustrates the broad range of content types that have been included in DLs. While some DLs cover just one type of content, others cover a range of types. Accordingly, it is important that key DL services, like searching and browsing, operate across those types. Thus, Chapter 9 discusses how information retrieval can be applied to digital images.

As a result of such support for searching and browsing, people gain access to information. Accessibility has many aspects. Universal access refers to providing access for all people, in all places (i.e., ubiquitously), and at all times (i.e., 24/7/365). Accessibility also connotes accommodating special needs or disabilities, including regarding perception, including visual (covering scale and color) and auditory. Further, accessibility refers to mobile use, relating to office, home, laptop, PDA, and a wide variety of other mobile devices. In addition to spatial coverage, there is access across time, often through an archive, which, if sustainable, aims toward permanence. Some archives result from a process of collection, but others are less costly, arising through donation or self-archiving [281]. Supporting such archives, the *Open Archives Initiative* [148, 477] arose in 1999, leading first to a protocol to support metadata harvesting (OAI-PMH) [476]. This allowed development of simple repositories; see Fig. 1.10. These repositories can harvest from other repositories, as can be seen in Fig. 1.11, making use of the protocol's support for selective and/or incremental collection, thus supporting various approaches to aggregation (e.g., over space, organization, or topic).

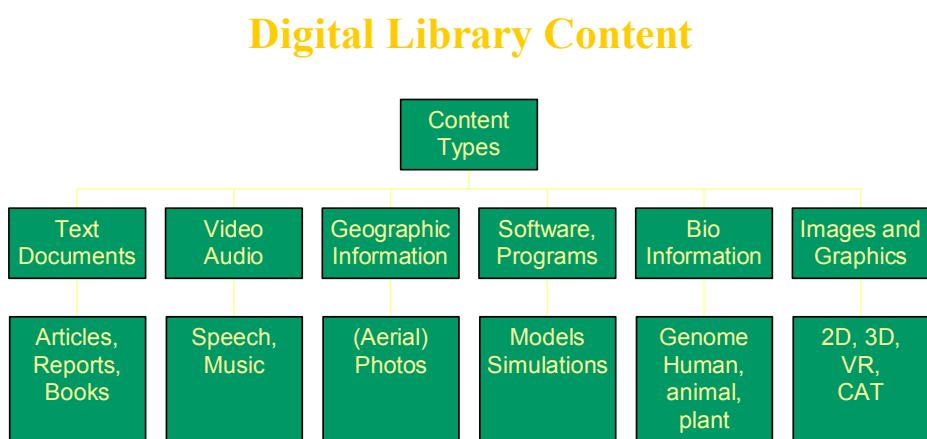


Figure 1.9: Digital Library Content.

### OAI – Repository Perspective

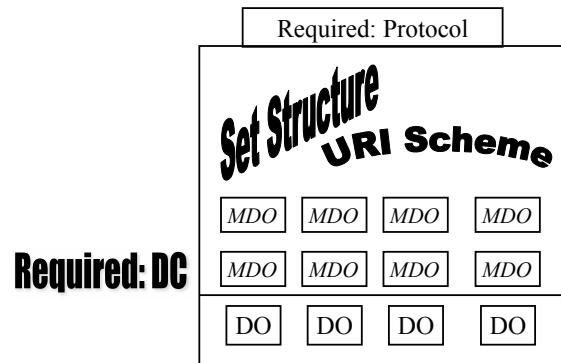


Figure 1.10: OAI – Repository Perspective.

### OAI – Black Box Perspective

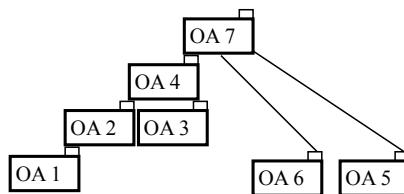


Figure 1.11: OAI — Black Box Perspective.

## 14 1. INTRODUCTION

The concept of being open, additionally, has a variety of aspects. Besides *open archives*, there are *open standards*, sometimes supported by *open source software*. While all three of these relate to digital libraries, open archives are most closely and specifically connected.

Though only recently made simple through OAI-PMH, the model of aggregation illustrated in Fig. 1.11 is not an new idea. Fig. 1.12 summarizes the similar, then almost prescient, vision of Licklider, from almost 50 years ago.

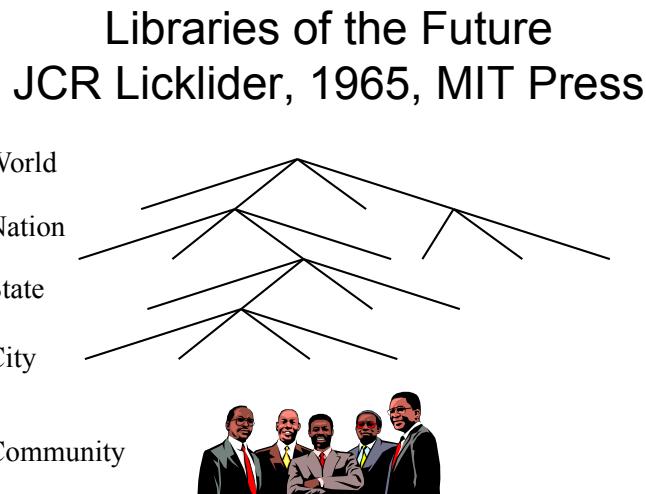


Figure 1.12: Libraries of the Future: JCR Licklider, 1965, MIT Press.

Actually, from an historical perspective, there has been a rather continuous development of technologies leading toward modern digital libraries, beginning from the earliest days of computing [439, 627, 237]. Thus, in 1945, Vannevar Bush<sup>1</sup> conceived of *Memex* [87], often viewed as the clearest early vision leading to hypertext, hypermedia, and the World Wide Web.

MIT, home to Dr. Bush, also hosted key early automated information systems, including Project Intrex, as well as TIP, hosted by the MIT Library, which more recently has been visible as the home for DSpace [162, 624, 625, 623].

<sup>1</sup>President Roosevelt's Science Advisor, Director of the Manhattan Project, and Founder of the US National Science Foundation

### 1.3. MOTIVATION 15

Other parts of the history of digital libraries come from connection with funding initiatives. In the USA, in the early 1990s, support was provided for a number of workshops to launch the field [200]. These led to the (first) Digital Library Initiative [282], followed by a second program, DLI2 [195, 380, 265]. In Europe, too, there was governmental support, largely connected through a Network of Excellence, DELOS [536, 105, 12].

Partly encouraged by these initiatives and programs, focused digital library conferences emerged. Most notable among these are JCDL (usually in North America), ECDL (now TPDL, usually in Europe), and ICADL (usually in Australasia). Information on these, as well as federations and associations, magazines and journals, and research laboratories—all important for learning more about DLs, is summarized in Fig. 1.13 [390, 200, 159, 27, 73, 56, 381, 680, 74, 203].

## For More Information (Examples)

- **Magazine:** [www.dlib.org](http://www.dlib.org)
- **Books:** Online: Fox, <http://fox.cs.vt.edu/DLSB.html> (1993)
  - MIT Press: Arms (1999), Bishop (2003), Borgman (2003, 2010), Licklider (1965)
  - Morgan Kaufmann: Witten... (several), Lesk (2<sup>nd</sup> edition)
- **Conferences**
  - TPDL: [www.tpdil2011.org](http://www.tpdil2011.org)
  - ICADL: [www.icadl.org](http://www.icadl.org)
  - JCDL: [www.jcdl.org](http://www.jcdl.org)
- **Associations**
  - ASIS&T DL SIG: [www.asis.org/SIG/dl.html](http://www.asis.org/SIG/dl.html)
  - IEEE TCDL: [www.ieee-tcdl.org](http://www.ieee-tcdl.org)
- **NSF call:** [www.nsf.gov/pubs/1998/nsf9863/nsf9863.htm](http://www.nsf.gov/pubs/1998/nsf9863/nsf9863.htm)
- **Labs:** VT: [www.dlib.vt.edu](http://www.dlib.vt.edu), TAMU: [www.csdl.tamu.edu/](http://www.csdl.tamu.edu/)

Figure 1.13: For More Information (Examples).

## 1.3 MOTIVATION

In 1991, in connection with the Envision project [209, 207] we interviewed library and information scientists, as well as those interested in computing and computing education, to identify requirements in the new area of digital libraries. Fig. 1.14 summarizes our findings. The first entry summarizes long-held desires for access to all knowledge, at any time, from our desktops [667]. The second entry connects these wishes to the advance of information

## 16 1. INTRODUCTION

systems, and to a goal of this work: to develop a comprehensive theoretical foundation, 5S. The third entry extends from the desktop to other devices and settings, highlights concern for quality (see Chapter 3), and points out the need for lowering barriers and decreasing the digital divide [472], as well as addressing the challenges of library budgets. The fourth entry ties in with the need for learning and for advancing scholarship, which builds upon prior knowledge. The fifth entry makes clear that there is a role for librarians, as partners not gatekeepers, since individuals should develop better skills for collecting and managing information, not delegating all responsibility to others. This fits well with the next entry, calling for universities to play more active roles; today we see them managing institutional repositories and many other information or knowledge management systems in addition to traditional library services. The seventh entry elaborates on this, which has led to handling of electronic theses and dissertations, e-portfolio systems, and courseware management systems. The final entry goes beyond basic quality concerns, to address system, user base, utility, and economic concerns.

### 1991 List of Objectives

- World Lit.: 24hr / 7day / from desktop
- Integrated “super” information systems: 5S:  
Table of related areas and their coverage
- Ubiquitous, Higher Quality, Lower Cost
- Education, Knowledge Sharing, Discovery
- Disintermediation -> Collaboration
- Universities Reclaim Property
- Interactive Courseware, Student Works
- Scalable, Sustainable, Usable, Useful

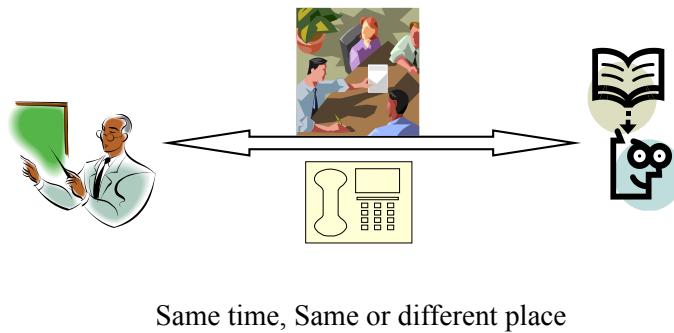
Figure 1.14: 1991 List of Objectives.

Given such calls for digital libraries, one might ask regarding *why* people have been interested in digital libraries. Clearly, there is great interest, as is illustrated in Fig. 1.13, but what are the deep reasons? As was discussed in Section 1.1, humans need information. As is clear from the rapidly growing interest in computer-supported communication (e.g., using Facebook), and is elucidated in Chapter 10, humans also need to communicate.

For millennia, people have been communicating and sharing information synchronously, that is, at the same time, generally by being in the same place, as is illustrated in Fig. 1.15. With the advent of telecommunication devices, synchronous communication had been ex-

tended beyond the limitations of space, either directly, as with telephone, or in broadcast mode, as with radio and television.

## Synchronous Scholarly Communication



Same time, Same or different place

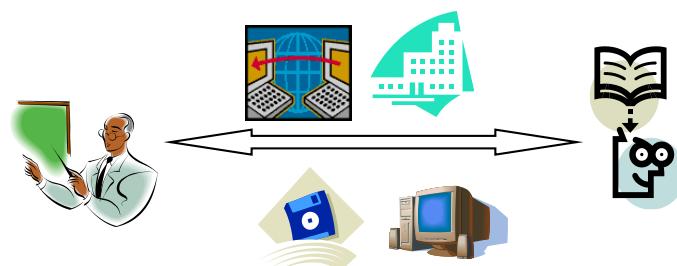
Figure 1.15: Synchronous Scholarly Communication.

To further extend communication and sharing of information across time, asynchronous methods are needed, as is illustrated in Fig. 1.16. These allow information to be recorded, so people can access it later, either in the same place or in different places, if something carrying that information is moved (sometimes by way of copies). Support for such asynchronous communication and access to information has been greatly enhanced for over 600 years through technologies related to paper and printing [73]. But with the emergence of digital libraries, such communication can be vastly extended and expanded.

Asynchronous communication has been particularly important regarding scholarly communication, including journals that first arose about 350 years ago, supported by authors, editors, and reviewers. As can be seen in Fig. 1.17, there is a chain of steps, or workflow, aimed to add value and ensure quality. Institutions like publishers, abstracting and indexing (A&I) groups, and book consolidators (that aggregate shipments to libraries), support the process, but also lead to added costs.

Today, as can be seen in Fig. 1.18, the entire process can be flattened [225]. While the same process as has been carried out for centuries is possible, other scenarios can be supported as well. For example, an author can upload a work into a digital library, make

## Asynchronous, Digital Library Mediated Scholarly Communication



Different time and/or place

Figure 1.16: Asynchronous, Digital Library Mediated Scholarly Communication.

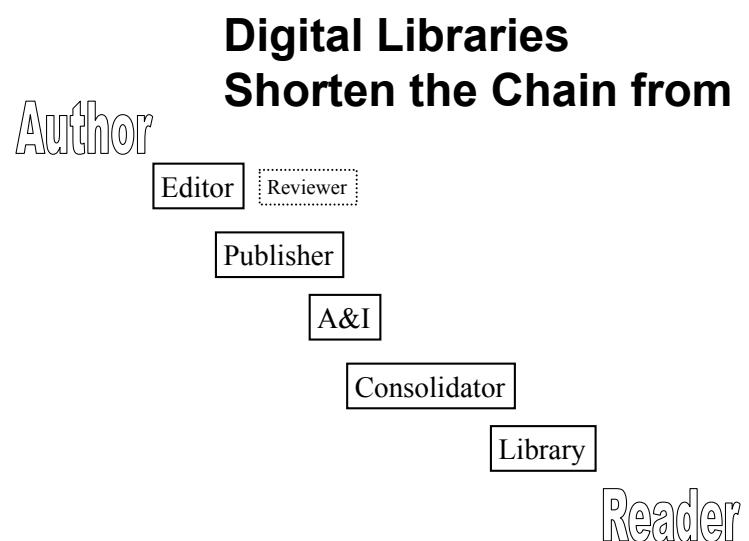


Figure 1.17: Digital Libraries Shorten the Chain from.

## 20 1. INTRODUCTION

it accessible to friends and colleagues, sometimes using a pre-print service or institutional repository. Over time, the work can be refined, and new versions added. Editors might notice that a work is having some impact, work with the author, typically with the aid of reviewers, and have a publisher's imprimatur added, whereupon the same work might be aggressively publicized. Since the same person may play different roles at different times, interacting with the digital library, less time may be needed to become familiar with its user interface. Additional savings in cost as well as time are possible, due to automation and flexible workflows. Further, new approaches to electronic publishing are possible, such as integration of conference and journal publishing activities.

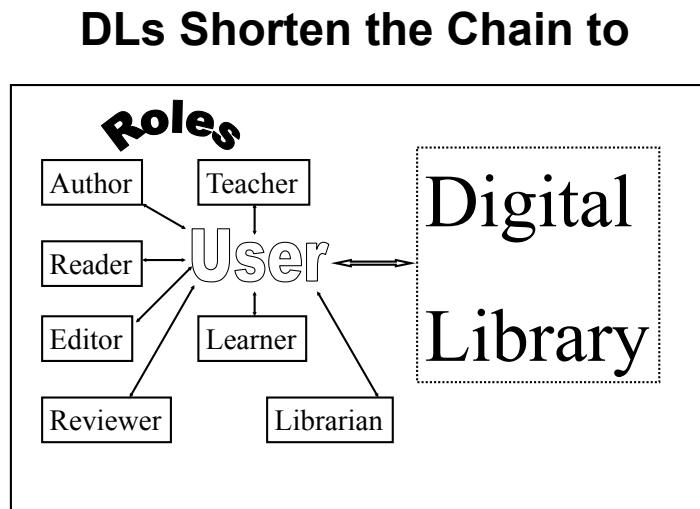


Figure 1.18: Digital Libraries Shorten the Chain to.

Since digital libraries can serve such a wide variety of needs, they have been of interest to many world leaders. For example, during a workshop to explore collaboration between the USA and India regarding digital libraries [644], the President of India called to encourage our deliberations and to share a poem he had written on the topic. Fig. 1.19 explains some of the reasons for such interest. What is particularly exciting is that digital libraries may help expand the base of understanding among the peoples of the earth, as we appreciate our respective cultural heritages, and advance in education and growth.

The broad range of applicability of digital libraries is summarized in Fig. 1.20, which arose from a workshop exploring collaboration between the USA and South Korea [198].

## Why of Global Interest?

- **National projects** can preserve antiquities and heritage: cultural, historical, linguistic, scholarly
- Knowledge and information are essential to economic and technological **growth, education**
- DL - a **domain for international collaboration**
  - wherein all can **contribute** and **benefit**
  - which leverages investment in **networking**
  - which provides useful **content** on Internet & WWW
  - which will **tie nations and peoples together** more strongly and through **deeper understanding**

Figure 1.19: Global Interest.

## 22 1. INTRODUCTION

The first column lists a broad set of application domains, in which digital libraries contribute, followed by an entry for a row so that cross-cutting issues can be considered. The second column lists representative institutions for each domain, while the third column gives examples of such institutions. Column four identifies technical challenges that should be addressed so digital libraries can serve better in a domain, leading to the benefits and positive impact summarized in the last column. For example, in the education domain, institutions served include schools, colleges, and universities; the NSDL (see Chapter 11) is an example of a large digital library developed to help them. Regarding education, better knowledge management should help in part because of increased access to data, and reuse of educational materials also should increase access to resources that often are developed at great cost. Chapter 11 explores these matters in great depth, since education is so important in the modern Information Age, and since sharing can help us address ongoing economic woes connected with educational institutions, while at the same time allowing enhanced learning through use of interactive multimedia resources.

As can be seen from the row in Fig. 1.20 about art and culture, a key benefit would be increased global understanding. For that to result, we need technical advances to improve our ability to digitize a wide variety of types of objects, supplemented by cheaper and more accurate description schemes that can lead to comprehensive catalogs. Regarding the row on science, there is particular need to support e-science, as is discussed in Chapter 12. For example, we must carefully describe scientific experiments, so they are reproducible, and must store data about those experiments for reuse, including parameters, raw data, results, and derived publications. This is of great importance as computing helps in the shift of emphasis in science toward a fourth paradigm [292], where data and computers are essential, as in simulation studies.

The last row mentions broad challenges, including preservation. Unless digital libraries will stand the test of time, and their content and services will be available ongoing, investment may be questioned, and those comfortable with other ways to access information may withhold trust. There are many works touching on this topic, and ongoing research [662, 383, 133, 156, 551, 522, 547, 311, 300, 167]. Fortunately, there already are practical [375] and promising approaches [397], if there is sufficient will, planning, and investment in this important problem.

These challenges reinforce and supplement what was articulated in 1991, summarized in Fig. 1.14. In particular, we must continue to work toward scalability, sustainability, interoperability, and integration. These points are discussed repeatedly in subsequent chapters.

## 1.4 DIGITAL LIBRARY CURRICULUM

Since digital libraries, and their many off-shoots (e.g., organizations like Google, and types of systems such as institutional repositories or content management systems) will be with us

J  
u  
n  
e  
  
2  
0  
0  
2

D L  C h a l l e n g e s  B e n e f i t s	Application Domain	Related Institutions	Examples	Technical Challenges	Benefit / Impact
Publishing	Publishers, Eprint archives	OAI		Quality control, openness	Aggregation, organization
Education	Schools, colleges, universities	NSDL, NCSTRL		Knowledge management, reuseability	Access to data
Art, Culture	Museum	AMICO, PRDLA		Digitization, describing, cataloging	Global understanding
Science	Government, Academia, Commerce	NVO, PDG, SwissProt, UK eScience, European Union Commission		Data models	reproducibility, faster reuse, faster advance
(e) Government	Government Agencies (all levels)	Census		Intellectual property rights, privacy, multi-national	Accountability, homeland security
(e) Commerce, (e) Industry	Legal institutions	Court cases, patents		Developing standards	Standardization, economic development
History, Heritage	Foundations	American Memory		Content, context, interpretation	Long term view, perspective, documentation, recording, facilitating, interpretation, understanding
Cross-cutting	Library, Archive	Web, personal collections		Multi-language, preservation, scalability, interoperability, dynamic behavior, workflow, sustainability, ontologies, distributed data, infrastructure	Reduced cost, increased access, preservation, democratization, leveling, peace, competitiveness

Figure 1.20: Challenges and Benefits - From 2002 Workshop [198]

## 24 1. INTRODUCTION

for the foreseeable future, it is important that there be suitable training and education for digital librarians, as well as for those working in related areas, like information retrieval.

In the computing field, recommendations for curricular work achieved widespread popularity at least back to 1968 [30]. Current programs build upon the 2001 recommendations [108], updated in 2008, with a new revision planned for 2013. But for focused fields, like digital libraries, a more in-depth analysis is needed. This builds upon related work, for information retrieval [201] and multimedia [210, 208].

In 2005, a team at Virginia Tech and the University of North Carolina at Chapel Hill began work on a tailored curriculum for digital libraries. We reviewed the existing literature to identify topical coverage, and used computer analysis to help with our identification of sub-areas. Our work led to a website [216] and to a parallel representation in Wikiversity [215] so that the community could more easily work to improve the resources developed.

Covering a field can be carried out from a variety of perspectives. Some are interested in questions like: What? Why? How? Others prefer an historical approach, considering origins, evolution, current status, research problems, and future work. Those with an economic background have particular interest in costs, equity, and sustainability. Those with social concerns focus on users (sometimes called patrons), management, and support of collaboration. Those with a technical bent may have backgrounds in human-computer interaction, hypertext/hypermedia, information retrieval, information science, library science, or Web technologies. Accordingly, as we developed curricular resources [520], we focused on many different modules (i.e., detailed lesson plans), each to address a particular interest.

Thus, we aimed to support the needs of those in library or information science programs [519]. At the same time, we prepared modules for those in the computing field [518]. Ultimately, that led to the framework illustrated in Fig. 1.21.

Clearly, at least two full semester long courses could be developed using this framework. The one on the left might fit best in a library school, while that on the right might fit in a computing program, but either type of academic program can make use of modules on any topics that are of interest. The middle portion of the figure identifies topics that are *core* for digital libraries, while the bottom portion covers related topics, that can be included in DL classes, or might be covered in other courses or studies.

Thanks to many faculty, researchers, and students, a diverse set of modules has been developed; see Table ???. Though almost all have been reviewed, some still need (more) field testing and refinement. Though most have been developed by instructors, some have been prepared by students, generally in graduate programs, but some undergraduate students have contributed through work on team term projects. In the lower part of the table, modules are associated with a software system, in which case the Core Topic is called Software. Development of new modules, and refinement of existing modules, are ongoing efforts—please see the online sites for additions and enhanced versions.

# DL Curriculum Framework

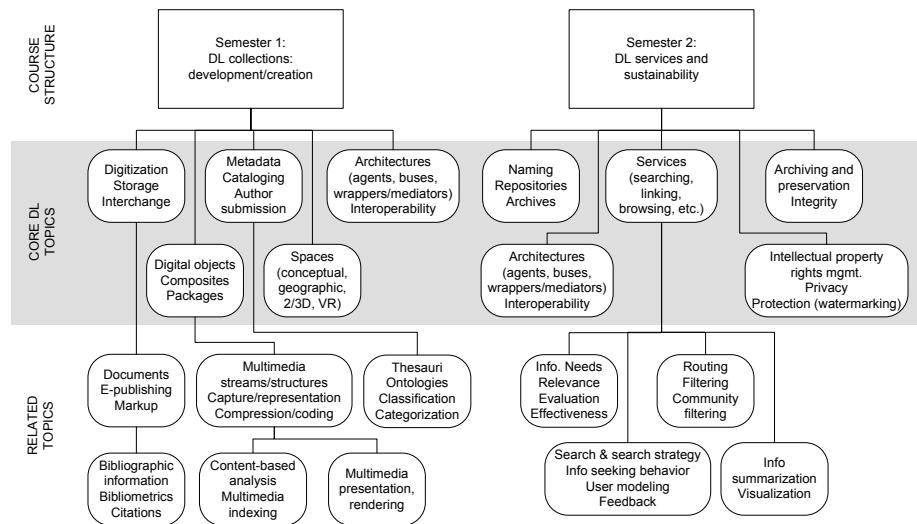


Figure 1.21: DL Curriculum Framework.

## 26 1. INTRODUCTION

**Table 1.1:** List of modules

Core Topic	Module	Author
1. Overview	1-a (10-c): Conceptual frameworks, models, theories, definitions 1-b: History of DLs and library automation	Faculty
2. Digital Objects	2-c (8-d): File formats, transformation, migration	Faculty
3. Collection Development	3-b: Digitization 3-e (7-e): Web Publishing 3-f (7-f): Crawling 4-b: Metadata	Faculty
4. Info/Knowledge Organization		Faculty
5. Architecture	5-a: Architecture overview 5-b: Application software 5-d: Protocols	Faculty
6. User Behavior/Interactions	6-a: Info needs, relevance 6-b: Online info seeking behavior, search strategy 6-d: Interaction design, usability assessment	Faculty
7. Services	7-a: Indexing and searching 7-a(1): Image Retrieval 7-b: Reference services 7-c: Recommender systems 7-d: Routing 7-e (3-e): Web Publishing 7-f (3-f): Crawling 7-g: Personalization	Faculty
8: Preservation	8-a: Preservation 8-b: Web archiving 8-d (2-c): File formats, transformation, migration	Faculty
9: Management and Evaluation	9-c: Evaluation and user studies	Faculty
10: DL Education and Research	10-c (1-a): Conceptual frameworks, models, theories, definitions	Faculty
<b>Software</b>	Apache Solr (enterprise search platform) WordNet (lexical database) NLTK (NLP and text analysis) CLUTO (clustering high-dimensional datasets) TREK Eval (for information retrieval systems) Lemur (language modeling and IR) R (language for statistical analysis) SEDNA XML Database (efficient XML retrieval) Weka (data mining in Java) Hadoop Map-Reduce (parallel processing of data) Media Computation (image manipulation) PureData (sound manipulation) Audacity (recording and editing sounds) Fingerprint (NIST biometric images software)	Grad Grad Grad Grad Grad Grad Grad Grad Grad Grad Grad Ugrad Ugrad Ugrad Ugrad Ugrad

## 1.5 HIGH LEVEL CONSTRUCTS

Digital libraries, viewed from a minimalist perspective [103], are characterized by a small set of essential constructs. Earlier in this chapter, some of those have been discussed, such as *digital object* and *metadata*. Also important are a number of higher level concepts, discussed in this sub-section.

First, there are *collections*. A library is not very interesting if it only has one object of content. When a digital library has a set of content objects, those digital objects constitute a collection. Accordingly, those who are responsible for building the collection are involved in the important work of collection development.

Second, to help manage the collection, there is a *catalog*. Catalogs contain metadata records, each describing a digital object in the collection. Likewise, every digital object in the collection should have a metadata record in the catalog. Often, then, there is a one-to-one relationship between entries in the collection and entries in the catalog. However, in some cases, when there are several metadata formats approved for the catalog, for example the Dublin Core [665, 666] and MARC [483], there can be a metadata record for a given digital object, in each of several metadata formats, .

Third, there are *repositories*. There are multiple connotations to this term. One arose from a seminal article about digital object services [327]:

“A repository is a network-accessible storage system in which digital objects may be stored for possible subsequent access or retrieval. The repository has mechanisms for adding new digital objects to its collection (depositing) and for making them available (accessing), using, at a minimum, the repository access protocol. The repository may contain other related information, services and management systems.”

An important result of this perspective is the system to manage repositories called Fedora [506, 608, 609, 142], which today is quite popular with digital library developers. Thus, a repository is a key software component in a digital library, managing its collection of digital objects. *Repository* also refers to a digital library having some special type and/or collection of content. Thus, there are repositories for video [238], multimedia [112], mathematical software [65], and educational resources [232]. Another connotation of repository is as shorthand for *institutional repository* (IR) [317, 404]. An IR manages a range of digital objects for an institution, such as a department or university. Accordingly, particular digital library systems that are used to manage an institutional repository also are referred to as repository systems. Popular examples include ePrints [279], DSpace [625], and BEPress [52]. Finally, research in digital libraries that focuses on the repository portion sometimes leads to special phrases like *open repository* [106] or *secure repository* [363].

Fourth, there are *archives*. One connotation is as in Open Archives, illustrated earlier in Fig. 1.10. Another connotation is similar to repository, i.e., a digital library having

## 28 1. INTRODUCTION

some special type or collection of content, e.g., arXiv, supporting especially physics and computing [241]. Among the largest of these is the Internet Archive [306], aiming to preserve all Internet content. Also very large are national archives, as in the USA [645], hybrid in that they have both digital and non-digital content. These build upon a long tradition of archival science, practiced by archivists, similar to librarians, but with more emphasis on preservation and somewhat less on access [632]. In the digital age, new models are needed for archival information systems, e.g., OAIS [109]. But much research is needed [287], as well as government policy [179].

Fifth, there are *services*. Often we hear of the service sector of the economy, as distinct from manufacturing, for example. Likewise, libraries are known for providing a variety of services, such as reference services [504]. In the Information Age, services are what computing systems provide. Thus, in client-server settings [434], servers perform services. In the context of the World Wide Web, there are web services [137], usually supporting distributed systems.

Finally, there is the concept of *federation*. For administrative, legal, political, or practical purposes, a set of services can be federated, so they act as a whole. For example, a client might want to search across a variety of online catalogs, using the Z39.50 standard [470, 23, 403] or its Web oriented successor, available as SRU and SRW [481]. Such federated search often involves mediators [169], allowing a heterogeneous set of remote services to be accessed from a single system [256, 230]. However, since some remote systems may not respond in a timely fashion, and since search ranking techniques vary widely, there may be effectiveness problems [223], especially if only some remote sites are selected for search [222]. Accordingly, many services shy away from federation, and, if there are not administrative or legal concerns, use harvesting, wherein a central system collects information, thus having greater control and higher performance [79, 78, 661].

## 1.6 DIGITAL LIBRARY SYSTEMS

The high level constructs described above lead to packaging into digital library systems. According to the Digital Library Manifesto [99] and the DELOS Reference Model [97], there are three levels at issue:

- DL: an organization providing content and services for a community;
- DLS: a software system that affords the functionality of the DL; and
- DLMS: a generic software system that can be instantiated on particular hardware and operated as a DLS.

There are many DLMSs available today, some discussed briefly before. Here we consider three, arising in distant locations around the globe.

## 1.6. DIGITAL LIBRARY SYSTEMS 29

Greenstone was developed in New Zealand [683, 676, 679, 681, 678]. It evolved from the MG information retrieval system [684]. Thanks to a related textbook [677] and open source software [263] that has been enhanced for over a decade, there are many users, and numerous digital librarians have learned by way of this system. Since it runs on a variety of platforms, including small ones, and has been disseminated in connection with UNESCO efforts, it has wide use for small collections in disparate locations.

In 2005, a bridge was developed between Greenstone and DSpace [682]. DSpace also is open source software [162], but was developed, with support from HP, at MIT, in Cambridge, MA, USA [624, 625, 623]. DSpace has played a key role in the emergence of institutional repository systems, since it is relatively easy to install and operate, especially to help manage the digital library needs of a university. With a simple model connecting collections and communities, and with support for uploading both metadata and digital objects, many libraries have found it easy to deploy DSpace. For example, a number of universities installed DSpace to manage their electronic theses and dissertations [288]. Due to its popularity, Doug Gorton's Master's Thesis research targeted DSpace in his work with 5S-based generation of digital libraries from specifications (using 5SGen) [262]—so an install, build, and configuration could proceed in approximately 30 minutes. DSpace has engendered a wide variety of enhancement and complementary software. Recently, there has been a shift toward integration of DSpace with Fedora, through the DuraSpace initiative [168].

Eprints [626] evolved at the University of Southampton out of the CogPrints e-print archive [280], shortly after the Santa Fe Convention that launched the Open Archives Initiative [148]. It has matured greatly, and is now at version 3. Since JISC in the UK has given strong grant support for institutional repositories, and since EPrints is the local favorite, there is a substantial base of use in UK and Europe. More broadly, EPrints is deployed for many e-print and pre-print collections.

Building digital libraries often makes use of these popular digital library management systems, but there are many other supported products available as well. This is an improvement from the situation of ten to fifteen years ago, when many digital libraries were home-grown. At Virginia Tech, we have devised a series of such systems, including CODER [199], MARIAN [94, 694, 217, 256, 255, 258], Open Digital Libraries [615, 616, 614], CITIDEL [211, 302, 206, 513, 328], ETANA [539, 537, 596, 598], and Ensemble [205, 18]. While developing research software is important, often practitioners prefer toolkits [471] or generic systems, when running a production operation.

Given the above background, perspectives, and discussion of various constructs and systems, it is appropriate, to help ensure a deeper understanding of the field, to focus on a formal approach, starting with an intuitive explanation.

## 1.7 5S INTUITION

In traditional libraries, around the world, handling of books operates in a similar fashion, regardless of location, language, or culture. Access to information through such libraries thus is straightforward, and there is a high degree of interoperability. On the other hand, information systems in general, and digital libraries in particular, generally operate in different ways, and are accessible through a diverse set of different interfaces.

For applications to work together to support a range of activities (such as annotating, organizing, indexing, searching, browsing, and visualizing) in scholarly tasks, it is important for them to interoperate with each other. Precise theoretical definitions can help address interoperability problems that arise from ad hoc development and diverse implementation efforts. If applications have a foundation to build upon, there is a better chance of interoperability among similar functioning or complimentary applications.

A formal metamodel can help researchers to develop precise theoretical definitions that address this interoperability problem. A metamodel formally defines the key components that comprise a system. These components, in turn, can be used to define various instances of the system. A model is an abstraction of phenomena in the real world; a metamodel is yet another abstraction, highlighting properties of the model itself. A model conforms to its metamodel in the way that a computer program conforms to the grammar of the programming language in which it is written<sup>2</sup>.

In the remainder of this chapter we introduce and explain the 5S framework, which integrates model and metamodel concepts. We begin with an intuitive explanation, and then proceed with formal definitions in the following section. The key constructs in all this begin with the letter ‘S’, and since they number five, we use the abbreviation ‘5S’.

### 1.7.1 STREAMS

Streams are sequences of elements of an arbitrary type (e.g., bits, characters, images, etc.). In this sense, they can model both static and dynamic content. The first includes, for example, textual material, while the later might be, for example, a presentation of a digital video, or a sequence of time and positional data (e.g., from a GPS) for a moving object.

A dynamic stream can represent an information flow—a sequence of messages encoded by the sender and communicated using a transmission channel possibly distorted with noise, to a receiver whose goal is to reconstruct the sender’s messages and interpret message semantics [594]. Dynamic streams are thus important for representing whatever communications take place in the digital library. Examples of dynamic streams include video-on-demand delivered to a viewer, a timed sequence of news sent to a client, a timed sequence of frames that allows the assembly of a virtual reality scenario, etc. Typically, a dynamic stream is understood through its temporal nature. A dynamic stream then can be

<sup>2</sup>Source: <http://en.wikipedia.org/wiki/Metamodeling>.

interpreted as a finite sequence of clock times and associated values<sup>3</sup> that can be used to define a stream algebra, allowing operations on diverse kinds of multimedia streams [407]. The synchronization of streams can be specified with Petri Nets [479] or other approaches.

In the static interpretation, the temporal nature is generally ignored or is irrelevant, and a stream corresponds to some information content that is interpreted as a sequence of basic elements, often of the same type. A popular type of static stream according to this view is text (sequence of characters). The type of the stream defines its semantics and area of application. For example, any text representation can be seen as a stream of characters, so that text documents, such as scientific articles and books, can be considered as structured streams.

### 1.7.2 STRUCTURES

A structure specifies the way in which parts of a whole are arranged or organized. In digital libraries, structures can represent hypertexts, taxonomies, system connections, user relationships, and containment—to cite a few. Books, for example, can be structured logically into chapters, sections, subsections, and paragraphs; or physically into cover, pages, line groups (paragraphs), and lines [234]. Structuring orients readers within a document’s information.

Markup languages (e.g., SGML, XML, HTML) have been the primary form of exposing the internal structure of digital documents for retrieval and/or presentation purposes [21, 124, 248]. Relational and object-oriented databases impose strict structures on data, typically using tables or graphs as units of structuring [48].

With the increase in heterogeneity of material continually being added to digital libraries, we find that much of this material is called “semistructured” or “unstructured”. These terms refer to data that may have some structure, where the structure is not as rigid, regular, explicit, or complete as the structure used by structured documents or traditional database management systems [3]. Query languages and algorithms can extract structure from these data [360, 4, 466]. Although most of those efforts have a “data-centric” view of semi-structured data, works with a more “document-centric view” have emerged [36, 228, 227]. In general, humans and natural language processing systems can expend considerable effort to unlock the interwoven structures found in texts at syntactic, semantic, pragmatic, and discourse levels.

### 1.7.3 SPACES

A space is a set of objects together with operations on those objects that obey certain constraints. The combination of operations on objects in the set is what distinguishes spaces from streams and structures. Since this combination is such a powerful construct, when a part of a DL cannot be described well using another of the Ss, a space may well

<sup>3</sup>These values are undefined or a value of type  $T$ , e.g., boolean, integer, text, or image.

## 32 1. INTRODUCTION

be applicable. Despite the generality of this definition, spaces are extremely important mathematical constructs. The operations and constraints associated with a space define its properties. For example, in mathematics, affine, linear, metric, and topological spaces define the basis for algebra and analysis [246]. In the context of digital libraries, Licklider discusses spaces for information [390, p. 62]. In the information retrieval discipline, Salton and Lesk formulated an algebraic theory based on vector spaces and implemented it in the SMART system [567]. “Feature spaces” are sometimes used with image and document collections and are suitable for clustering or probabilistic retrieval [545]. Spaces also can be defined by a regular language applied to a collection of documents. Document spaces are a key concept in many digital libraries.

Human understanding can be described using conceptual spaces. Multimedia systems must represent real as well as synthetic spaces in one or several dimensions, limited by some metric or presentational space (windows, views, projections) and transformed to other spaces to facilitate processing (such as compression [147, 700]). Many of the synthetic spaces represented in virtual reality systems try to emulate physical spaces. Digital libraries may model traditional libraries by using virtual reality spaces or environments [45, 140]. Also, spaces for computer-supported cooperative work provide a context for virtual meetings and collaborations [132, 524].

Again, spaces are distinguished by the operations on their elements. Digital libraries can use many types of spaces for indexing, visualizing, and other services they perform. The most prominent of these for digital libraries are measurable spaces, measure spaces, probability spaces, vector spaces, and topological spaces.

### 1.7.4 SCENARIOS

One important type of scenario is a story that describes possible ways to use a system to accomplish some function that a user desires. Scenarios are useful as part of the process of designing information systems. Scenarios can be used to describe external system behavior from the user’s point of view [358]; provide guidelines to build a cost-effective prototype [619]; or help to validate, infer, and support requirements specifications and provide acceptance criteria for testing [295, 620, 368]. Developers can quickly grasp the potentials and complexities of digital libraries through scenarios. Scenarios tell what happens to the streams, in the spaces, and through the structures. Taken together the scenarios describe services, activities, and tasks—and those ultimately specify the functionalities of a digital library.

For example, user scenarios describe one or more users engaged in some meaningful activity with an existing or envisioned system. This approach has been used as a design model for hypermedia applications [485]. Human information needs, and the processes of satisfying them in the context of digital libraries, are well suited to description with scenarios, including these key types: fact-finding, learning, gathering, and exploring [671].

Additionally, scenarios can aid understanding of how digital libraries affect organizations and societies, and how challenges to support social needs relate to underlying assumptions of digital libraries [384]. Scenarios also may help us understand the complexities of current publishing methods, as well as how they may be reshaped in the era of digital libraries, by considering publishing paths, associated participants, and publication functions [669].

The concepts of state and event are fundamental to understanding scenarios. Broadly speaking, a state is determined by what contents are in specified locations, as, for example, in a computer memory, disk storage, visualization, or the real world. The nature of the values and state locations related to contents in a system are granularity-dependent and their formal definitions and interpretations are out of the scope of this chapter; the reader is referred to [674] for a lengthy discussion. An event denotes a transition or change between states, for example, executing a command in a program. Scenarios specify sequences of events, which involve actions that modify states of a computation and influence the occurrence and outcome of future events. Dataflow and workflow in digital libraries can be modeled using scenarios.

### 1.7.5 SOCIETIES

A society is a set of entities and the relationships between them. The entities include humans as well as hardware and software components, which either use or support digital library services. Societal relationships make connections between and among the entities and activities.

Examples of specific human societies in digital libraries include patrons, authors, publishers, editors, maintainers, developers, and the library staff. There are also societies of learners and teachers. In a human society, people have roles, purposes, and relationships. Societies follow certain rules and their members play different roles—participants, managers, leaders, contributors, or users. Members of societies have activities and relationships. During their activities, society members often create information artifacts—art, history, images, data—that can be managed by the library. Societies are holistic—substantially more than the sums of their constituents and the relationships between them. Electronic members of digital library societies, i.e., hardware and software components, are normally engaged in supporting and managing services used by humans.

A society is the highest-level component of a digital library, which exists to serve the information needs of its societies and to describe the contexts of its use. Digital libraries are used for collecting, preserving, and sharing information artifacts between society members. Cognitive models for information retrieval [51, 174, 76], for example, focus on user's information-seeking behavior (i.e., formation, nature, and properties of a user's information need) and on the ways in which information retrieval systems are used in operational environments.

### 34 1. INTRODUCTION

Several societal issues arise when we consider them in the digital library context. These include policies for information use, reuse, privacy, ownership, intellectual property rights, access management, security, etc. [541]. Therefore, societal governance (law and its enforcement) is a fundamental concern in digital libraries. Language barriers are also an essential concern in information systems, and internationalization of online materials is an important issue in digital libraries, given their globally distributed nature [478].

Economics, a critical societal concern, is also key for digital libraries [326]. Collections that were “born electronic” are cheaper to house and maintain, while scanning paper documents to be used online can be relatively expensive. Internet access is widely available and in many settings is inexpensive. Online materials are seeing more use, including from distant locations. Since distribution costs of electronic materials are very low, digital delivery makes economic sense. However, it brings the problem of long-term storage and preservation, which must be adequately addressed, if the information being produced today is to be accessible to future generations [397, 398].

## 1.8 FORMALIZATION OF SS

“DL development must move from an art to a science [and it needs] unifying and comprehensive theories and frameworks across the lifecycle of digital library (DL) information.” [308] (p. 266)

In this section, we precisely and unambiguously formalize most of the informal digital library concepts introduced in previous sections. Figure 1.22 shows a map of the most important concepts and formal definitions. Each concept is associated with the corresponding definition number of its formal definition; arrows indicate that a concept is formally defined in terms of previously defined concepts that point to it<sup>4</sup>. The mathematical preliminaries (Defs. A1–A14) are found in Appendix A.

### 1.8.1 5S FORMALISMS

**Definition 1** A *stream* is a sequence whose codomain is a nonempty set.

**Definition 2** A *structure* is a tuple  $(G, L, \mathcal{F})$ , where  $G = (V, E)$  is a directed graph with vertex set  $V$  and edge set  $E$ ,  $L$  is a set of label values, and  $\mathcal{F}$  is a labeling function  $\mathcal{F} : (V \cup E) \rightarrow L$ .

As a derivative of this definition, the next one follows.

**Definition 3** A *substructure* of a structure  $(G, L, \mathcal{F})$  is another structure  $(G', L', \mathcal{F}')$  where  $G' = (V', E')$  is a subgraph of  $G$ ,  $L' \subseteq L$  and  $\mathcal{F}' : (V' \cup E') \rightarrow L'$ .

<sup>4</sup>The notion of a tuple (def. A.4) is used in most definitions, so, for simplicity, we are not showing arrows coming out of that concept in Figure 1.22. Other popular definitions are treated likewise.

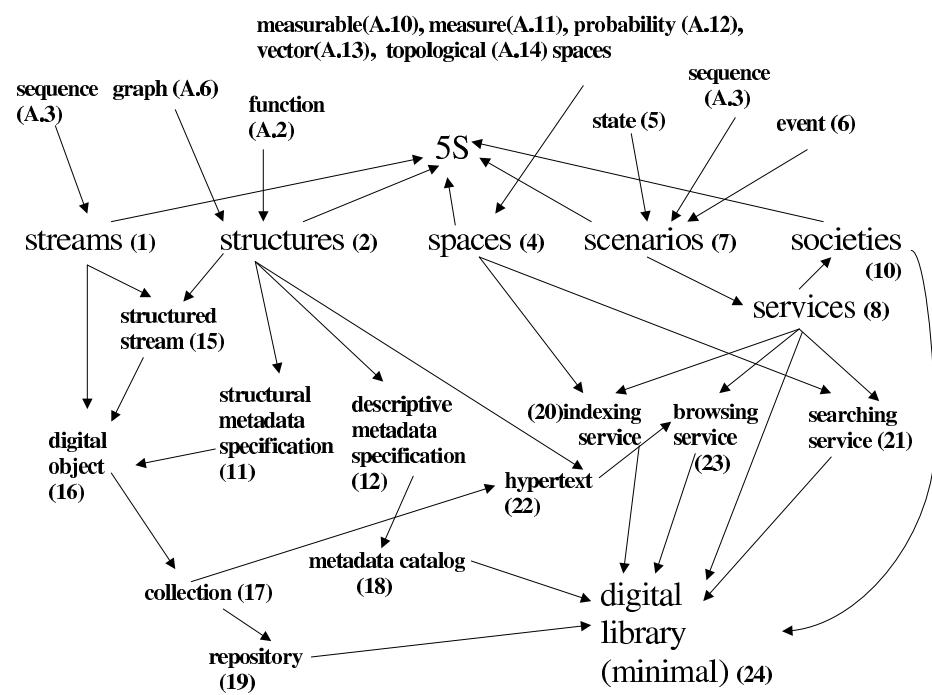


Figure 1.22: 5S map of formal definitions

### 36 1. INTRODUCTION

**Definition 4** A *space* is a measurable space, measure space, probability space, vector space, topological space, or a metric space<sup>5</sup>.

**Probability** studies the possible outcomes of given events (or experiments) together with their relative likelihood and distributions. Probability is defined in terms of a **sample space**  $S$ , which is a set whose elements are called **elementary events**. More formally, in terms of a probability space, the set of possible events for an experiment consists of the  $\sigma$ -algebra  $\mathbb{B}$  and a sample space is defined as the largest set  $S \in \mathbb{B}$ . The measure  $\mu$  is called a probability distribution.

**Probabilistic information retrieval** (PIR) takes a more subjective interpretation of probability, called the *bayesian* interpretation, which sees probability as a statistical procedure which endeavors to estimate parameters of an underlying probability distribution based on the observed distribution. In PIR the sample space is the set  $Q \times D$  of all possible queries and documents and the probability distribution tries to estimate, given a query  $q \in Q$  the probability that a document  $d \in D$  will be **relevant** to the query, using any evidence at hand. Normally the words in the documents and in the query are the major sources of evidence. A precise definition of probability of relevance is dependent on the definition of relevance and different PIR models have different interpretations [134].

Vector spaces are the basis for a widely used information retrieval model, the Vector Space Model (VSM) [568]. In this model, a document space  $D$  is a vector space where a document  $d_i \in D$  is represented by a  $t$ -dimensional vector  $d_i = (w_{i1}, w_{i2}, \dots, w_{it})$ ,  $w_{ij}$  being the weight (a numerical value) of the  $j$ th index term  $t_j$  of  $d_i$ ,  $w_{ij} \geq 0$ . An *index term* is normally a word (or variant), occurring in the text of the document, whose semantics helps in defining the document's main themes. However, in general, an index term may be any value describing some aspect of the document, such as a feature value (e.g., color, shape, elevation, temperature) or descriptor (e.g., element in a thesaurus or classification system), or concept, or complex linguistic expression (e.g., phrase, entry in a gazetteer). Furthermore, it is possible to use their representation vectors, i.e., their terms and term weights, to define a number of functions such as *degree of similarity*  $s : D \times D \rightarrow \mathbb{R}$  between documents.

Vector spaces and measure spaces are often built on top of topological spaces, the latter being the more basic concept. Any use of the concept of distance implies an underlying **metric space**, which is a topological space whose open sets are defined by  $\{y \mid d(x, y) < r\}$ , where  $d(x, y)$  is the distance between  $x$  and  $y$ .

**Definition 5** A *system state* (from now on, just *state*) is a function  $s : L \rightarrow V$ , from labels  $L$  to values  $V$ . A *state set*  $S$  consists of a set of state functions  $s : L \rightarrow V$ .

Labels represent a logical *location* associated with some value in a particular state. Thus  $s_i(X)$  is the value, or the contents, of location  $X$  in state  $s_i \in S$ . The nature of the

<sup>5</sup>See Appendix definitions 9-14 for formal definitions of each of these spaces.

values related to contents in a system is granularity-dependent and its definition is out of the scope of this chapter. Normally there are simple values of basic datatypes such as strings and numbers or higher-level DL objects such as digital objects and metadata specifications.

**Definition 6** A *transition event* (or simply *event*) on a state set  $S$  is an element  $e = (s_i, s_j) \in (S \times S)$  of a binary relation on state set  $S$  that signifies the transition from one state to another. An event  $e$  is defined by a condition function  $c(s_i)$  which evaluates a Boolean function in state  $s_i$  and by an action function  $p$ .

This transition event is not a *probabilistic* event [125]. Rather, it is more like the events in networked operating systems theory [601], transitions in finite state machines [144], those modeled by the Unified Modeling Language (UML) [67], or transitions between places in Petri Nets [479].

The condition is used to describe circumstances under which a state transition can take place. An action models a reference to an operation, command, subprogram or method, responsible to perform the actual state transition. Events and actions can have parameters that abstract data items associated with attributes (labels) of a state.

**Definition 7** A *scenario* is a sequence of related transition events  $\langle e_1, e_2, \dots, e_n \rangle$  on state set  $S$  such that  $e_k = (s_k, s_{k+1})$ , for  $1 \leq k \leq n$ .

We also can interpret a scenario as a path in a directed graph  $G = (S, \Sigma_e)$ , where vertices correspond to states in the state set  $S$  and directed edges are equivalent to events in a set of events  $\Sigma_e$  (and correspond to transitions between states). (Technically,  $G$  is a pseudodigraph<sup>6</sup>, since loops  $(s_i, s_i)$  are possible as events.)

**Definition 8** A *service, activity, task, or procedure* is a set of scenarios.

Note that the scenarios defining a service can have shared states. Such a set of related scenarios has been called a “scenario view” [295] and a “use case” in the UML [67]. In this framework, a simple transmission service of streams can be formally specified as:

**Definition 9** Let  $T = \langle t_1, t_2, \dots, t_n \rangle$  be a stream. Let event  $e_{t_i} = (s_{t_i}, d_{t_i}$ <sup>7</sup>) and event  $a_{t_i} = (d_{t_i}, s_{t_{i+1}})$ . A transmission of stream  $T$  is the scenario (sequence of related events)  $e_T = \langle e_{t_1}, a_{t_1}, e_{t_2}, a_{t_2}, \dots, e_{t_n} \rangle$ .

Scenarios are *implemented* to make a working system and the so-called “specification-implementation” gap must be overcome [550]. Formally, the implementation of scenarios can be mapped to an abstract machine represented by a deterministic finite automaton (DFA). This automaton  $M = (Q, \Sigma_e, \delta, q_0, F)$  is such that  $M$  is the user-perceived conceptual state machine of the system and accepts a language  $L(M)$  over the set of events  $\Sigma_e$ .

<sup>6</sup>A digraph which permits both loops and multiple edges between nodes.

<sup>7</sup> $d_{t_i}$  is the state that indicates that the destination has received stream item  $t_i$

### 38 1. INTRODUCTION

A grammar  $G = (V, \Sigma_e, R, s_0)$  for the language  $L(M)$  is such that the non-terminals set  $V$  corresponds to the state set  $S$ , the terminals are the finite set of events  $\Sigma_e$ ,  $s_0$  is a distinguished initial state initializing all locations in that state, and  $R$  is a finite set of rules. Each rule in  $R$  is of the form  $s_i \rightarrow es_j$  and conveys the system from state  $s_i$  to  $s_j$  as a consequence of event  $e$ , or is of the form  $s_i \rightarrow e$  when  $s_j \in F$  is a final state. The grammar and the corresponding conceptual state machine make up the abstract formal model which the analyst uses to capture, represent, and display system behavior in terms of scenarios. Alternatively, denotational semantics [674] and object-oriented abstractions [549] offer a programming language perspective for the question of formal scenario implementation.

**Definition 10** A *society* is a tuple  $(C, R)$ , where

1.  $C = \{c_1, c_2, \dots, c_n\}$  is a set of conceptual communities, each community referring to a set of individuals of the same class or type (e.g., actors, service managers);
2.  $R = \{r_1, r_2, \dots, r_m\}$  is a set of relationships, each relationship being a tuple  $r_j = (e_j, i_j)$ , where  $e_j$  is a Cartesian product  $c_{k_1} \times c_{k_2} \times \dots \times c_{k_{n_j}}$ ,  $1 \leq k_1 < k_2 < \dots < k_{n_j} \leq n$ , which specifies the communities involved in the relationship and  $i_j$  is an activity (cf. Def. 8) that describes the interactions or communications among individuals.

The second part of the definition emphasizes the collaborative nature of societies as in the case of users and service managers engaged in performing DL services. Scenarios describe the service behavior exactly in terms of interactions among the involved societies. For example, an ETD submission service involves interactions between graduate students and an ETD submission workflow manager (an electronic member of a service managers society).

## 1.9 FORMALIZATION OF MINIMAL DIGITAL LIBRARY

As pointed out in previous sections, there is no consensual definition of a digital library. This makes the task of formally defining this kind of application and its components extremely difficult. In this section, we approach this problem by constructively defining a “core” or a “minimal” digital library, i.e., the minimal set of components that make a digital library, without which, in our view, a system/application cannot be considered a digital library. Each component (e.g., collections, services) is formally defined in terms of an S construct or as combinations or compositions of two or more of them. The set-oriented and functional mathematical formal basis of 5S allows us to precisely define those components as functional compositions or set-based combinations of the formal Ss.

Informally, a digital library involves a managed *collection* of information with associated *services* involving *communities* where information is stored in digital formats and accessible over a network. Information in digital libraries is manifest in terms of *digital*

### 1.9. FORMALIZATION OF MINIMAL DIGITAL LIBRARY 39

*objects*, which can contain textual or multimedia content (e.g., images, audio, video), and *metadata*. Although the distinction between data and metadata often depends on the context, metadata commonly appears in a structured way and covering different categories of information *about* a digital object. The most common kind of metadata is *descriptive metadata*, which occurs in catalogs and indexes and includes summary information used to describe objects in a DL. Another common characteristic of digital objects and metadata is the presence of some internal structure, which can be explicitly represented and explored to provide better DL services. Basic services provided by digital libraries are indexing, searching, and browsing. Those services can be tailored to different communities depending on their roles, for example, creators of material, librarians, patrons, etc.

In the following we formally define those concepts of *metadata (structural and descriptive)*, *digital object*, *collection*, *catalog*, *repository*, *indexing service*, *searching service*, *browsing service*, and finally *digital library*.

**Definition 11** *A structural metadata specification is a structure.*

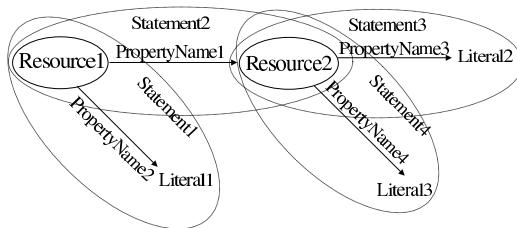
This simple definition emphasizes the role of structural metadata as a representation or abstraction of relationships between digital objects and their component parts (cf. Def. 16). The graph-based representation of this type of metadata can be explicitly expressed, as in the case of markup [124], or implicitly computed [458, 121].

The next definition, for **descriptive metadata specifications**, is inspired by developments in the metadata area, mainly those related to the *Semantic Web* [54] and the Resource Description Framework (RDF) [654], and emphasizes the semantic relationships implied by the labeling function in a structure. Figure 2.3(a) illustrates the basic constructs. Statements, which are triples corresponding to a specific resource (the thing being described) together with a named property about the resource plus the value of that property for that resource, are promoted to first-class concepts. Figure 1.23(b) shows an example of an instantiation of the construct for a descriptive metadata specification about an electronic thesis with four statements: Statement1 = (Thesis1, ‘author’, ‘M.A.Goncalves’), Statement2 = (Thesis1, ‘degree’, Degree1), Statement3 = (Degree1, ‘level’, ‘doctoral’), and Statement4 = (Degree1, ‘grantor’, ‘Virginia Tech’). Below we define the notions of **descriptive metadata specification** and **metadata format** more formally.

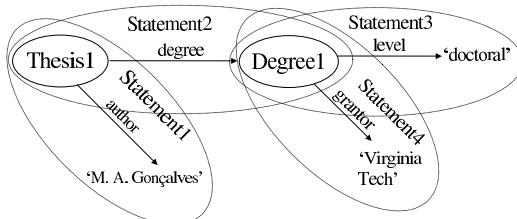
**Definition 12** *Let  $\mathcal{L} = \bigcup D_k$  be a set of literals defined as the union of domains  $D_k$  of simple datatypes (e.g., strings, numbers, dates, etc.). Let also  $\mathcal{R}$  and  $\mathcal{P}$  represent sets of labels for resources and properties respectively. A descriptive metadata specification is a structure  $(G, \mathcal{R} \cup \mathcal{L} \cup \mathcal{P}, \mathcal{F})$ , where:*

1.  $\mathcal{F} : (V \cup E) \rightarrow (\mathcal{R} \cup \mathcal{L} \cup \mathcal{P})$  can assign general labels  $\mathcal{R} \cup \mathcal{P}$  and literals from  $\mathcal{L}$  to nodes of the graph structure;
2. for each directed edge  $e = (v_i, v_j)$  of  $G$ ,  $\mathcal{F}(v_i) \in \mathcal{R} \cup \mathcal{L}$ ,  $\mathcal{F}(v_j) \in \mathcal{R} \cup \mathcal{L}$  and  $\mathcal{F}(e) \in \mathcal{P}$ ;

40 1. INTRODUCTION



(a)



(b)

Figure 1.23: Overview of descriptive metadata with example

3.  $\mathcal{F}(v_k) \in \mathcal{L}$  if and only if node  $v_k$  has outdegree 0.

The triple  $st = (\mathcal{F}(v_i), \mathcal{F}(e), \mathcal{F}(v_j))$  is called a **statement** (derived from the descriptive metadata specification), meaning that the resource labeled  $\mathcal{F}(v_i)$  has property  $\mathcal{F}(e)$  with value  $\mathcal{F}(v_j)$  (which can be designated as another resource or literal).

**Definition 13** Let  $D_{\mathcal{L}_{MF}} = \{D_1, D_2, \dots, D_i\}$  be the set of domains that make up a set of literals  $\mathcal{L}_{MF} = \bigcup_{j=1}^i D_j$ . As for metadata specifications, let  $\mathcal{R}_{MF}$  and  $\mathcal{P}_{MF}$  represent sets of labels for resources and properties, respectively. A **metadata format** for descriptive metadata specifications is a tuple  $MF = (V_{MF}, \text{def}_{MF})$  with  $V_{MF} = \{\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_k\} \subset 2^{\mathcal{R}_{MF}}$  a family of subsets of the resource labels  $\mathcal{R}_{MF}$  and  $\text{def}_{MF} : V_{MF} \times \mathcal{P}_{MF} \rightarrow V_{MF} \cup D_{\mathcal{L}_{MF}}$  is a property definition function.

Therefore a metadata format, through the property definition function, constrains the kinds of resources that can be associated together in statements of a metadata specification as well as the basic datatype domains, which are associated with pairs (resource-property) related to literals [107]. For example, for any set of labels  $\mathcal{R}$  for resources, the Dublin Core metadata format defines that  $\text{def}_{DC}(\mathcal{R}, \text{'title'}) = \text{String}$  and  $\text{def}_{DC}(\mathcal{R}, \text{'subject'}) = \text{SubjectTerms}$  where *SubjectTerms* is a finite set of labels for Resources corresponding to controlled terms. The following definition follows from the previous two definitions:

**Definition 14** A descriptive metadata specification  $MS = (G_{MS}, \mathcal{R}_{MS} \cup \mathcal{L}_{MS} \cup \mathcal{P}_{MS}, \mathcal{F}_{MS})$  **conforms with** a metadata format  $MF = (V_{MF}, \text{def}_{MF})$  if  $\mathcal{R}_{MS} \subseteq \mathcal{R}_{MF}$ ,  $\mathcal{L}_{MS} \subseteq \mathcal{L}_{MF}$ ,  $\mathcal{P}_{MS} \subseteq \mathcal{P}_{MF}$ , and for every statement  $st = (r, p, l)$  derived from  $MS$ ,  $r \in \mathcal{R}_k$  for some  $\mathcal{R}_k \in V_{MF}$  and  $p \in \mathcal{P}_{MS}$  implies  $l \in \text{def}_{MF}(\mathcal{R}_k, p)$ .

**Definition 15** Given a structure  $(G, L, \mathcal{F})$ ,  $G = (V, E)$  and a stream  $S$ , a **Structured-Stream** is a function  $V \rightarrow (\mathbb{N} \times \mathbb{N})$  that associates each node  $v_k \in V$  with a pair of natural numbers  $(a, b)$ ,  $a < b$ , corresponding to a contiguous subsequence  $[S_a, S_b]$  (segment) of the stream  $S$ .

Therefore, a StructuredStream defines a mapping from nodes of a structure to segments of a stream. An example in a textual stream can be seen in Figure 1.24 . From the example, it can be deduced that several structures can be imposed over one stream and vice-versa. Also, it can be seen that segments associated with a node should include the segments of its children (in the case of a hierarchical tree), although it is not equal to the union of those, as “gaps” or “holes” can occur between child segments [458]. Finally, it should be noted that this definition works also for multimedia streams like audio, video, and images.

**Definition 16** A **digital object** is a tuple  $do = (h, SM, ST, \text{StructuredStreams})$  where

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

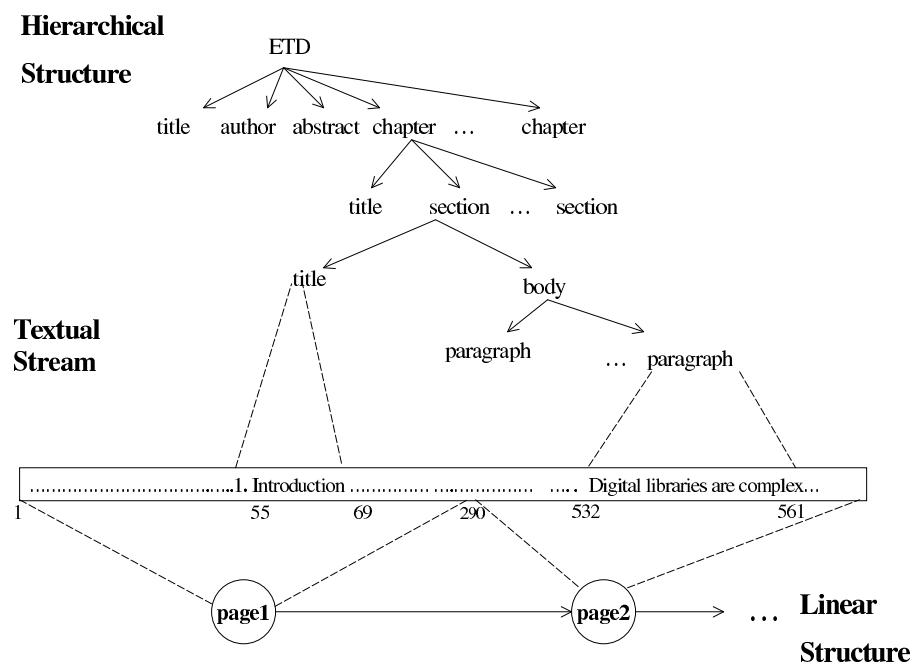


Figure 1.24: A StructuredStream for an ETD (adapted from [458])

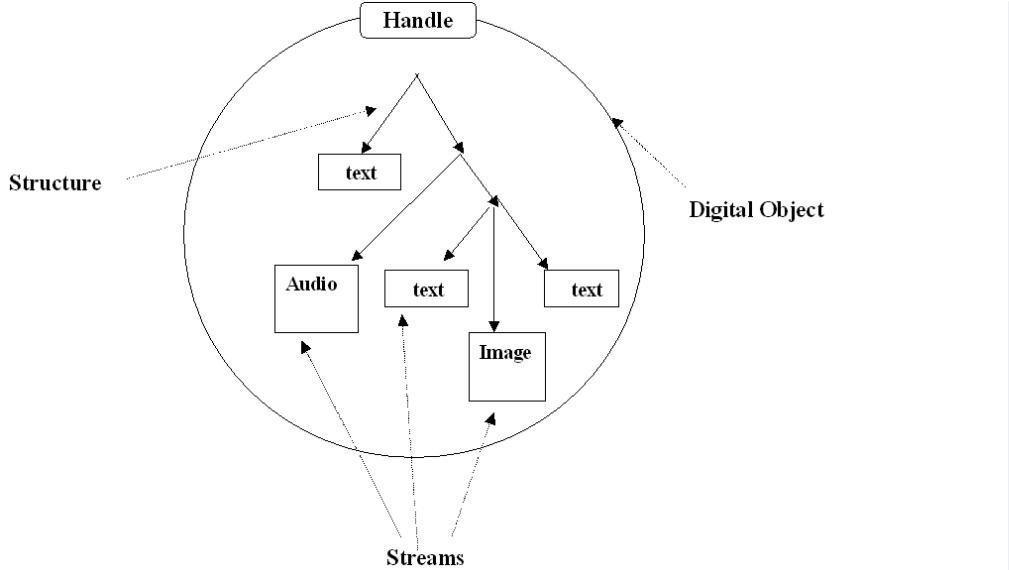


Figure 1.25: A simple digital object

2.  $SM = \{sm_1, sm_2, \dots, sm_n\}$  is a set of streams;
3.  $ST = \{st_1, st_2, \dots, st_m\}$  is a set of structural metadata specifications;
4.  $StructuredStreams = \{stsm_1, stsm_2, \dots, stsm_p\}$  is a set of StructuredStream functions defined from the streams in the  $SM$  set (the second component) of the digital object and from the structures in the  $ST$  set (the third component).

Figure 1.25 shows an example of a very simple digital object with one structure and several streams. Two important aspects must be pointed out about this formal definition of a digital object:

1. Any real implementation does not need to enforce physical containment of the several component parts of a digital object; for example, we could have pointers to external streams.
2. The definition does not consider active behavior of digital objects [609, 462] which supports operations like different disseminations or exporting of subparts. While there is no explicit restriction regarding this, the definition conforms to our minimalist approach.

**Definition 17** A **collection**  $C = \{do_1, do_2, \dots, do_k\}$  is a set of digital objects.

## 44 1. INTRODUCTION

**Definition 18** Let  $C$  be a collection with  $k$  handles in  $H$ . A **metadata catalog**  $DM_C$  for  $C$  is a set of pairs  $\{(h, \{dm_1, \dots, dm_{k_h}\})\}$ , where  $h \in H$  and the  $dm_i$  are descriptive metadata specifications.

**Definition 19** Let  $C$  be a collection with handles  $H$ . A **repository** is a tuple  $(R, get, store, del)$ , where  $R \subset 2^C$  is a family of collections and the functions “get”, “store,” and “del” satisfy:

1.  $get : H \rightarrow C$  maps a handle  $h$  to a digital object  $get(h)$ .
2.  $store : C \times R \rightarrow R$  maps  $(do, \tilde{C})$  to the augmented collection  $\{do\} \cup \tilde{C}$ .
3.  $del : H \times R \rightarrow R$  maps  $(h, \tilde{C})$  to the smaller collection  $\tilde{C} - \{get(h)\}$ .

Thus a repository encapsulates a set of collections and specific services to manage and access the collections.

**Definition 20** Let  $I : 2^{\mathcal{T}} \rightarrow 2^H$  be an index function where  $\mathcal{T}$  is a set of indexing features and  $H$  is a set of handles. An **index** is a set of index functions. An **indexing service** is a single scenario  $\{\langle is_1, is_2, \dots, is_n \rangle\}$  comprised of pipelined scenarios  $is_1, is_2, \dots, is_n$  in which the starting state  $s_{k_0}$  of the first event of the initial scenario  $is_1$  has a collection  $s_{k_0}(K) = C$  and/or a metadata catalog  $s_{k_0}(Y) = DM_C$  for collection  $C$  as its values and the final state  $s_{k_f}$  of the final scenario  $is_n$  has an index  $I_C = s_{k_f}(Z)$  as its value ( $K$ ,  $Y$ , and  $Z$  being labels of the respective states).

The interpretation of the index and the indexing service is dependent upon the underlying indexing space. Features of an indexing space can be words, phrases, concepts, or multimedia characteristics, like shape or color, appearing or associated with the content of a digital object (in its descriptive and structural metadata or streams). Normally, if a vector space is considered, terms are treated as unrelated, therefore defining orthogonal vectors that span a space  $\mathcal{T}$  with dimension  $m$ . If a probabilistic space  $p = (X, \mathbb{B}, \mu)$  is used,  $\mathcal{T} = X$  is the set of distinct terms and is called a *sample space*. Also an index can be thought of as a mapping from an indexing space to a *document (digital object) space* defined by the collection.

The indexing service normally takes the shape of a *pipeline service* where scenarios themselves are executed in sequence and the final state of a scenario is the starting state of the next one. A very simple instance of such an indexing service is shown in Figure 1.26 for indexing of textual material. The indexing service is composed of three scenarios organized as a pipeline of the following scenarios: 1) tokenization, which identifies unique terms inside the textual streams; 2) stopword removal, which filters out terms not useful for retrieval; and 3) stemming, which removes affixes and allows retrieval of syntactic variations of query terms [37]. Each one of the scenarios can be thought of as doing some transformation in the representations of digital objects in order to produce the index function. Note again

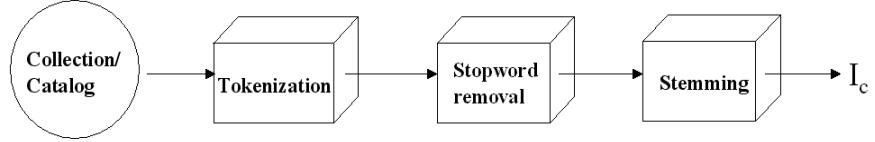


Figure 1.26: Simple indexing service

that we are making use of our minimalist approach by not considering complex indexes, for example, defining locations inside streams of digital objects for phrase, proximity, or structural queries.

**Definition 21** Let  $Q$  be a set of conceptual representations for user information needs, collectively called *queries*. Let  $M_{I_C} : Q \times (C \times DM_C) \rightarrow \mathbb{R}$  be a matching function, associated with an index  $I_C$ , that associates a real number with a query  $q \in Q$  and a digital object  $do \in C$  and possibly its descriptive metadata specifications  $ms \in DM_C$ , indicating how well the query representation matches with the digital object, structurally, by content, or regarding the descriptive metadata specifications. A **searching service** is a set of searching scenarios  $\{sc_1, sc_2, \dots, sc_t\}$ , where for each query  $q \in Q$  there is a searching scenario  $sc_k = \langle e_0, \dots, e_n \rangle$  such that  $e_0$  is the start event triggered by a query  $q$  and event  $e_n$  is the final event of returning the matching function values  $M_I(q, d)$  for all  $d \in C$ .

The components of a digital object  $do$ , are denoted by  $do(1)$ ,  $do(2)$ , etc. Therefore,  $do_k(2)$  denotes the second component, i.e., the stream set component of a digital object  $do_k$ ,  $do_k(3)$  its structural metadata set component (third component), and  $do_k(4)$  its set of StructuredStreams functions (fourth component). Let also  $G[v]$  denote the subgraph of a directed graph  $G$  containing node  $v$  and all points and edges reachable starting from  $v$ . A substructure defined by  $G[v]$  inherits the labeling of the structure defined with  $G$ . Finally, let  $f : A \rightarrow B$  and let  $\mathcal{D}$  be any non-empty subset of  $A$ . The **restriction** of  $f$  to  $\mathcal{D}$ , denoted by  $f|_{\mathcal{D}}$ , is a subset of  $f$  and is a function from  $\mathcal{D}$  to  $B$ . Then, for a collection  $C$ :

1.  $AllStreams = (\cup_{do_k \in C} do_k(2))$  and  $AllSubStreams = \cup_{sm_t \in AllStreams} \{sm_t[i, j] \mid sm_t = \langle a_0, a_1, \dots, a_n \rangle, 0 \leq i \leq j \leq n \}$  will be the set of all streams and substreams (segments of streams) of all digital objects in the collection  $C$ ;
2.  $AllSubStructuredStreams = \bigcup_{k,j} (SubStructuredStream_{k,j})$  where:
  - (a)  $d_k \in C$ ;
  - (b)  $G_{k,j} = (V_{k,j}, E_{k,j})$  is the first component of some structure  $st_{k,j} \in d_k(3)$ ;
  - (c)  $\mathcal{H}_{k,j} = \{G_{k,j}[v_t] \mid vt \in V_{k,j}\}$  corresponds to the set of all substructures of  $st_{k,j}$ ;

## 46 1. INTRODUCTION

- (d)  $\text{SubStructuredStream}_{k_j} = \{\mathcal{S}|_{V'} \mid (V', E') \in \mathcal{H}_{k_j}, \mathcal{S} \in d_k(4)\}$  is a Structured-Stream function defined from the structure  $st_{k_j}$ , and  $\mathcal{S}|_{V'}$  is the restriction of  $\mathcal{S}$  to  $V'$ .

Therefore,  $AllSubStructuredStreams$  corresponds to the set of all possible substructures and their corresponding connections to streams inside digital objects of the collection.

**Definition 22** Let  $H = ((V_H, E_H), L_H, \mathcal{F}_H)$  be a structure and  $C$  be a collection. A **hypertext**  $HT = (H, \text{Contents}, \mathcal{P})$  is a triple such that:

1.  $\text{Contents} \subseteq C \cup AllSubStreams \cup AllSubStructuredStreams$  is a set of contents that can include digital objects of a collection  $C$ , all of their streams (and substreams) and all possible restrictions of the StructuredStream functions of digital objects.
2.  $\mathcal{P} : V_H \rightarrow \text{Contents}$  is a function which associates a node of the hypertext with the node content.

A hyperlink is an edge in the hypertext graph. Source nodes of a hyperlink are called “anchors” and are generally associated via function  $\mathcal{P}$  with segments of streams. Also, in this definition, two basic types of hyperlinks can be identified: *structural* and *referential* [659]. Structural hyperlinks allow navigation inside internal structures and across streams of digital objects. Referential hyperlinks usually have their target nodes associated with different digital objects or their subcomponents.

Figure 1.27 illustrates the definition. The hypertext is made by structural hyperlinks that follow the structural metadata and external referential links. Links originate from (segments of) streams. Link targets for, respectively, links 1, 2, and 3, are an entire digital object, a portion of its StructuredStream function (in the figure, represented by the subgraph pointed to by the link and the associated streams) and one of its streams, in this case an image.

An example of such a hypertext is the Web. The Web is a structure where hypertext links connect nodes that can be associated with: 1) complete HTML pages that can be considered digital objects; 2) substructures of a HTML page, for example, a section of the page; and 3) links to streams, e.g., images, audios, or text. The Distributed Graph Storage (DGS) system also implements similar ideas with structural and hyper-structural links representing, respectively, the internal structures of digital objects and hypertext constructs [592]. It should be noted that for the sake of brevity we are not describing links to services, for example, external plugins that can be invoked by browsers or Web forms.

**Definition 23** A **browsing service** is a set of scenarios  $\{sc_1, \dots, sc_n\}$  over a hypertext (meaning that events are defined by edges of the hypertext graph  $(V_H, E_H)$ ), such that traverse link events  $e_i$  are associated with a function  $\text{TraverseLink} : V_H \times E_H \rightarrow$

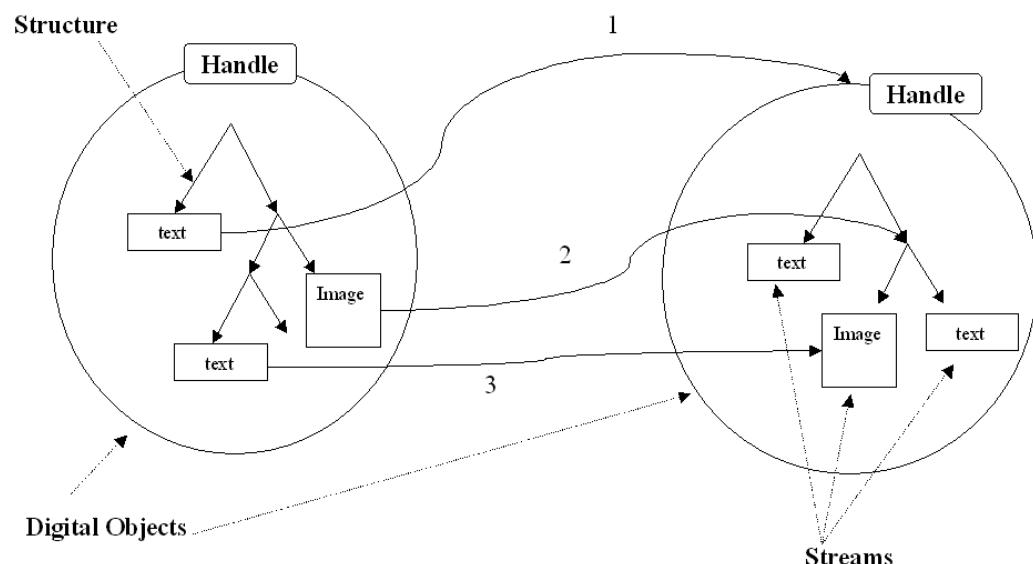


Figure 1.27: A simple hypertext

## 48 1. INTRODUCTION

*Contents*, which given a node and a link retrieves the content of the target node, i.e.,  $\text{TraverseLink}(v_k, e_{k_i}) = \mathcal{P}(v_t)$  for  $e_{k_i} = (v_k, v_t) \in E_H$ .

Therefore, by this definition, every browsing service is associated with an underlying hypertext construct. This view unifies the three modes of browsing defined by Baeza-Yates and Ribeiro-Neto [37]: flat browsing, structured guided, and navigational mode. The third one is the most general case and fits exactly our framework. The first two can be considered special cases. In flat browsing the hypertext has a flat organization, for example, an ordered list of documents or a set of points in an image, and the graph structure of the hypertext corresponds to a disconnected bipartite graph. In the second one, which includes classification hierarchies and directories, the hypertext graph is a tree. Many semi-structured wrapper algorithms disclose this hypertext “hidden” structure in the Web. Once revealed, this structure can be recorded in databases or represented in other semi-structured models to allow queries or transformations. Methodologies like PIPE [534] make use of this information to personalize Web sites. Note also that more sophisticated kinds of hypertext can be defined by extending the current definition. For example, we could relax the function  $\mathcal{P}$  to be a relation and associate different contents with the same node, which could be achieved by having different modes of traversing the same link in an extension of the *TraverseLink* function<sup>8</sup>. However, the present definition is simpler and serves well our minimalist approach<sup>9</sup>.

**Definition 24** A *digital library* is a 4-tuple  $(\mathcal{R}, \text{Cat}, \text{Serv}, \text{Soc})$ , where

- $\mathcal{R}$  is a repository;
- $\text{Cat} = \{DM_{C_1}, DM_{C_2}, \dots, DM_{C_K}\}$  is a set of metadata catalogs for all collections  $\{C_1, C_2, \dots, C_K\}$  in the repository;
- $\text{Serv}$  is a set of services containing at least services for indexing, searching, and browsing;
- $\text{Soc}$  is a society.

We should stress that the above definition (illustrated in Fig. 1.22) only captures the syntax of a digital library, i.e., what a digital library is. Many semantic constraints and consistency rules regarding the relationships among the DL components (e.g., how the scenarios in *Serv* should be built from  $\mathcal{R}$  and *Cat* and from the relationships among communities inside the society *Soc*, or what the consistency rules are among digital objects in collections of  $\mathcal{R}$  and metadata records in *Cat*) are not specified here; please see Chapter 3.

<sup>8</sup>This extended approach also generalizes the notion of link directionality where bi-directional links or non-directional links correspond just to different ways of traversing the link (e.g., SOURCE\_TO\_SINK, SINK\_TO\_SOURCE, BOTH).

<sup>9</sup>Note also that libraries can support *serendipity* or ‘random links’.

## 1.10 DIGITAL LIBRARY TAXONOMY

A taxonomy is a classification system of empirical entities with the goal of classifying cases according to their measured similarity on several variables [44]. Classifications are a premier descriptive tool and as such, they give a foundation towards an explanation for a phenomena. Classifications provide a terminology and vocabulary for a field and help to reduce complexity and achieve parsimony of description by logically arranging concepts through the identification of similarities and differences. We have built a taxonomy for digital libraries as a classification system of terms involved with the field. Our taxonomy describes the digital library field in conceptual terms and therefore its organization is amenable to be interpreted in the light of 5S. This interpretation aims toward a more informal conceptual understanding of the ‘Ss’ and corresponding DL components.

In the process of building such a taxonomy, we have considered the principles of taxonomies in social sciences, notably cluster analysis, and faceted classification schemes [649]. In particular we were guided by the idea that writing about a subject unequivocally reveals the appropriate facets for that subject [194], and that those facets are enough to describe the phenomenon [535]. We followed an agglomerative strategy using subjective relational concepts like association and correlation. During the construction of the taxonomy we tried to accommodate all the terms found in the literature and marginal fields, guarantee mutual exclusivity, and ensure consistency and clarity.

To collect the unstructured list of concepts, we went through the early literature to find all features, issues, and roles utilized, and identified specific terms [253]. As a starting point, we used an initial set of terms and phrases listed alphabetically in [204]. To this list we added other terms from various articles. When this was reasonably voluminous, we produced a grouping of terms of similar or related meaning into “notational families” known as facets. Each group was given a label that described the idea behind the homogeneity of the group or the main variable considered. From there, we grouped the clusters, and so on, until we achieved convergence into one unique facet called “digital library.”

Once the initial taxonomy was complete, we noticed certain terms were missing or ambiguous, so we added terms and qualified them in each context. After several iterations of successive clustering, declustering, and reclustering, we released a more concrete and consistent working set for peer review and then improved the taxonomy based on comments received. The resulting taxonomy is shown in Figure 1.28.

We must point out that, as with any classification system, our taxonomy must evolve to accommodate changes in the digital library field. However, two factors should contribute to the stability of the taxonomy, and therefore to its relative longevity. First the taxonomy was derived from a significant corpus of digital library literature; therefore it is more stable than personal opinions. Second, the higher-level groupings are significantly abstract so that they may be applied to many fields, with possible additions or changes necessary only at the level of specific categories. Clearly, such changes are likely due to the youth and rapid

## 50 1. INTRODUCTION

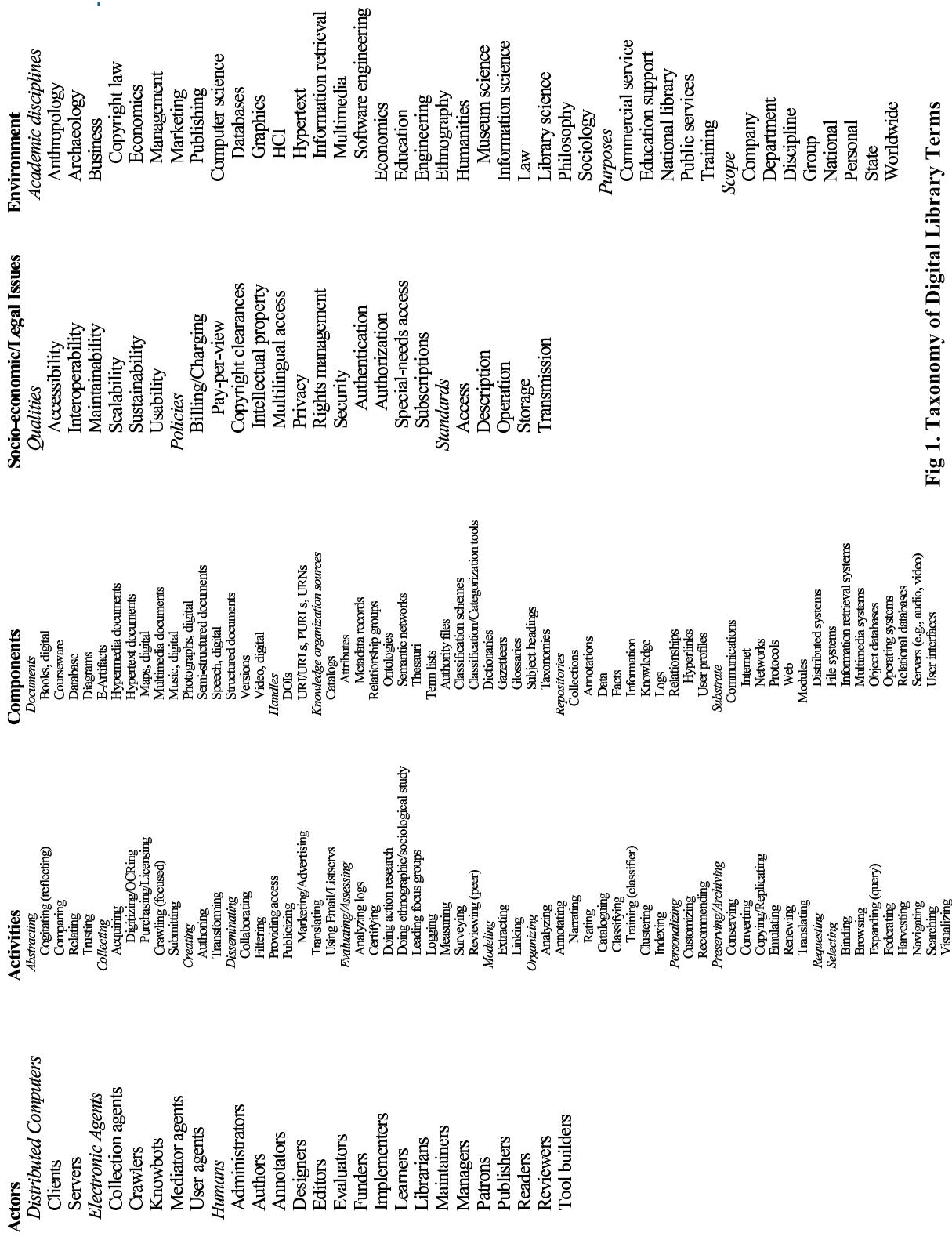


Fig 1. Taxonomy of Digital Library Terms

Figure 1.28: Taxonomy of digital library terms

## 1.10. DIGITAL LIBRARY TAXONOMY 51

development of the field. In the following we describe the main facets and sub-facets of the taxonomy, making use of 5S as an analytical tool.

**Actors: Who interacts with/within DLs?** In our context, actors are the users of a digital library. Actors interact with the DL through an interface design that is (or should be) affected by the actors' preferences and needs. Actors who have preferences and needs in common, display similar behavior in terms of services they use and interactions they practice. We say these actors form a *digital community*, the building blocks of a digital library society<sup>10</sup>. Communities—of students, teachers, librarians—interact with digital libraries and use digital libraries to interact, following pre-specified scenarios. Communities can act as a query-generator service, from the point of view of the library, and as a teaching, learning, and working service, from the point of view of other humans and organizations. Communications between actors and among the same and different communities occur through the exchange of streams. Communities of autonomous agents and computers also play roles in digital libraries. They instantiate scenarios upon requests by the actors of a DL. To operate, they need structures of vocabulary and protocols. They act by sending (possibly structured) streams of queries and retrieving streams of results.

**Activities: What happens in DLs?** Activities of digital libraries — abstracting, collecting, creating, disseminating, evaluating, modeling, organizing, personalizing, preserving, requesting, and selecting — all can be described and implemented using scenarios and occur in the DL setting as a result of actors using services. Furthermore, these activities make and characterize relationships within and between societies, streams, and structures. Each activity happens in a setting, arena, or space. The relationships developed can be seen in the context of larger structures (e.g., social networks [588, 331]).

**Components: What constitutes DLs?** Digital libraries can contain repositories of knowledge, information, data, metadata, relationships, logs, annotations, user profiles, and documents. They can be associated with higher-level structuring and organizational materials: term lists (e.g., authority files, dictionaries), classification tools (e.g., subject headings and taxonomies), thesauri, ontologies, and metadata catalogs. These knowledge organization sources are normally applied to collections of digital objects and support a number of services such as metadata-based resource discovery, query expansion with thesauri, hierarchical browsing with classification systems, and ontology-based crosswalks among disparate metadata formats and vocabularies. Finally, DLs are served by a substrate—a foundational complex amalgamation of different combinations of Ss that involves computers, network

<sup>10</sup>Digital communities are formed by actors who interact with a DL possibly through the same interface paradigm. The actors might belong to distinct social communities of the real world. For instance, a digital community might be instantiated by the adoption of a particular architecture and interface for a DL (e.g., a chat room or MOO). This instantiation is somewhat arbitrary and artificial. Social communities, on the other hand, appear much more naturally as a result of complex social interactions.

## 52 1. INTRODUCTION

connections, file and operating systems, user interfaces, communication links, and protocols.

**Socio-economic, Legal Aspects: What surrounds the DL?** This facet is mainly related to the societal aspects of the DL and their relationships and interactions, including regulations, measures, and derivatives. It abstracts aspects surrounding the other DL issues and involves policies, economic issues, standards, and qualities. For example, policies may dictate that only certain communities have the right to use specific portions of a collection. Some of these DL issues can be established regarding normative structured documents. Policies and quality control also can be enforced by specific services, for example, authentication, authorization [245], encryption, and specific practices (scenarios) or protocols, which can involve other communication services and serialized streams.

**Environment: In what contexts are DLs embedded?** The environment involves a set of spaces (e.g., the physical space, or a concept space defined by the words of a natural language) that defines the use and the context of a DL. The environment also involves the society that sets up the DL and uses it. But the environment is also how the DL fits into the structure of community and its organization and dictates the scenarios by which its activities are performed. Those who pursue *Academic Disciplines* define a problem area “per se” and build a rational consensus of ideas and information about the problem that leads to a solution [577]. Thus they carve out a space for their approaches (e.g., in terms of concepts in a domain language, etc.), and structure some subject knowledge jointly with specific scenarios that define the methods or activities used to solve their specific problems. *Purposes* and *Scope* define types of societies served by the DL and determine a specific library structure.

## 1.11 SUMMARY

Two important efforts in formally defining DLs are the 5S framework [250, 254] and the DELOS reference model [97]. The 5S framework uses the notions of streams, structures, spaces, scenarios, and societies to describe essential DL concepts, such as digital object, metadata, collection, and services. These, in turn, are used to define a *minimal* digital library. This minimal definition has been extended to define, among others, an integrated DL [595] (Chapter 5), exploring services in DLs [595, 597] (Chapter 2), and a quality model for DLs [259]. On the other hand, the DELOS reference model emphasizes being exhaustive and listing all possible DL concepts and then defining them. In some cases, a deep analysis also has resulted; for example, a formal model of annotations by Agosti and Ferro [14] rigorously defines an annotation and its components.

Thus, this chapter provides both a broad introduction, and a foundation for the following chapters, where the 5S framework is explored at the basic, advanced, and applied levels.

## 1.12 EXERCISES AND PROJECTS

1. Pick your favorite digital library. Describe it at a high level using the 5S approach.

## CHAPTER 2

# Exploration

by Rao Shen

*Abstract:* Exploring services for digital libraries (DLs) include two major paradigms, browsing and searching, as well as other services such as clustering and visualization. In this chapter, we formalize and generalize DL exploring services within a DL theory. 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. We conducted a user study to evaluate ETANA-DLs exploring services and to test our hypotheses.

## 2.1 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 [490]). 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 [49, 50, 233, 249, 418]. 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 [328], which combines browsing, searching, document clustering, and infor-

mation 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:

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

To address the above mentioned questions, we

1. Generalize DL exploring services such as browsing, searching, clustering, and visualization in the context of the 5S DL theory (see Chapter 1 and [250, 254]) and develop theorems and lemmas based on the formal generalization.
2. Prove that browsing and searching can be converted and mapped to each other under certain conditions based on the theorems and lemmas developed.
3. Use an integrated archaeological DL, ETANA-DL (<http://etana.dlib.vt.edu>) [537, 596], as a case study to illustrate the application of our theoretical approach. We conducted a user study to evaluate ETANA-DLs 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.

## 56 2. EXPLORATION

The remainder of this chapter is structured as follows. Section 2 discusses related work. Section 3 formalizes DL exploring services. Section 4 describes the exploring services for our archaeological DL, developed based on the theorems and lemmas. Summaries are outlined in section 4.

### 2.2 RELATED WORK

The idea of integrating searching and browsing can be found in some early systems in the 1980s, such as I3R [136] and RABBIT [673]. Though I3R 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 10 years after RABBIT and I3R appeared, searching and browsing integration resurfaced in many efforts, such as PESTO [101] and DataWeb [430]. 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 [442] 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 [444], which adopted PESTOs 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 [685], WebGlimpse [413], ScentTrails [490], and SenseMaker [38]. AMIT (Animated Multiscale Interactive TreeViewer) [685] is a Java applet that integrates fisheye tree browsing

with search and filtering techniques. WebGlimpse [413] allowed the search to be limited to a neighborhood of the current document.

ScentTrails [490] annotated the hyperlinks of retrieved Web pages with search cues: indications that a link leads to content that matches the search query. The annotation was 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 [38] increased the fluidity between browsing and searching DLs by introducing structured-based filtering and structured-based searching. In SenseMaker, a user issued a query and aggregated the retrieved results into bundles by bundling criterion (e.g., same author). Structured-based filtering allowed 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 structured-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.

Web browsing experience has been used to improve Web search application by Yahoo! and Google. Users browsing history can be interpreted by the clickthrough logs, which are a large and important source of user behavior information. This feedback provides detailed and valuable information about users interactions with the system as the issued query, the retrieved documents and their ranking.

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 the next section, we will show that related works like those above can be viewed as cases of our theoretical approach. We first formalize the DL exploring services in the context of a DL theory, 5S (see Chapter 1 and [250, 254]); then we prove that, when certain conditions are met, searching and browsing are duals; thus, mapping or conversions between them are readily supported.

## 2.3 EXPLORING SERVICE FORMALIZATION

**Notation:** Let  $C$  be a collection (a collection is a set of digital objects; see Def. 17 in Section 1.9 for details), and  $2^C$  be the set of all subsets of  $C$ . Let  $\phi$  be an 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 Section 1.8 for details)

## 58 2. EXPLORATION

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 Section 1.8 and all possible restrictions of the StructuredStream (see Def. 15. in Section 1.9 for details) functions of digital objects.
3.  $P : V_H \longrightarrow 2^{Contents}$  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 defined in Def. 22 in Section 1.9.

According to the definition of a minimum DL in Section 1.9, 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)  $\{cluster_1, cluster_2, \dots, cluster_k\}$ , where  $cluster_i$  is denoted as a cluster belonging to  $subC$ , and  $\bigcup_{i=1}^k cluster_i = subC$ .

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

Let  $VSpa$  be a vector space (see Def. 13 in Appendix A) 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 2.1** 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 Section 1.8 for details),  $Contents_q$  can include digital objects and all of their streams, and  $P_q$  is a mapping function  $P_q : V_q \longrightarrow Contents_q$ .

The notion of *conceptual representations* for user information needs was used in Section 1.9 to define searching service, however, it was not formally defined. Def. 2.1 is a formal definition for conceptual representations for user information needs. Based on Def. 2.1, 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.

1. Example a): Textual retrieval:  $q$  is a key word named energy.

A user's information need is something about energy, she may explicitly express it as a key word "energy".

### 2.3. EXPLORING SERVICE FORMALIZATION 59

$q = ((V_q, E_q), L_q, F_q), \text{Contents}_q, P_q$ , where  $V_q = \{v_1\}, E_q = \phi, L_q = \phi, F_q : V_q \rightarrow L_q, \text{Contents}_q$  is the stream of string “energy”, and  $P_q : V_q \rightarrow \text{Contents}_q$ .

In this case,  $H_q = ((V_q, E_q), L_q, F_q)$  is a one-node graph (see Fig.2.1), and  $P_q$  maps that node to its contents, i.e., string “energy” (indicated by the dashed arrows in Fig. 2.1).



Figure 2.1:  $q$  is a key word named energy.

2. Example 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 [596]).  $q$  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 user’s interested 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*.).  $q = ((V_q, E_q), L_q, F_q), \text{Contents}_q, P_q$ , where  $V_q = \{v_1, v_2\}, E_q = \phi, L_q = \{\text{objecttype}, \text{site}\}, F_q : V_q \rightarrow L_q, \text{Contents}_q$  is the stream of strings “animal bones” and “Nimrin”, and  $P_q : V_q \rightarrow \text{Contents}_q$ .

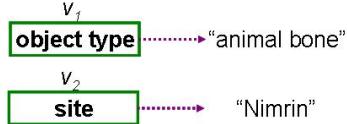


Figure 2.2:  $q$  is a structured query named animal bones from the Nimrin site.

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. 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. 2.2).

Structured query  $q$  was defined as a set of attribute-value pairs:  $q = \{A_1 : value_{1q}, \dots, A_k : value_{kq}, \dots, A_n : value_{nq}\}$ , where  $A_k$  is an attribute or metadata field and each  $value_{kq}$  a value belonging to the domain of  $A_k$  [252]. We find that this definition can be derived from Def2.1 (definition of a set of conceptual representations

## 60 2. EXPLORATION

for user information needs). By Def2.1, 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  $v_k$ .

3. Example 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. 2.3 (a).  $q = ((V_q, E_q), L_q, F_q), Contents_q, P_q)$ , where  $V_q = \{v_1, v_2, v_3, v_4, v_5\}, E_q = \{e_1, e_2, e_3, e_4, e_5\}, L_q = \{\text{'fire}', \text{'earth'}, \text{'metal'}, \text{'water'}, \text{'wood'}\}, F_q : V_q \cup E_q \rightarrow L_q, Contents_q$  is the stream of the five spatially related sub-images with their location information, and  $P_q : V_q \rightarrow Contents_q$ .

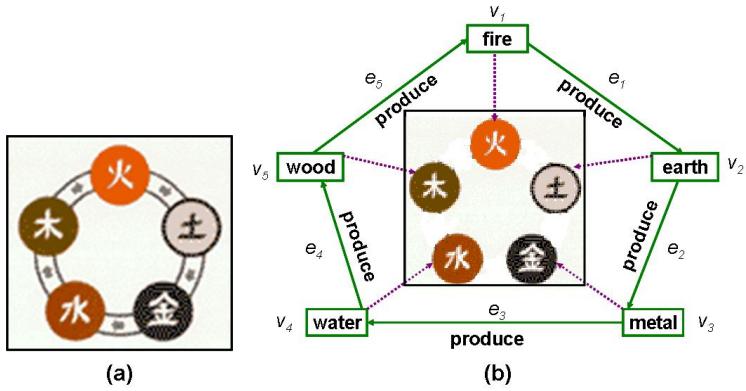


Figure 2.3:  $q$  is an image of 5 spatially related sub-images.

In this case,  $H_q$  is a graph of five nodes with labels ‘fire’, ‘earth’, ‘metal’, ‘water’, ‘wood’, ‘produce’ respectively as illustrated in Fig. 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. 2.3). This kind of query representation has been used to retrieve images according to spatial relationships of objects or layout representations (e.g., [55, 589]).

4. Example d) Navigation starting point

$q = ((V_q, E_q), L_q, F_q), Contents_q, P_q)$ , where  $V_q = \{v_1\}, E_Q = \phi, L_q = \{\text{'ETANA - DL'}\}, F_q : V_q \rightarrow L_q, Contents_q$  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. 2.4), and  $P_q$  maps that node to its contents, i.e., the ETANA-DL homepage (indicated by the dashed arrows in Fig. 2.4). In this situation, a user does not have an explicit information need like a query though she may have a conceptual

### 2.3. EXPLORING SERVICE FORMALIZATION 61

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.



Figure 2.4:  $q$  is a user's navigation start point.

**Definition 2.2** An Exploration Space (**ESpa**) is a tuple,  $\mathbf{ESpa} = (Q, \text{Contents}, OP_{Set})$ , where  $Q$  is a set of conceptual representations for user information needs (see Def.2.1), Contents can include digital objects of a collection  $C$  ( $C$  is a set of digital objects), 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  $\text{Contents}$ .  $OP_{viz}, OP_s, OP_b, OP_{clu} \subseteq OP_{Set}$ , 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. 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

## 62 2. EXPLORATION

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.

<b>Nimrin</b>	<b>Bone</b>	<b>ID 1</b>	<a href="#">Partition NW</a>	<a href="#">Subpartition N40/W25</a>	<a href="#">Locus 178</a>	<a href="#">Container 212</a>	<b>PIECES 3</b>
AGES IRON II	AGE 900-800 BC						
BONE METAPODIAL	ANIMAL SHEEP / GOAT						
COMMENTS							
[View complete record]	[Add to Items of Interest]	[Share Item]					
<b>Nimrin</b>	<b>Bone</b>	<b>ID 1169</b>	<a href="#">Partition NW</a>	<a href="#">Subpartition N40/W25</a>	<a href="#">Locus 159</a>	<a href="#">Container 77</a>	<b>PIECES 1</b>
AGES IRON II	AGE 850-800 BC / L 9BC						
BONE METAPODIAL	ANIMAL MEDIUM MAMMAL						
COMMENTS UNIDENTIFIED, IM							
[View complete record]	[Add to Items of Interest]	[Share Item]					
<b>Nimrin</b>	<b>Bone</b>	<b>ID 1370</b>	<a href="#">Partition NW</a>	<a href="#">Subpartition N35/W20</a>	<a href="#">Locus 64</a>	<a href="#">Container 168</a>	<b>PIECES 1</b>
AGES IRON II	AGE 800-700 BC						
BONE METAPODIAL	ANIMAL MEDIUM MAMMAL						
COMMENTS UNIDENTIFIED							
[View complete record]	[Add to Items of Interest]	[Share Item]					

Figure 2.5: Example of  $OP_{viz}$

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

where  $Sim_{clu} = \{OP_{clu1}(cluster_x, cluster_y) | cluster_x \in 2^C, cluster_y \in 2^C\}$ , where  $OP_{clu1} : 2^C \times 2^C \longrightarrow R$  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 [565]. The range of  $OP_{clu}$  is a set of the  $Contents$  associated with collection  $C$ . Note that  $OP_{viz}$  may be applied on 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. 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. 2.7.

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

$Sim_s = \{OP_q(q, do) | q \in Q, do \in C\}$ , where  $OP_q : Q \times C \longrightarrow R$  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 Section 1.9, the retrieved results were not

### 2.3. EXPLORING SERVICE FORMALIZATION 63



Figure 2.6: Example of *cluster<sub>x</sub>* and *cluster<sub>y</sub>* in ETANA-DL

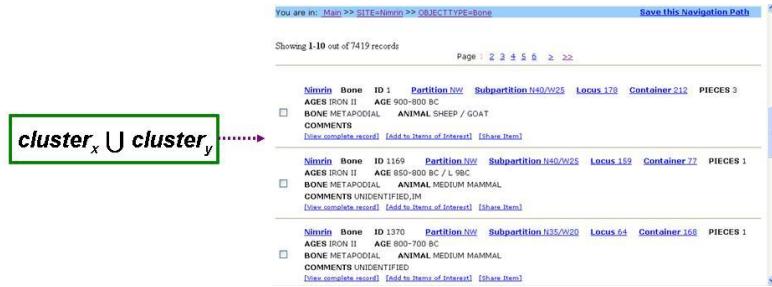


Figure 2.7: Example of clustering result

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  $OP_s$ :

$q$  is a structured query named ‘‘animal bones from the Nimrin site’’ as illustrated in Fig. 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) [566]. 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. 2.8.

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

The *TraverseLink* function defined in Section 1.9 was intended to achieve the same result as  $OP_b$ . We think both the domain and range of *TraverseLink* function may

## 64 2. EXPLORATION

Showing 1-10 out of 7419 records

7419 animal bone records are similar to the query

Nimrin	Bone	ID	Partition	Subpartition	Locus	Container	Pieces
Nimrin	Bone	ID 6607	Partition NW	Subpartition N25/W50	Locus 112	Container 404	Pieces 3
AGES PERSIAN AGE 539-332 BC / PERSIAN							
BONE LONG BONE ANIMAL MEDIUM MAMMAL							
COMMENTS FRAG							
[View complete record] [Add to Items of Interest] [Share Item]							
Nimrin	Bone	ID 5026	Partition NW	Subpartition N25/W20	Locus 12	Container 282	Pieces 1
AGES PERSIAN AGE 600-500 BC / EARLY PERSIAN							
BONE PROXIMAL PHALANX ANIMAL SHEEP / GOAT							
COMMENTS COMPLETE							
[View complete record] [Add to Items of Interest] [Share Item]							
Nimrin	Bone	ID 3674	Partition NW	Subpartition N30/W20	Locus 20	Container 327	Pieces 4
AGES BYZANTINE AGE AD 324-491 / E BYZ							
BONE CAUDAL VERTEBRA ANIMAL MEDIUM MAMMAL							
COMMENTS IMMATURE							
[View complete record] [Add to Items of Interest] [Share Item]							

Figure 2.8: Example of function  $OP_s$  in ETANA-DL

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 an 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  $2^{Contents}$  instead of  $Contents$  as the range of function  $TraverseLink$ . 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. 2.9).

**Definition 2.3** An exploring service (**ESer**) is a set of scenarios  $\{sc_1, \dots, sc_n\}$  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. 2.10 shows two constructs of an exploring service. The left part of Fig. 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,

### 2.3. EXPLORING SERVICE FORMALIZATION 65

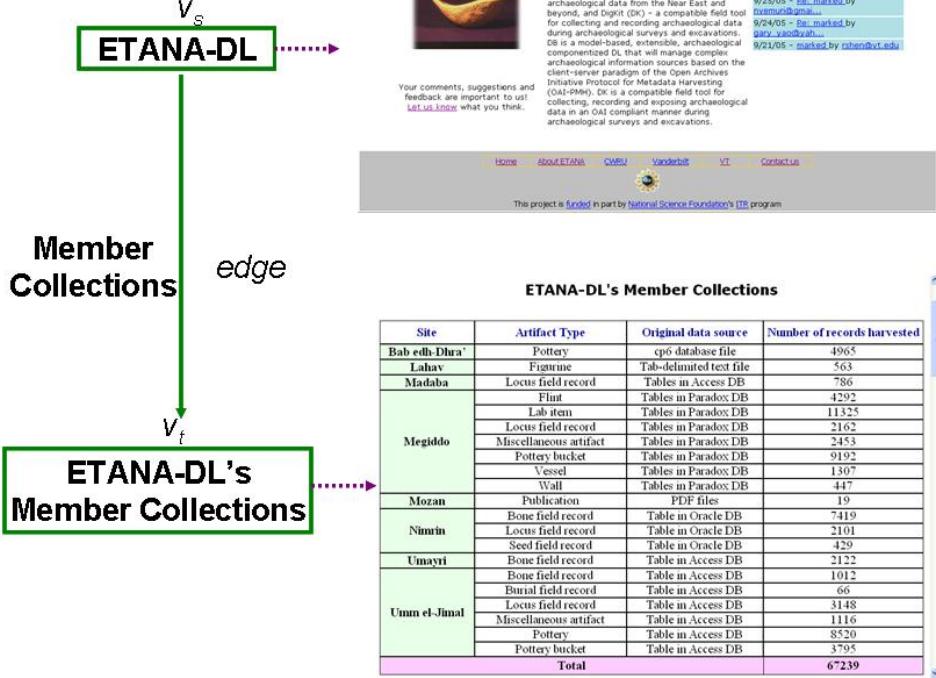


Figure 2.9: Example of function  $OP_b$  in ETANA-DL

clustering, or visualization service as illustrated in the following theorems and lemmas according to Def.2.1, Def. 2.2, and Def.2.3. 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. 2.11).

Fig. 2.12 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 state 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.

## 66 2. EXPLORATION

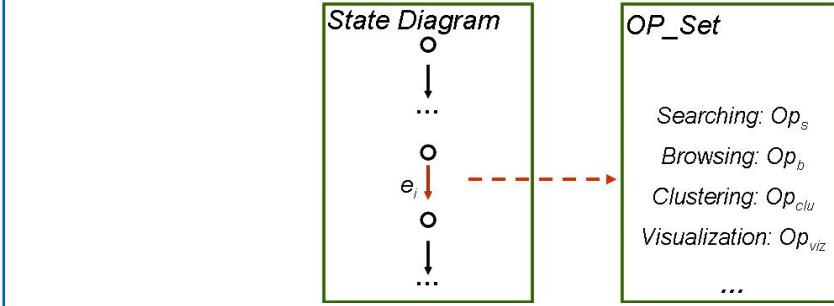


Figure 2.10: Constructs for an exploring service

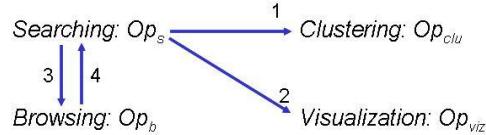


Figure 2.11: Sequence of operations

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

The event  $e_i$  in Fig. 2.13 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.2.1), 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  $OP_{viz}$  function may be applied on searching results.

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

By Def. 23 of Section 1.9, a browsing service is associated with an underlying hypertext construct. Event  $e_i$  in Fig. 2.14 models a path through a website a user follows to access the target node. It invokes operation  $OP_b$  defined in Def.2.2. The output of  $OP_b$

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 $Op_b$ )	✓	✓	✓	

Figure 2.12: Relationship among theorems (lemmas) and operations

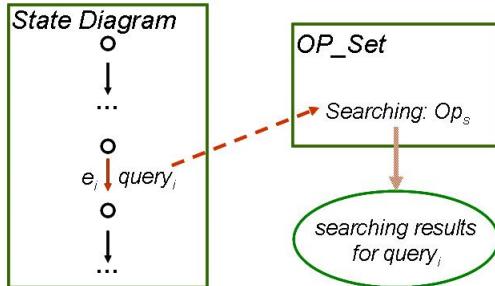


Figure 2.13: An exploring service is a searching service.

is the contents of the target node. A sequence of target nodes,  $v_{t,0}, v_{t,1}, \dots, v_{t,i}, \dots, v_{t,k}$ , associated with a sequence of events,  $e_0, e_1, \dots, e_i, \dots, e_k$ , is denoted as a user's navigation path  $\pi$ .

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  $\pi$  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  $\pi$  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

## 68 2. EXPLORATION

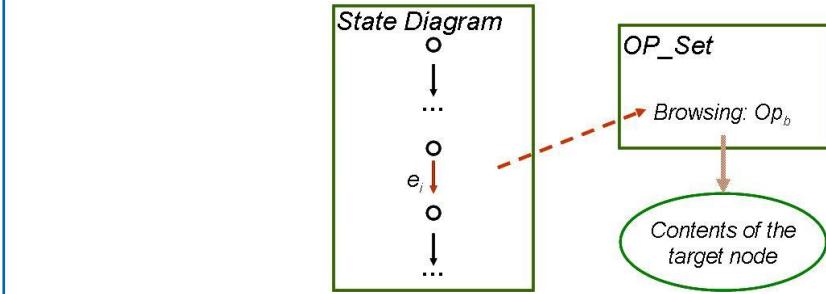


Figure 2.14: An exploring service is a browsing service.

hypertext; however, the vertex does not belong to  $V_q$  (i.e., the node is not included in the users navigation path  $\pi$ ). 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.2.2, 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.

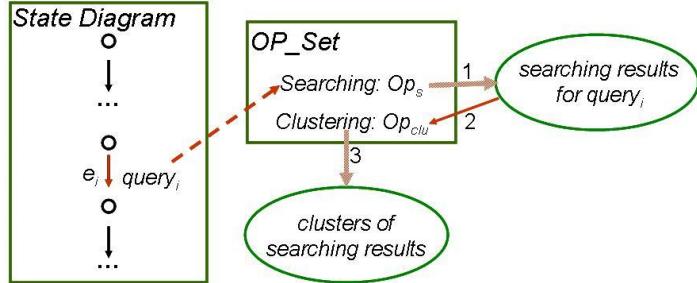


Figure 2.15: An exploring service is a browsing service.

The event  $e_i$  in Fig. 2.15 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  $sim_{clu} = OP_{clu}(cluster_x, cluster_y)$  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  $Espabrowse = (Q_{browse}, Contents_{browse}, OP\_Set_{browse})$  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  $\pi$  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  $\Pi$  be as a set of  $\pi$ , where  $\pi$  is a user's navigation path, a sequence of target nodes,  $v_{t\_0}, v_{t\_1}, \dots, v_{t\_i}, \dots, v_{t\_k}$ , associated with a sequence of events,  $e_0, e_1, \dots, e_i, \dots, e_k$ .

1.  $Eser_{browse}$  can be converted to  $Eser_{search}$ , denoted  $Eser_{browse} \Rightarrow Eser_{search}$ , if  $\exists M_1 : \Pi \longrightarrow Q_{search}$ , 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  $\exists M_2 : Q_{search} \longrightarrow \Pi$ , 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  $v_{t\_k} \in V_q$  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 users navigation path  $\pi$ . Therefore,  $Eser_{search} \Rightarrow Eser_{browse}$ .

Example:

The rectangle shown in Fig. 2.16 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. 2.16); the second one  $v_{t\_1}$  is related to a page about 9541 bone records (illustrated by an arrow numbered 2 in Fig. 2.16); 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

## 70 2. EXPLORATION

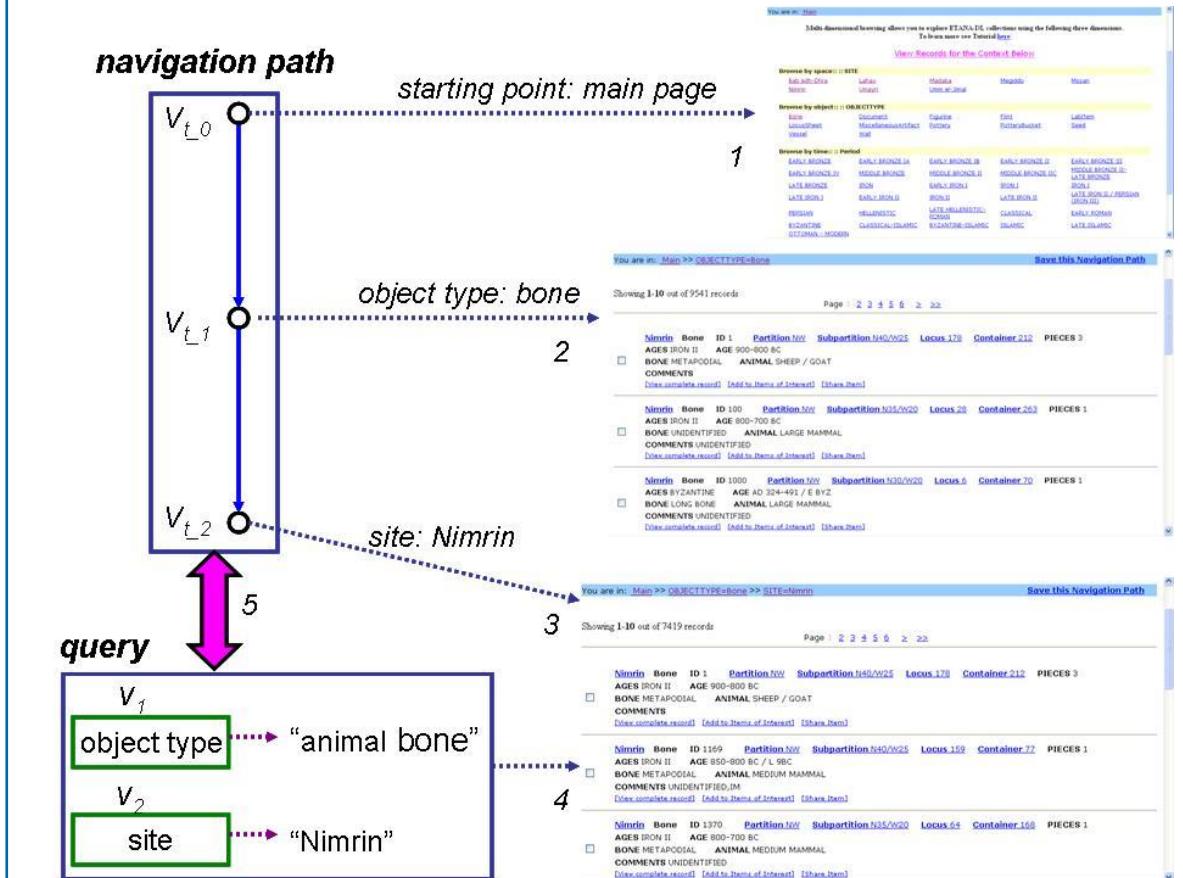


Figure 2.16: Example of mapping between navigation path and a structured query

Fig. 2.16) and it displays the retrieved results for a structured query (illustrated by an arrow numbered 4 in Fig. 2.16). The bidirectional arrow numbered 5 in Fig. 2.16 denotes that the navigation path and the structured query can be mapped to each other.

**Lemma 2:** Given  $Q_{search} = \{q_1, q_2, \dots, q_n\}$ ,  $\Pi = \{\pi_1, \pi_2, \dots, \pi_n\}$ , where  $\pi_i$  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. 2.2),  $OP_s^{-1}(contents_i) = q_i$ , and  $P_b^{-1}(contents_i) = \pi_i$ , then  $\exists M_1, \exists M_2, Eserbrowse \Rightarrow Esersearch$ , and  $Esersearch \Rightarrow Eserbrowse$ .

Proof:

### 2.3. EXPLORING SERVICE FORMALIZATION 71

1.  $\exists M_1, \forall \pi_i \in \Pi, M_1(\pi_i) = OP_s^{-1}(OP_b(v_{i.t.k-1}, v_{i.t.k})) = OP_s^{-1}(\text{contents}_i) = q_i$ , therefore, according to Lemma 1,  $\exists M_1 : \Pi \rightarrow Q_{\text{search}}$  and  $E_{\text{serbrowse}} \Rightarrow E_{\text{sersearch}}$ .
2.  $\exists M_2, \forall q \in Q_{\text{search}}, M_2(q_i) = OP_b^{-1}(OP_s(q_i)) = OP_b^{-1}(\text{contents}_i) = \pi_i$ , therefore, according to Lemma 1,  $\exists M_2 : Q_{\text{search}} \rightarrow \Pi$  and  $E_{\text{sersearch}} \Rightarrow E_{\text{serbrowse}}$ . As shown in Fig. 2.17, both “ $query_i$ ” and “ $\pi_i$ ” are associated with the same results, therefore,  $\exists M_1 : M_1(query_i) = \pi_i, \exists M_2 : M_2(\pi_i) = query_i, E_{\text{serbrowse}} \Rightarrow E_{\text{sersearch}}$  and  $E_{\text{sersearch}} \Rightarrow E_{\text{serbrowse}}$ .

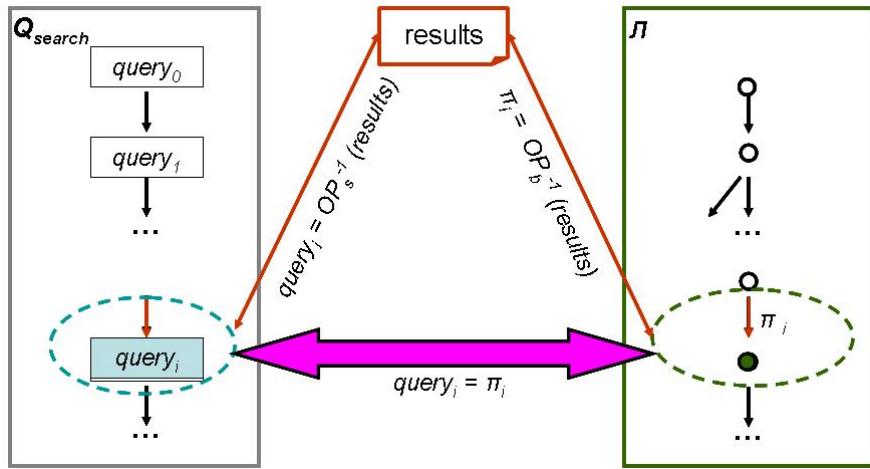


Figure 2.17: “ $query_i$ ” and “ $\pi_i$ ” are associated with the same results.

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. 2.18. In this example, searching results are displayed along with the query  $q$  and browsing results are displayed along with the corresponding navigation path  $\pi$ . Therefore, there exists function  $M_1$  and  $M_1$ , such that  $OP_s^{-1}(\text{results}) = q$  and  $OP_b^{-1}(\text{results}) = \pi$ , where  $\text{results}$  are represented by the 3 acacia seed records.

PESTO [101], DataWeb [430], and MIX [442] 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,  $E_{\text{serbrowse}} \Rightarrow E_{\text{sersearch}}$ .

## 72 2. EXPLORATION



Figure 2.18: Example of Lemma 2

**Lemma 3:** Let  $Esp_{postBrowse} = (Q_{postBrowse}, Contents_{postBrowse}, OP\_Set_{postBrowse})$  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  $OP_s : (Q_{postBrowse} \times C_{postBrowse}) \times Sim_s \rightarrow 2^{Contents}$ , 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 [101], DataWeb [430], and MIX [442] can be generalized as shown in Fig. 2.19. The arrow numbered 1 points to the browsing results associated with navigation path  $\pi_i$ . Since  $\pi_i$  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  $query$ . 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. 2.19. 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.

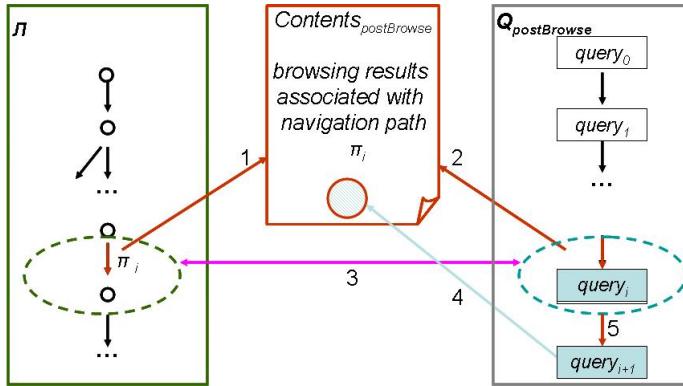


Figure 2.19: “ $query_{i+1}$ ” is refined from “ $query_i$ ” after browsing.

**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  $Q_{postRetr} = \{(V_{q_{postRetr}}, E_{q_{postRetr}}), L_{q_{postRetr}}, F_{q_{postRetr}}, Contents_{q_{postRetr}}, P_{q_{postRetr}}\}$  (see Def.2.1),  $Contents_{postRetr}$  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  $OP\_Set_{postRetr} = \{OP_s, OP_{clu}\}$ ,  $cluCon_{retr} = \{OP_{clu}((cluster_x, cluster_y), sim_{clu}) | cluster_x, cluster_y \subseteq C_{retr}\} = \{cluCon_{retr-1}, cluCon_{retr-2}, \dots, cluCon_{retr-i}, \dots, cluCon_{retr-z}\}$  is the contents of clustered retrieved results, where  $sim_{clu} = OP_{clu1}(cluster_x, cluster_y)$  (see Def.2.2),

## 74 2. EXPLORATION

$\Pi = \{\pi_1, \pi_2, \dots, \pi_i, \dots, \pi_z\}$ , where  $\pi_i = (v_0, v_i)$  is a navigation path consisting of only two nodes,  $v_0, v_i \in V_{q_{postRetr}}$ , 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. 2.20. 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.

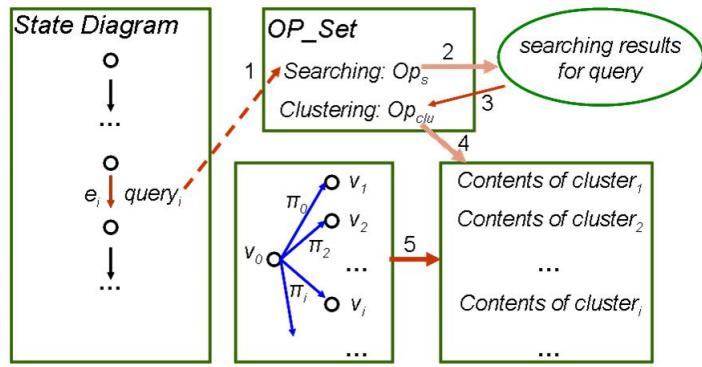


Figure 2.20: Switch from searching to browsing.

Proof:

$\exists v \in V_{q_{postRetr}}, v \in V_H$ , and  $e_i$ , 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  $\pi_i$ , therefore by Theorem 2,  $Eser_{postRetr}$  is a browsing service.

Categorizing or clustering searching results is a case of switching searching to browsing. ScentTrails [490] 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.

The event  $e_i$  in Fig. 2.21 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:

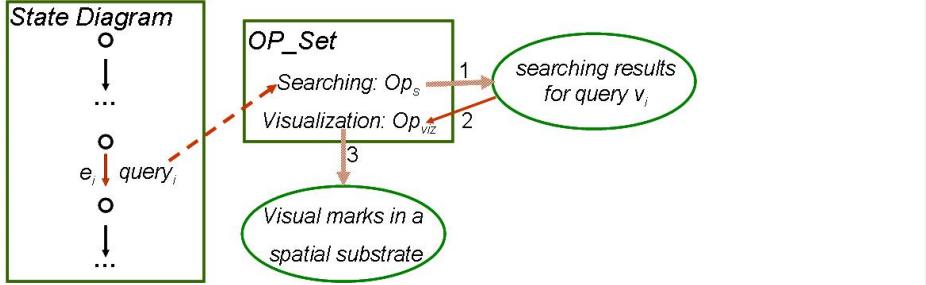


Figure 2.21: An exploring service is a visualization service.

$\forall q \in Q$ , 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.

## 2.4 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.

## 76 2. EXPLORATION

### 2.4.1 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.

The screenshot shows a web-based interface for multi-dimensional browsing. At the top, a blue header bar displays the navigation path: You are in: [Main](#) >> [SITE=Bab edh-Dhra](#) >> [PARTITION=A](#) >> [SUBPARTITION=056](#). To the right of the path is a link to [Save this Navigation Path](#). Below the header is a search bar with the placeholder "Search within this context for" and a "Go" button. A pink link [View Records for the Context Below](#) is visible. The main content area is divided into three horizontal sections, each with a yellow background and black text. The first section is labeled "Browse by space::" followed by the navigation path [SITE=Bab edh-Dhra::PARTITION=A::SUBPARTITION=056::LOCUS](#), with a link to [Unclassified](#). The second section is labeled "Browse by object::" followed by the object type [OBJECTTYPE](#), with a link to [Pottery](#). The third section is labeled "Browse by time::" followed by the time period [Period](#), with links to [EARLY BRONZE II](#) and [EARLY BRONZE III](#).

Figure 2.22: 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.

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. 2.22 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 = Babedh – Dhra* >> *PARTITION = A* >> *SUPARTITION = 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 di-

## 2.4. CASE STUDY: EXPLORING SERVICES IN ETANA-DL 77

Tomb #056 in Area A of Bab edh-Dhra,  
Time Period: EARLY BRONZE III

You are in: Main >> SITE=Bab edh-Dhra >> PARTITION=A >> SUBPARTITION=056 >> Period=EARLY BRONZE III [Save this Navigation Path](#)

Search within this context for  [Go](#)

[View Records for the Context Below](#) [View Records](#)

Browse by space:: SITE=Bab edh-Dhra::PARTITION=A::SUBPARTITION=056::LOCUS  
[Unclassified](#)

Browse by object:: :: OBJECTTYPE  
[Pottery](#)

Browse by time:: Period=EARLY BRONZE III::Chronology  
No SubCategories Present

Showing 1-1 out of 1 records

Page [1](#)

<a href="#">Bab edh-Dhra</a>	Vessel Number 029	<a href="#">Tomb Area A</a>	<a href="#">Tomb Number 056</a>
<input type="checkbox"/>	Ages EARLY BRONZE III	Basic Category Small bowls and Saucers	
	Rim Treatment unavailable	Handle Type unavailable	Mouth Width 104 Base Width 44
	<a href="#">View complete record</a> <a href="#">Add to Items of Interest</a> <a href="#">Share Item</a>		



Figure 2.23: Save current navigation path for later use and view records

mensions 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. 2.23) 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*.

### 2.4.2 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” [101]. The main idea is that browsing will present a useful starting point for active exploration of an answer space. Subsequent brows-

## 78 2. EXPLORATION

ing 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 = Babedh – Dhra >> PARTITION = A >> SUPARTITION = 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. 2.24. 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.

The screenshot shows a web-based search interface. At the top, a blue header bar displays the navigation path: 'You are in: Main >> SITE=Bab edh-Dhra >> PARTITION=A >> SUBPARTITION=056' and a 'Save this Navigation Path' button. Below the header is a search form with the placeholder 'Search within this context for' followed by a text input field containing 'saucer' and a 'Go' button. A pink link 'View Records for the Context Below' is visible. The main content area contains three sections with yellow headers: 'Browse by space:: SITE=Bab edh-Dhra::PARTITION=A::SUBPARTITION=056::LOCUS' with a link to 'Unclassified'; 'Browse by object:: :: OBJECTTYPE' with a link to 'Pottery'; and 'Browse by time:: :: Period' with links to 'EARLY BRONZE II' and 'EARLY BRONZE III'.

Figure 2.24: 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. 2.25). They are organized into three dimensions after the user clicks the button "View search results hierarchically" (see Fig. 2.26). 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. 2.27. 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.

## 2.4. CASE STUDY: EXPLORING SERVICES IN ETANA-DL 79

The screenshot shows the ETANA-DL search interface. At the top, there's a header bar with the title 'ETANA-DL Managing complex information applications: An archaeology digital library'. Below it is a navigation bar with links to 'Home', 'Member Collections', 'First Time Visit', 'Login', and 'Help'. A search bar contains the query 'equus', with a 'Go' button and links to 'Advanced Search' and 'Browse'. The main content area displays a message 'Total number of hits for equus : 88'. Below this is a button labeled 'View search results hierarchically'. A pagination bar shows 'Showing 1-10 out of 88 records' with links to pages 1 through 6 and a '>>' link. Two search results are listed:

- Record 1:** Umayri Bone ID 432 Partition B Subpartition 7K92 Locus 001 Container  
19 PIECES 1 AGES BONE ANIMAL EQUUS COMMENTS [View complete record] [Add to Items of Interest] [Share Item]
- Record 2:** Umayri Bone ID 910 Partition A Subpartition 7J69 Locus 005 Container  
22 PIECES 3 AGES PERSIAN BONE ANIMAL EQUUS COMMENTS [View complete record] [Add to Items of Interest] [Share Item]

Figure 2.25: Equus records are retrieved through basic searching

### 2.4.3 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. 2.28 . 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

## 80 2. EXPLORATION

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

[Home](#) | [Member Collections](#) | [First Time Visit](#) | [Login](#) | [Help](#)

Search ETANA-DL for   | [Advanced Search](#) | [Browse](#)

You are in: >> [query=equus](#)

Search within this context for

[View Records for the Context Below](#)

**Browse by space:: :: SITE**

<a href="#">Nimrin</a>	<a href="#">Umayri</a>
------------------------	------------------------

**Browse by object:: :: OBJECTTYPE**

<a href="#">Bone</a>
----------------------

**Browse by time:: :: Period**

<a href="#">IRON II</a>	<a href="#">ISLAMIC</a>	<a href="#">LATE HELLENISTIC- ROMAN</a>	<a href="#">PERSIAN</a>	<a href="#">MIDDLE BRONZE</a>
<a href="#">IRON I</a>	<a href="#">BYZANTINE</a>	<a href="#">EARLY BRONZE</a>	<a href="#">LATE IRON II</a>	<a href="#">LATE IRON II / PERSIAN (IRON III)</a>
<a href="#">MIDDLE BRONZE IIC</a>	<a href="#">EARLY BRONZE III</a>	<a href="#">EARLY IRON I</a>	<a href="#">CLASSICAL-ISLAMIC</a>	<a href="#">OTTOMAN - MODERN</a>
<a href="#">MIDDLE BRONZE II</a>	<a href="#">LATE BRONZE</a>			

Figure 2.26: Retrieved equus records are organized into 3 dimensions

You are in: >> [query=equus](#) >> SITE=Nimrin

Showing 1-10 out of 36 records

Page 1 2 3 4

<a href="#">Nimrin</a>	<a href="#">Bone</a>	ID 1472	<a href="#">Partition NW</a>	<a href="#">Subpartition N40/W25</a>	<a href="#">Locus 184</a>	<a href="#">Container 252</a>	PIECES 1
AGES IRON II AGE 900-800 BC							
<input type="checkbox"/>	<a href="#">BONE TOOTH</a>		<a href="#">ANIMAL EQUUS</a>				
<b>COMMENTS</b>							
<a href="#">[View complete record]</a> <a href="#">[Add to Items of Interest]</a> <a href="#">[Share Item]</a>							

Figure 2.27: Browse the 36 equus records from the Nimrin site after searching

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.2.2, 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

#### 2.4. CASE STUDY: EXPLORING SERVICES IN ETANA-DL 81

(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.

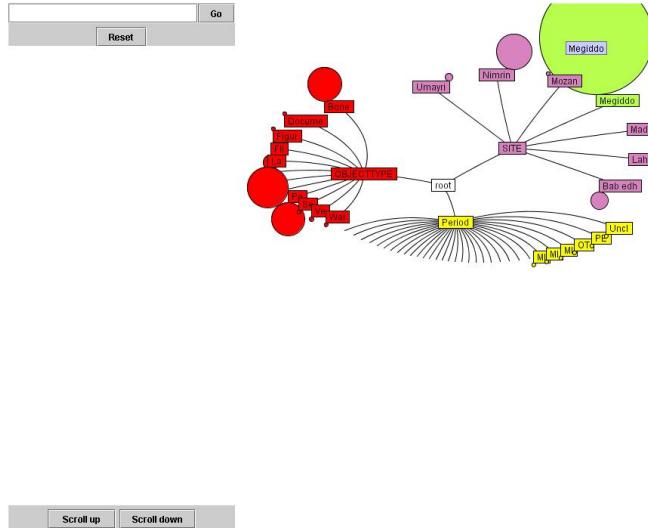


Figure 2.28: Initial interface of EtanaViz

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. 2.29. 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.

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,

## 82 2. EXPLORATION

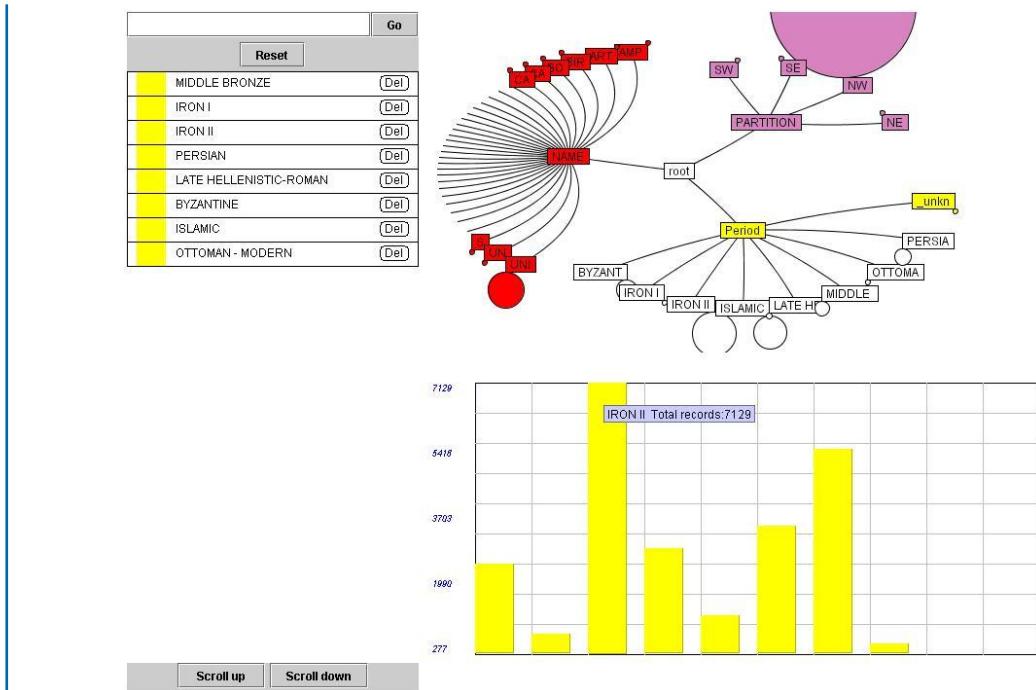


Figure 2.29: Total number of animal bones across Nimrin culture phrases

and four entries with different colors are included in the list on the left of the screen (see Fig. 2.30).

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 [668].

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.

## 2.4. CASE STUDY: EXPLORING SERVICES IN ETANA-DL 83

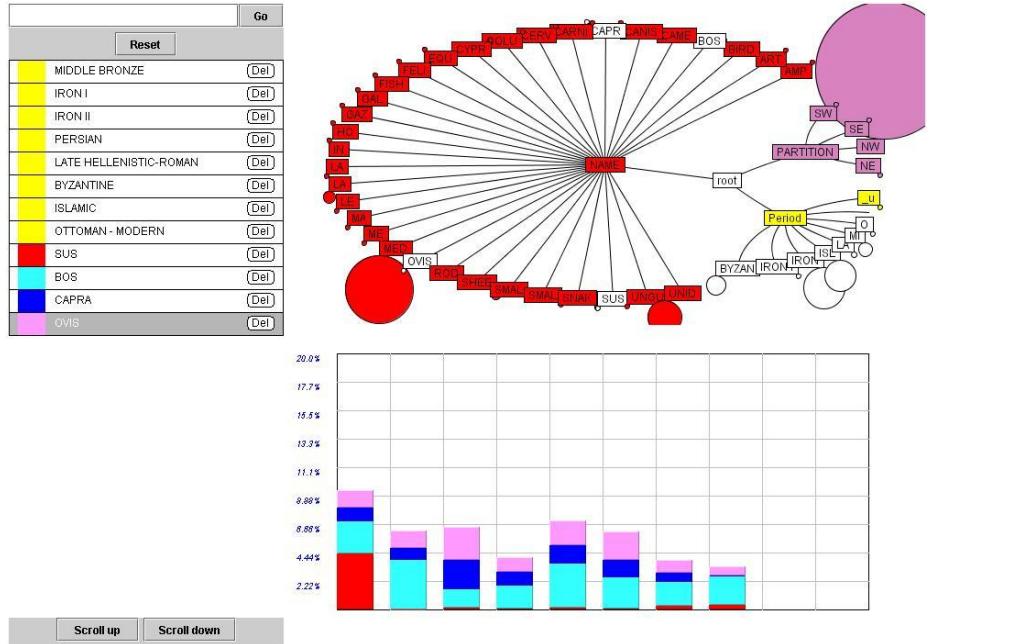


Figure 2.30: Percentages of animal bones across Nimrin culture phrases

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 [668].

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.

### 2.4.4 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

## 84 2. EXPLORATION

all the findings, we focus on only the findings that help validate the hypotheses related to browsing, searching, and visualization. When browsing service can be mapped to searching (*browsing*  $\Rightarrow$  *searching*), 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](http://etana.vt.edu:8080/etana/servlet/surveyTasks?submit_start). The experiment was conducted through four sessions. Each user was required to

- learn the online tutorial of ETANA-DL;
- complete a pre-evaluation questionnaire;
- perform tasks using ETANA-DL. After completion of each task, he (she) was asked to fill out a task-related questionnaire and give comments.
- 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 Fig.2.31) 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 Fig. 2.31 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 searchbrowsing 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"

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

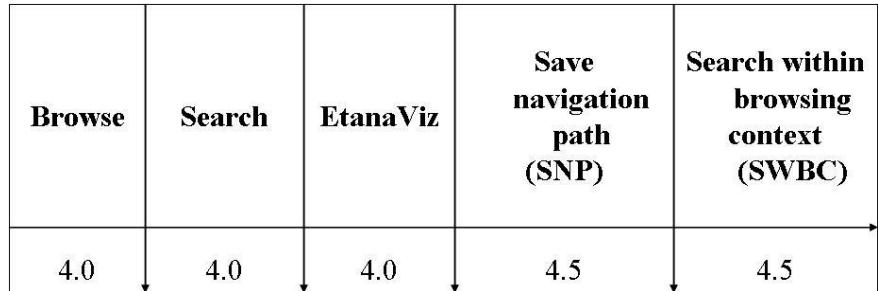


Figure 2.31: Impression about ETANA-DL services (mean value)

- H1: Impression about SWBC is larger than that for browsing at significance level 0.05.
- H2: Impression about SWBC is larger than that for searching at significance level 0.05.
- H3: Impression about SNP is larger than that for browsing at significance level 0.05.
- H4: 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). 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. 2.32 shows the average time for each of the four 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

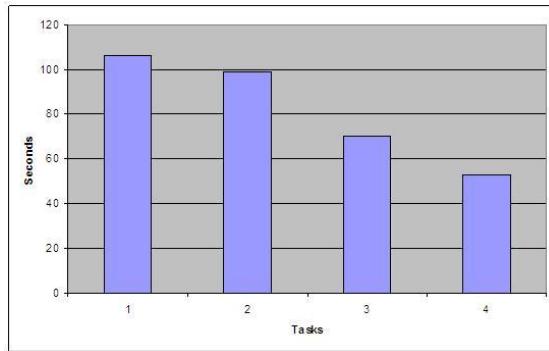


Figure 2.32: Average time on tasks

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 [171, 172].

## 2.5 SUMMARY

Exploring services for digital libraries (DLs) include two major paradigms, browsing and searching, as well as other services such as clustering and visualization. In this chapter, we formalize and generalize DL exploring services within a DL theory. 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. We conducted a user study to evaluate ETANA-DL's exploring services and to test our hypotheses.

## **2.6 EXERCISES AND PROJECTS**

1. How does faceted search fit in with the integration of searching and browsing?

## CHAPTER 3

# Evaluation

by Marcos André Gonçalves

*Abstract:* Evaluation is a necessity if scientific studies of digital libraries are to proceed and have impact.

### 3.1 INTRODUCTION

Once we establish the importance of digital libraries and their broad applicability, questions about the utility, usability, and cost of these systems appeared and greater attention has been given to their evaluation. To define what makes a DL a system of good quality or that has a potential to satisfy its users can be difficult and hard to summarize, since it depends on which of the many aspects of a DL are being considered. As has been pointed out by Fuhr et al. [229], when evaluating quality, people interested in DLs have disparate views of these systems and, as a consequence, focus on different aspects that are relevant to their specific point of view.

### 3.2 RELATED WORK

Works such as [260] and [342] present standards for DL log formats with the goal of recording data for the evaluation of DLs. Such formats are very important for tools such as 5SQual. In [260], an XMLLog format is described that captures detailed information about system behavior and access to its services, storing data that indicate critical aspects about user interactions with the DL, thus providing valuable information for system evaluation. [342] builds on that work and proposes a multilevel record scheme for DL logging.

Different approaches to evaluate the success of a DL have been studied (e.g., [581], [579], [229], [580], [641], [359], [595], and [231]) involving users, collections, and systems, aimed at identifying generalizable metrics or context specific methods. But the literature that reports evaluations with actual data is not substantial. It seems that evaluation theorists and practitioners do not communicate well, as noticed in [580]. 5SQual addresses that challenge; it is a tool that implements and follows a theoretical quality model for DLs, and that can help administrators in the evaluation of real DLs.

### 3.3 BACKGROUND AND CONTEXT

System-oriented perspective of Quality in DLs.

(Cover Rao's usage dimensions as per the lecture slides.)

### 3.4 FORMALIZATION

We draw from the 5S formal framework as well as from our own experience in building DLs since 1991 to derive a list of quality dimensions described below. We follow the standard terminology used in the *social sciences* [34]. We will use the term *composite quality indicator*<sup>1</sup> (or in short *quality indicator*) to refer to the proposed quantities instead of the stronger term *quality measure*. Only after one has a number of indicators, and they are validated<sup>2</sup> and tested for reliability<sup>3</sup>, can they be composed into reliable “measures”. Despite partial tests of validity (for example, through focus groups<sup>4</sup>) the proposed quality indicators do not qualify as measures yet. Also, it should be stressed that the proposed quantities are only approximations of or give quantified indication of a quality dimension. They should not be interpreted as a complete specification of a quality dimension, since more factors/variables could be relevant than are specified here. We will, however, reserve the right to use the term “measure” when talking about standard measures that have long been used by the CS / LIS communities. The distinction should be clear in context.

Table ?? shows a summary of proposed candidate dimensions of quality for some of the most important DL concepts defined above and factors affecting the measurement of the corresponding quality dimensions<sup>5</sup>. The following sections explain these indicators in detail by:

1. motivating them and discussing their meaning/utilization;
2. formally defining them and specifying their corresponding numerical computation; and
3. illustrating their use by applying the indicators/metrics in the context of some real-world DLs (e.g., ACM DL, CITIDEL, NDLTD).

Table ?? connects the proposed dimensions with some ‘S’-related concepts involved in their definition. In the same way that the formalized 5S theory helps to precisely define the higher level DL concepts used here, we will use these formalizations to help define the quality indicators and their corresponding computations.

<sup>1</sup>An indicator composed of two or more simpler indicators or variables.

<sup>2</sup>According to [34], validity refers to the extent to which a specific measurement provides data that relate to commonly accepted meanings of a particular concept. There are numerous yardsticks for determining validity: face validity, criterion-validity, content validity, and construct validity.

<sup>3</sup>Also according to [34], reliability refers to the likelihood that a given measurement procedure will yield the same description of a given phenomena if that measurement is repeated.

<sup>4</sup>A type of face validity.

<sup>5</sup>For simplicity, we focus on a DL concept and an indicator, not mentioning all other DL concepts that also relate. Thus, while we assign “relevance” to “digital object”, we are aware that users and queries clearly are involved too.

### 90 3. EVALUATION

**Table 3.1:** DL high-level concepts and corresponding DL dimensions of quality with respective metrics

DL Concept	Dimension of Quality	Factors/Variables Involved in Measuring
Digital object	Accessibility	Collection, # of structured streams, rights management metadata, communities
	Pertinence	Context, information, information need
	Preservability	Fidelity (lossiness), migration cost, digital object complexity, stream formats
	Relevance	Query (representation), digital object (representation), external judgment
	Similarity	Same as in relevance, plus: citation/link patterns
	Significance	Citation/link patterns
	Timeliness	Age, time of latest citation, collection freshness
Metadata specification	Accuracy	Accurate attributes, # of attributes in the record
	Completeness	Missing attributes, schema size
	Conformance	Conformant attributes, schema size
Collection	Completeness	Collection size, size of the ‘ideal collection’
Catalog	Completeness	# of digital objects without a set of metadata specifications, size of the described collection
	Consistency	# of sets of metadata specifications per digital object
Repository	Completeness	# of collections
	Consistency	# of collections in repository, catalog/collection pairwise consistency
Services	Composability	Extensibility, reusability
	Efficiency	Response time
	Effectiveness	Precision/recall (search), F1 measure (classification)
	Extensibility	# of extended services, # of services in the DL, # of lines of code per service manager
	Reusability	# of reused services, # of services in the DL, # of lines of code per service manager
	Reliability	# of service failures, # of accesses

**Table 3.2:** Dimensions of quality and Ss involved in their definitions

DL Concept	Dimension of Quality	Some 'S' Concepts Involved
Digital object	Accessibility	Societies (actor), Structures (metadata specification), Streams + Structures (structured streams)
	Pertinence	Societies (actor), Scenarios (task)
	Preservability	Streams, Structures (structural metadata), Scenarios (process (e.g., migration))
	Relevance	Streams + Structures (structured streams), Structures (query), Spaces (Metric, Probabilistic, Vector)
	Similarity	Same as in relevance, plus: Structures (citation/link patterns)
	Significance	Structures (citation/link patterns)
	Timeliness	Streams (time), Structures (citation/link patterns)
Metadata specification	Accuracy	Structure (properties, values)
	Completeness	Structure (properties, schema)
	Conformance	Structure (properties, schema)
Collection	Completeness	Structure (Collection)
Catalog	Completeness	Structure (Collection)
	Consistency	Structure (Collection)
Repository	Completeness	Structure (Collection)
	Consistency	Structure (Catalog, Collection)
Services	Composability	see Extensibility, reusability
	Efficiency	Streams (time), Spaces (operations, constraints)
	Effectiveness	see Pertinence, Relevance
	Extensibility	Societies + Scenarios (extends, inherits_from, redefines)
	Reusability	Societies + Scenarios (includes, reuses)
	Reliability	Societies + Scenarios (uses, executes, invokes)

## 92 3. EVALUATION

### 3.5 DIGITAL OBJECTS

#### 3.5.1 ACCESSIBILITY

A digital object is accessible by a DL actor or patron, if it exists in the repository of the DL, a service is able to retrieve the object, and: 1) an overly restrictive rights management property of a metadata specification does not exist for that object; or 2) if such exists, the property does not restrict access for the particular society to which the actor belongs or to that actor in particular. A quality indicator for calculating accessibility is a function, which depends on all those factors and the granularity of the rules (e.g., entire object, structured streams). It should be noted that digital object accessibility as defined here is different from the common view of “Web site accessibility”, which is concerned with creating better ways to provide Web content to users with disabilities [553]. For reasons of space we omit discussion of indicators associated with that type of accessibility.

Let  $access\ constraint$  be a property of some metadata specification of a digital object  $do_x$  whose values include the sets of communities that have the right to access specific (structured) streams within the object. Also let  $struct\_streams(do_x) = \Omega_x$  be the set of structured streams of  $do_x$ . The accessibility  $acc(do_x, ac_y)$  of a digital object  $do_x$  to an actor  $ac_y$  is:

- 0, if there is no collection  $C$  in the DL repository R such that  $do_x \in C$ ;
- otherwise  $acc = (\sum_{z \in struct\_streams(do_x)} r_z(ac_y)) / |struct\_streams(do_x)|$ , where:  
 $r_z(ac_y)$  is a rights management rule defined as an indicator function:
  - 1, if
    - z has no access constraints; or
    - z has access constraints and  $ac_y \in cm_z$ , where  $cm_z \in Soc(1)$  is a community that has the right to access z; and
  - 0, otherwise.

Notice that, from a broader perspective, the accessibility of a given digital object could be affected, not only by rights management, but also by technological constraints, such as lack of Acrobat Reader to open a full-text paper in PDF format, temporary network disconnection, or restriction on the number of simultaneous users, etc. In this work, though, we have tried to focus on easily measurable intrinsic properties of the objects themselves and of the relationships between the actor and the objects.

**Example of use.** At Virginia Tech, a student can choose, at the moment of submission, to allow her electronic thesis or dissertation (ETD) to be viewed worldwide, by those at the originating university, or not at all. The “mixed” case occurs when some portions (e.g., particular chapters or multimedia files) have restricted access while others are more widely available. The majority of Virginia Tech students choose their documents to

### 3.5. DIGITAL OBJECTS 93

be viewable worldwide—some initially choose not to grant worldwide access, because of concerns regarding patents or publication of results in journals/conferences.

Therefore the accessibility  $acc(etd_x, ac_y)$  of a Virginia Tech ETD  $etd_x$  is:

- 0, if  $etd_x$  does not belong to the VT-ETD collection;
- otherwise  $(\sum_{z \in struct\_streams(etd_x)} r_z(ac_y)) / |struct\_streams(etd_x)|$ , where:  
 $r_z(ac_y)$  is a rights management rule defined as an indicator function:  
 1, if  
 $etd_x$  is marked as “worldwide access” or  
 $etd_x$  is marked as “VT only” and  $ac_y \in VT_{cmm}$ , where  $VT_{cmm}$  is the community of Virginia Tech ID holders accessing  $z$  through a computer with a Virginia Tech registered IP address.  
 0, otherwise.

Table ?? shows a partial view of the number of unrestricted (worldwide, accessibility = 1 to everybody), restricted to VT campus (accessibility = 0 worldwide, 1 to members of  $VT_{cmm}$ ), mixed, along with the degree of accessibility  $acc(etd_x, ac_y)$  of the mixed ETDs for non- $VT_{cmm}$  members  $ac_y$ , as of March 25, 2003. For example, five out of the six chapters (structured streams) of the third mixed ETD under the letter A were available only to VT. The rights management rule therefore is 0 for all those chapters, thus making its overall accessibility to non-VT actors 1/6 or 0.167. Note that accessibility for the Virginia Tech ETDs has improved since 2003; indicators like this may be of help for those who work on collection development policies.

**Table 3.3:** Accessibility of VT-ETDs (first column corresponds to the first letter of author’s name)

First letter of author’s name	Unrestricted	Restricted	Mixed	Degree of accessibility for users not in the VT community
A	164	50	5	mix(0.5, 0.5, 0.167, 0.1875, 0.6)
B	286	102	3	mix(0.5, 0.5, 0.13)
C	231	108	7	mix(0.11, 0.5, 0.5, 0.5, 0.33, 0.09, 0.33)
D	159	54	2	mix(0.875, 0.666)
E	67	26	1	mix(0.5)

#### 3.5.2 PERTINENCE

Pertinence is one of the most “social” quality indicators since it is a relation between the information carried by a digital object and an actor’s information need. It depends heavily on the actor’s knowledge, background, current task, etc.

### 94 3. EVALUATION

Let  $Inf(do_i)$  represent the “information”<sup>6</sup> (not physical) carried by a digital object  $do_i$  in any of its components,  $IN(ac_j)$  be the information need<sup>7</sup> of an actor  $ac_j$ , and  $Context(ac_j, k)$  be an amalgam of societal factors that affect the judgment of pertinence of  $do_i$  by  $ac_j$  at time  $k$ . These include, among others, task, time, place, the actor’s history of interaction, and a range of other factors that are not given explicitly but are implicit in the interaction and ambient environment. A complete formalization of context is out of the scope of this work. The reader is referred to a workshop on “Context in Information Retrieval” for a number of papers on the subject [303].

Also, we define, for future reference, two time dependent sub-communities of actors,  $users$ , and  $external-judges \subset Ac$ , as:

- $users$ : set of actors with an information need who use DL services to try to fulfill/satisfy that need,
- $external-judges$ : set of actors responsible for determining the relevance (see Section 3.4) of a document to a query. We also assume that an external-judge can not be assigned to judge the relevance of a document to a query representing her own information need, i.e., at each point in time  $users \cap external-judges = \emptyset$ .

The pertinence of a digital object  $do_i$  to a user  $ac_j$  at a time  $k$  is an indicator function<sup>8</sup>  $Pertinence(do_i, ac_j, k) : Inf(do_i) \times IN(ac_j) \times Context(ac_j, k)$  defined as:

- 1, if  $Inf(do_i)$  is judged by  $ac_j$  to be informative with regards to  $IN(ac_j)$  in context  $Context(ac_j, k)$ ;
- 0, otherwise.

Since pertinence is a subjective judgment made by a *user* in a particular context it can ultimately only be assessed by the user herself.

#### 3.5.3 PRESERVABILITY

Preservability is a quality property of a digital object that reflects a state of the object that can vary due to changes in hardware (e.g., new recording technologies), software (e.g., release of a new version of the software used to create/display the object), representation formats (e.g., new image standard such as JPEG 2000), and processes to which the object is submitted (e.g., migration).

There are four main technical approaches to digital preservation:

<sup>6</sup>Information and information need, by themselves, are hard notions to formally define. One comprehensive attempt is presented in [435].

<sup>7</sup>Certain authors such as Taylor [628] and Mizzaro [436] make a distinction between the “real” and the “perceived” information need. We will not make this distinction here, in the interest of brevity.

<sup>8</sup>We agree with Voorhees [652], Greisdorf [264], and others who argue for non-binary pertinence/relevance functions, but such is not normal practice. We will leave extensions to our definitions for these cases for future work.

1. Migration: transforming from one digital format to another format, normally a successive subsequent one (e.g., from JPEG to JPEG 2000) [133].
2. Emulation: re-creating the original operating environment by saving the original programs and or creating new programs that can emulate the old environment [552].
3. Wrapping: packaging the object to be preserved with enough human readable metadata to allow it to be decoded in the future [664].
4. Refreshing: copying the stream of bits from one location to another, whether the physical medium is the same or not [383].

Note that here we are not considering physical deterioration of the medium in which the object is stored, since this is a property of the medium itself, not the object. However, we acknowledge that this is an important problem, for which “refreshing” is the normally used approach.

For cost, operational, and technical reasons, migration is the most widely used of the three techniques mentioned above [664]. However, the ideal solution should be some combination of all the techniques [664, 300]. One example that applies such a combination is the UVC-based approach [398]. Nonetheless, for the purpose of the discussion below, we will concentrate on migration issues.

A digital object’s preservability can be affected by its obsolescence and the fidelity of the migration process (see Figure 3.1). Obsolescence reflects the fact that a very obsolete object is really hard and costly to migrate, given the difficulty of finding appropriate migration tools and the right expertise. Fidelity reflects the differences between the original and the migrated object or, in other words, reflects the distortion or the loss of information inherent in the migration process that is absorbed by the object. The more obsolete the object, and the less faithful the migration process, the lower the object’s preservability. Preservability also is affected by contextual issues of specific DLs. For example, while it is desirable to always use the most faithful migration process, a DL custodian may not have sufficient resources (money, storage, personnel) to apply that process to its digital objects during migration. Based on the above discussion and on the fact that these two factors are orthogonal, we can define the preservability of a digital object  $do_i$  in a digital library  $dl$  as a tuple:

$$\text{preservability}(do_i, dl) = (\text{fidelity of migrating}(do_i, \text{format}_x, \text{format}_y), \text{obsolescence}(do_i, dl)). \quad (3.1)$$

As mentioned before, obsolescence is a complex notion to capture, depending on many contextual factors. Since the choice of how to deal with obsolescence generally depends on resources at the disposal of the DL custodian, one possible idea is to approximate its value by using the actual cost of migrating the object [571]. While a complete cost model for

**96 3. EVALUATION**

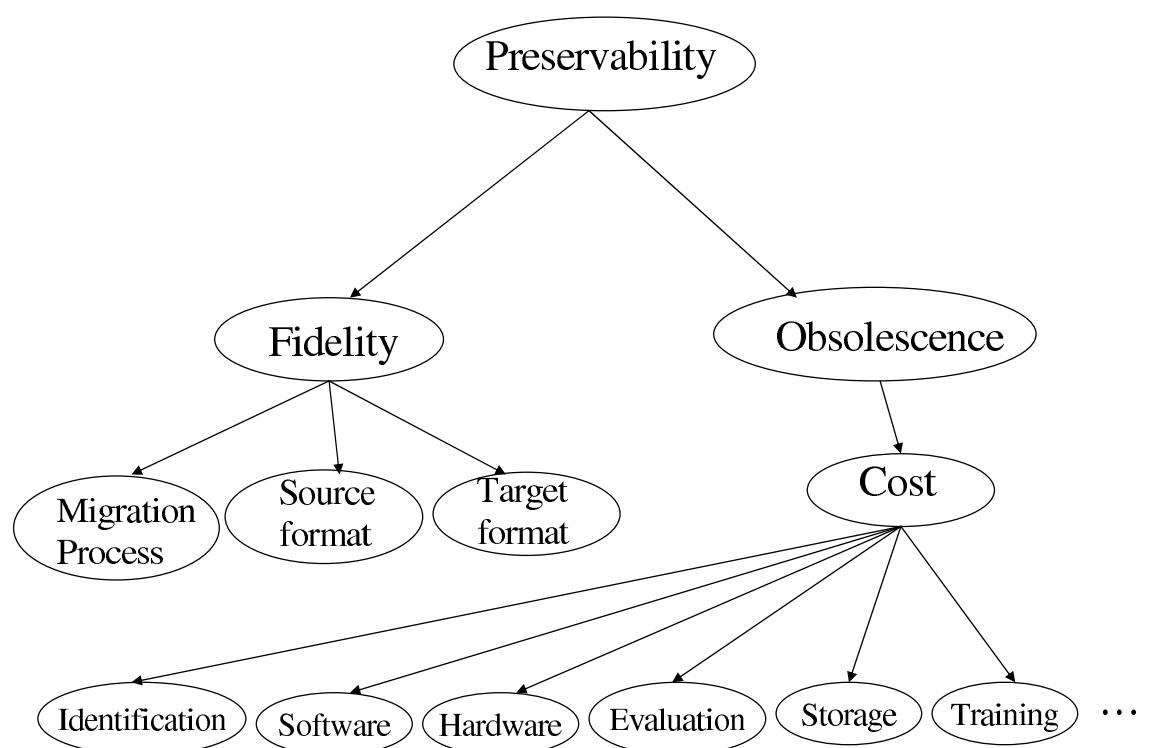


Figure 3.1: Factors in preservability (all links should be assumed to have “depends on” as their labels)

preservability/obsolescence is beyond the scope of this work, we recognize many factors that can affect the cost, including:

- capital direct costs—
  - software development/acquiring or license updating for newer versions,
  - hardware (for preservation processing or storage);
- indirect operating costs—
  - identifying candidate digital objects,
  - evaluating/examining/negotiating intellectual property issues and rights,
  - storage, and
  - staff training (on software / procedures).

Obsolescence then can be defined as  $\text{obsolescence}(do_i, dl) = \text{cost of converting/ migrating the digital object } do_i \text{ within the context of the specific digital library } dl$ .

The fidelity of the migration process  $p$  of a digital object  $do_i$  from  $format_x$  to  $format_y$  can be defined based on the inverse of the distortion or noise introduced by the migration process  $mp$ , i.e.,

$$\text{fidelity}(do_i, format_x, format_y) = \frac{1}{\text{distortion}(mp(format_x, format_y)) + 1.0}.$$

Distortion can be computed in a number of ways depending on the type of object and transformation [583]. One very common measure, when converting between similar formats, is the *mean squared error (mse)*. In the case of a digital object, *mse* can be defined as follows. Let  $\{x_n\}$  be a stream of a digital object  $do_i$  and  $\{y_n\}$  be the converted/migrated stream; the mean squared error  $\text{mse}(\{x_n\}, \{y_n\}) = \frac{1}{N} * \sum_{n=1}^N (x_n - y_n)^2$ , where  $N$  is the size of each stream. The average mean square error for the whole object  $do_i$  can be calculated as the average of *mse* for all its streams. This assumes that the other components (graphs and fuctions) of a digital object will be converted exactly.

**Example of Use.** Let us consider the following scenario adapted from [300]. In 2004, a librarian receives an email notifying her that a special collection <sup>9</sup> of 1,000 digital images, stored in TIFF version 5.0, is in danger of becoming obsolete, due to the fact that the latest version of the display software no longer supports TIFF 5.0. The librarian decides to migrate all digital photos to JPEG 2000, which now has become the *de facto* image preservation standard, recommended by the Research Libraries Group (RLG) [300].

The librarian does a small search for possible migration options and finds a tool, costing \$500, which converts TIFF 5.0 directly to JPEG 2000. Let us consider that the

<sup>9</sup>Preservation of a collection, instead of a digital object, also may involve preserving all the structures (e.g., classification schemes, etc.) imposed on the collection.

### 98 3. EVALUATION

amount of time taken by the librarian and the system administrator to order the software, install it, learn it, and apply it to all digital images combined takes 20 hours. Assume also that the hourly rate in this library is \$66.60 per hour per employee <sup>10</sup>. In order to save space, the librarian chooses to use in the migration a compression rate which produces an average *mse* of 8 per image. In this scenario, the preservability of each digital image would correspond to: preservability (image-TIFF 5.0, *dl*) =  $(1/9, (\$500 + \$66.60 * 20) / 1000) = (0.11, \$1.83)$ .

#### 3.5.4 RELEVANCE

A digital object is *relevant* [577] in the context of an expression of an information need (e.g., a query) or interest (e.g., profile) and a service (e.g., searching, recommending). A role of an information satisfaction service is to provide ways to find the most relevant information for the user, which in the case of DLs is carried by digital objects and their metadata specifications.

The relevance of a digital object to a query is an indicator function *Relevance*(*do<sub>i</sub>*, *q*) defined as:

- 1, if *do<sub>i</sub>* is judged by an *external-judge* to be relevant to *q*;
- 0, otherwise.

The most common measures for relevance estimates/predictions are based on statistical properties of the streams of the digital object and the queries. For example, in the vector space model, this relevance is estimated based on the distance between the vectors representing the objects and queries (as measured by the angle between them), and the components of these vectors are derived from values such as frequency of a term in a document, the frequency of the term in the collection, document size, document structure, query size, collection size, etc. Note that, in contrast to pertinence, relevance is a relation between a *representation* of a document and a *representation* of an information need (i.e., query). Also, it is supposed to be an objective, public, and social notion that can be established by a general consensus in the field, not a subjective, private judgment between the actor and her information need [193, 334].

The distinction we have made between pertinence and relevance is derived from a view held by part of the information science community [128, 578, 577, 193, 334]. We have just formalized the two notions in the context of our framework. In Saracevic's work, for example, relevance, as defined by us, is called systemic or algorithmic relevance, and is a relationship between a text and a query. Pertinence, or cognitive relevance, is a relationship between the state of knowledge and cognitive information need of a user and the objects

<sup>10</sup>1800 is the number of hours in a work-year (37.5 hrs/wk \* 48 wks/yr) and \$110,000 the total annual cost of an employee working for this DL, based on salary, benefits, expenses.

retrieved. Cognitive correspondence, informativeness, novelty, information quality, and the like are criteria by which cognitive relevance is inferred.

The external judges should evaluate the relevance of the object to the query without the cognitive load resulting from contextual interference, therefore their judgments should be more objective and more generally applicable.

### 3.5.5 SIGNIFICANCE

Significance of a digital object can be viewed from two perspectives: (1) relative to its pertinence or relevance or (2) in absolute terms, irrespective of particular user requirements. Absolute significance can be calculated based on *raw citedness*—the number of times a document  $do_i$  is cited, or the frequency of occurrence of citations whose target is  $do_i$ . Other factors may play a role in the significance of a document such as the prestige of the journal publishing the work, its use in courses, awards given, etc., but these are very hard to quantify/measure.

**Example of Use.** We used 98,000 documents from the ACM Digital Library, which corresponded to approximately 1,093,700 (outgoing) citations (average of 11.53 citations per document). Table ?? shows the top five documents in the ACM collection with the highest values of significance.

Table 3.4: Documents with highest degree of significance

Document	Publication	Year	Significance
Computer programming as art	CACM	1974	279
A generalized processor sharing approach to flow control in integrated services networks: the single-node case	IEEE/ACM Transactions on Networking (TON)	1993	138
The entity-relationship model – toward a unified view of data	ACM Transactions on Database Systems	1976	130
A relational model of data for large shared data banks	CACM	1970	121
Revised report on the algorithmic language scheme	ACM SIGPLAN Notices	1986	116

Notice that significance, as defined, is supposed to increase with time, as more people take notice of the work and acknowledge it through citations. As such, publication date affects this indicator (and timeliness, see below, as well) since older publications have more chance of being cited.

### 3.5.6 SIMILARITY

Similarity metrics reflect the relatedness between two or more digital objects. An object similar to another relevant or pertinent object has a good chance of also having these properties, but an object *too* similar to another supposedly different object can reveal a lack of quality (e.g., plagiarism, which might be found through plagiarism software) unless it is a variant version which can be identified through a de-duping process.

### 100 3. EVALUATION

Similarity can be measured based on the digital object's content (streams) (e.g., use and frequency of words), digital object's internal organization (structures), or patterns of citations/links. For example, similarity between two documents can be calculated using the cosine distance between the vectors representing the documents [37]. This idea can be expanded to calculate similarity between corresponding structured streams of documents (e.g., using their title and abstract texts). Other measures, such as “bag-of-words” and Okapi can be used to calculate similarity as well [37].

Similarity measures also may use link or citation information to compute the relatedness of two objects. Among the most popular citation-based measures of similarity are: co-citation [602], bibliographic coupling [338], and the Amsler measure. The last one is a combination of the previous two, so we will explain only the Amsler measure [20].

According to Amsler, two documents  $d_i$  and  $d_j$  are related if (1)  $d_i$  and  $d_j$  are cited by the same document, (2)  $d_i$  and  $d_j$  cite the same document, or (3)  $d_i$  cites a third document  $d_k$  that cites  $d_j$ . Thus, let  $Pd_i$  be the set of parents of  $d_i$ , and let  $Cd_i$  be the set of children of  $d_i$ . The Amsler similarity between two pages  $d_i$  and  $d_j$  is defined as:

$$Amsler(d_i, d_j) = \frac{|(Pd_i \cup Cd_i) \cap (Pd_j \cup Cd_j)|}{\max(|Pd_i \cup Cd_i|, |Pd_j \cup Cd_j|)}. \quad (3.2)$$

Eq. 3.2 tells us that, the more links (either parents or children)  $d_i$  and  $d_j$  have in common, the more they are related. The absolute Amsler degree of a document  $d_i$  in collection  $C$  is defined as  $\sum_{d_j \in C - \{d_i\}} Amsler(d_i, d_j)$ .

**Example of use.** Table ?? shows the top five documents in the ACM collection we studied with the highest absolute values of Amsler.

Table 3.5: Documents with highest absolute Amsler degree

Document	Publication	Year	Amsler
Computer programming as an art	CACM	1974	69.15
Compiler transformations for high-performance computing	CSUR	1994	64.31
Analysis of pointers and structures	Prog. language design and implementation	1990	62.56
Query evaluation techniques for large databases	CSUR	1993	59.81
A schema for interprocedural modification side-effect analysis with pointer aliasing	TOPLAS	2001	57.90

#### 3.5.7 TIMELINESS

Timeliness of a digital object is the extent to which it is sufficiently up-to-date for the task at hand [517]. It can be a function of the time when the digital object was created, stored, accessed, or cited.

### 3.6. METADATA SPECIFICATIONS AND METADATA FORMAT 101

Since the timeliness of an object is directly related to the information it carries, which can still be timely even if the object is “old”, a good quality indicator of this quality dimension is the time of the latest citation, since it’s a measure that:

1. captures the fact that the information carried by the object is still relevant by the time the citing object was published;
2. is independent from the actor that receives the object and the time the object is delivered; and
3. reflects the overall importance of the object inside its community of interest.

As it is known that many documents are never cited, an alternative is to consider the age of the object itself. Therefore the timeliness of a digital object  $do_i$  can be defined as:

- (current time or time of last freshening) - (time of the latest citation), if object is ever cited, otherwise as
- (current time or time of last freshening) - (creation time or publication time), if object is never cited.

Time of last freshening, which is defined as the time of the creation/publication of the most recent object in the collection to which  $do_i$  belongs, may be used instead of current time if the collection is not updated frequently.

**Example of use.** Figure 3.2 shows the distribution of timeliness (0 through 10) for documents in the ACM DL with citations. Time of last freshness is 2002. It can be seen, discounting the first set of values (timeliness=0), that there is an inverse relation between timeliness and the size of the set of documents with that value: the smaller the value, the bigger the set, meaning that as time passes there is less chance that a document will be cited.

## 3.6 METADATA SPECIFICATIONS AND METADATA FORMAT

Three main dimensions of quality can be associated with metadata specifications and metadata formats: accuracy, completeness, and conformance.

### 3.6.1 ACCURACY

Accuracy is defined in terms of properties of a metadata specification for a digital object. Accuracy of a triple  $(r, p, v)$  (i.e.,  $(resource, property^{11}, value)$ ) refers to the nearness of

<sup>11</sup>In this chapter we will use the terms ‘metadata property’, ‘metadata attribute’, and ‘metadata field’ interchangeably.

102 3. EVALUATION

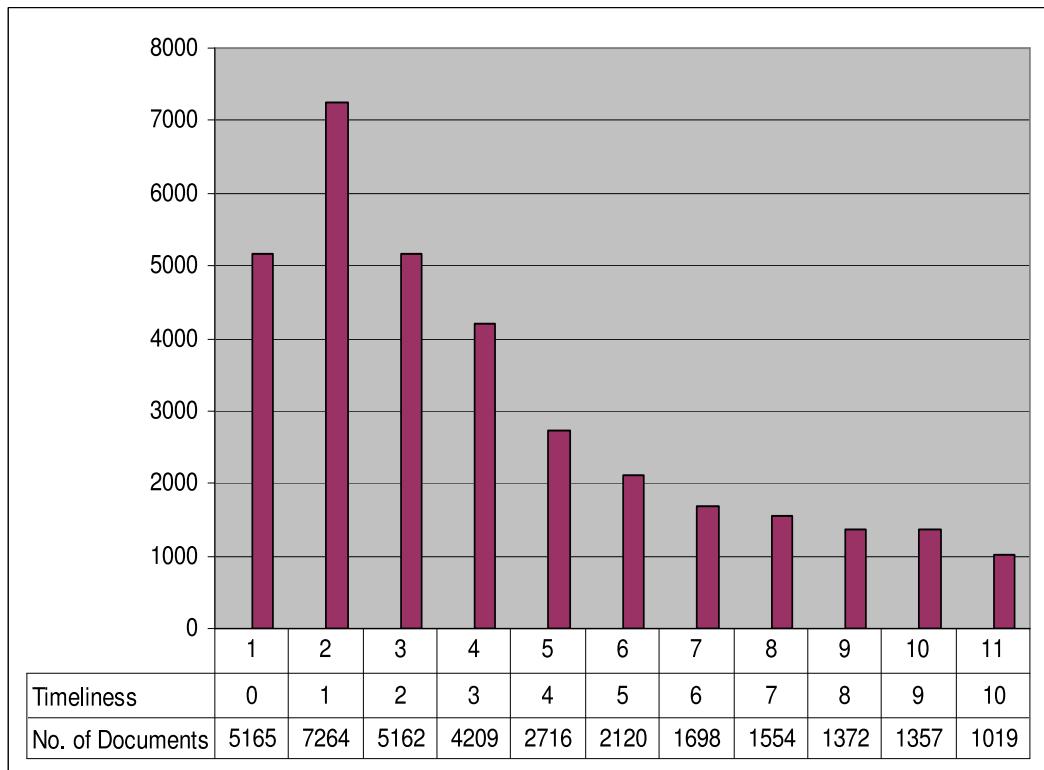


Figure 3.2: Timeliness in the ACM Digital Library

the value  $v$  to some value  $v'$  in the attribute range that is considered the correct one for the (resource, property) pair  $(r, p)$  [542]. Notice that in 5S, a metadata specification is defined as a structure  $(G, L, F)$ ,  $G$  being a graph,  $L$  a set of labels, and  $F$  a labelling function associating components (i.e., nodes and vertices) of the graph with labels. In other words, a metadata specification can be seen as a labeled digraph. The triple  $st = (F(v_i), F(e), F(v_j))$  is called a statement (derived from the descriptive metadata specification), meaning that the resource labeled  $F(v_i)$  has property or attribute  $F(e)$  with value  $F(v_j)$ . A metadata specification for a digital object is completely accurate with respect to a digital object if all the (derived) triples are accurate, assuming some appropriate accuracy threshold. The degree of accuracy of triple  $(r, p, v)$  can be defined as an indicator function or with specific rules for a particular schema/catalog. It is dependent on several factors, including the attribute's range of values  $v$ , intended use, etc. Examples are given below. Thus, the degree of accuracy  $acc(ms_x)$  of a metadata specification  $ms_x$  can be defined as

$$acc(ms_x) = \frac{\sum_{\forall(r,p,v) \text{ from } ms_x} \text{degree of accuracy of } (r, p, v)}{\text{total number of triples } (r, p, v) \text{ from } ms_x} \quad (3.3)$$

**Example of Use.** To illustrate the application of such an indicator we used OCLC's NDLTD Union Catalog. We chose OCLC's NDLTD Union Catalog because of its numerous problems regarding metadata accuracy, observed while creating a collection for filtering experiments [692]. For example, author information is very commonly found in the title field ("The concept of the church in the thought of Rudolf Bultmann – by Paul E. Stambach.") and sometimes the abstract contains all kinds of information (see below) but not the thesis/dissertation's summary. We defined the following rules for the dc.author<sup>12</sup>, dc.title, and dc.abstract fields.

- Degree of accuracy of  $(*, dc.title, *)$  for OCLC's NDLTD Union Catalog = 1, if dc.title does not contain author information; 0.5 otherwise. In case it is empty or null it receives a 0 (zero) value.
- Degree of accuracy of  $(*, dc.abstract, *)$  = 1 if the field corresponds to the thesis or dissertation's summary; 0 otherwise. The decision of whether a dc.abstract field corresponds to a summary or not was based on the size of the text and a number of heuristics. For example, 1) if dc.abstract is equal to "Thesis" or "Dissertation", it is not a summary; 2) if dc.abstract contains phrases like "Title from \*" (e.g., "Title from first page of PDF file"), "Document formatted into pages", "Includes bibliographical references", "Mode of access", among others, it is not a summary.

According to these two rules the average OCLC accuracy for all its metadata records (approximately 14,000 records, in September 2003<sup>13</sup>) was calculated as around 0.79, assuming a maximum of 1.

<sup>12</sup>The author field in the Dublin Core standard.

<sup>13</sup>Over 250K in Nov. 2006, and over 1.8M in Nov. 2010

## 104 3. EVALUATION

### 3.6.2 COMPLETENESS

Completeness is a pervasive quality dimension that is associated with many of the DL concepts. The general notion of completeness can be defined as: (number of units of a particular concept) / (ideal number of units of that concept). This notion can be adapted or instantiated to specific DL concepts.

Completeness of metadata specifications refers to the degree to which values are present in the description, according to a metadata standard. As far as an individual property is concerned, only two situations are possible: either a value is assigned to the property in question, or not. The degree of completeness of a metadata specification  $ms_x$  can be defined as<sup>14</sup>

$$Completeness(ms_x) = 1 - \frac{\text{no. of missing attributes in } ms_x}{\text{total no. of attributes in the schema for } ms_x} \quad (3.4)$$

Notice that the assumption here is that the more complete, the better. However, we acknowledge that there can be situations, for example, determined on purpose in accordance with local needs, in which this is not always true.

**Example of application.** Figure 3.3 shows the average of completeness of all metadata specifications (records) in catalogs of the NDLTD Union Archive administered by OCLC as of February 23, 2004, relative to the Dublin Core metadata standard (15 attributes).

### 3.6.3 CONFORMANCE

The conformance of a metadata specification to a metadata standard/format/schema has been formally defined in Section 1.9. In that definition a value of an attribute is conformant to its schema if it has the data type of the attribute (e.g., string, date, number, etc.). That definition can be extended to include cardinality (i.e., considering mandatory/optional fields) and multiplicity (i.e., considering repeatable fields) issues.

A metadata specification  $ms_x$  is *cardinally conformant* to a metadata format if:

1. it conforms with its schema in terms of the data types of its attributes according to Definition 14 of metadata schema in Section 1.9;
2. each attribute  $att_{xy}$  of  $ms_x$  appears at least once if  $att_{xy}$  is marked as mandatory in the schema; and
3.  $att_{xy}$  does not appear more than once if it is not marked as repeatable in the schema.

From now on, we will use conformance to refer to the stronger definition of *cardinally conformant*. Different from completeness, an attribute may be missing in a metadata

<sup>14</sup>According to the definition of completeness in [254]

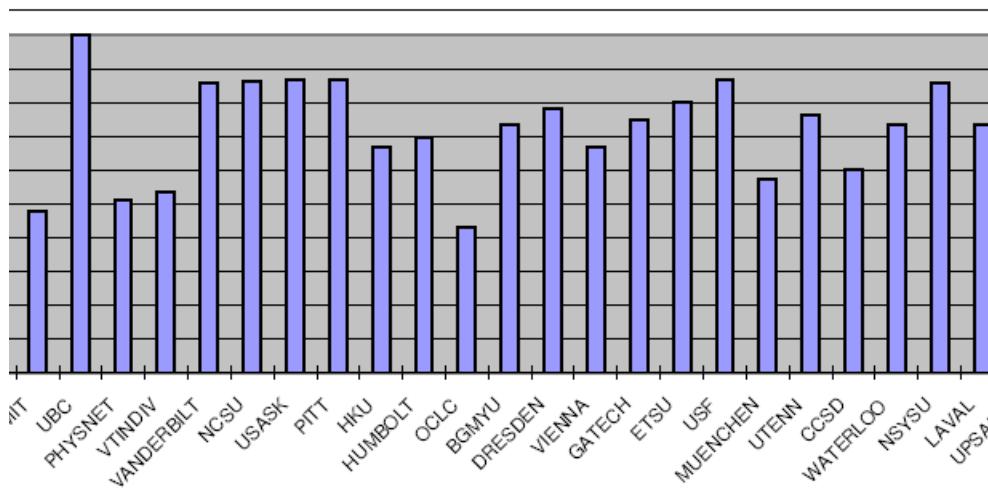


Figure 3.3: Average completeness of catalogs in NDLTD (as of February 2004)

### 106 3. EVALUATION

specification, but the attribute still can be considered conformant, if it is not marked as mandatory in the mandatory schema. The degree of conformance of a metadata specification  $ms_x$  can be defined as

$$\text{Conformance}(ms_x) = \frac{\sum_{\substack{\text{attributes } att_{xy} \text{ in schema } ms_x}} \text{degree of conformance of } att_{xy}}{\text{total number of attributes in the schema for } ms_x} \quad (3.5)$$

The degree of conformance of  $att_{xy}$  is an indicator function defined as 1 if  $att_{xy}$  obeys all conditions specified in the above definition; 0 otherwise.

**Example of use.** Figure 3.4 shows the average conformance of the metadata records in the catalogs of the NDLTD Union Archive, relative to the ETD-MS metadata standard for electronic theses and dissertations<sup>15</sup>. ETD-MS, different from the Dublin Core in which all fields are optional, defines six mandatory fields: dc.title, dc.creator, dc.subject, dc.date, dc.type, dc.identifier. Also the range for the dc.type is defined as the set {‘Collection’, ‘Dataset’, ‘Event’, ‘Image’, ‘InteractiveResource’, ‘Software’, ‘Sound’, ‘Text’, ‘PhysicalObject’, ‘StillImage’, ‘MovingImage’, ‘Electronic Thesis or Dissertation’}. If any value other than these words/phrases is used for the attribute, it is defined as non-conformant.

## 3.7 COLLECTION, METADATA CATALOG, AND REPOSITORY

### 3.7.1 COLLECTION COMPLETENESS

A complete DL collection is one which contains all the existing digital objects that it should hold. Measuring completeness of a collection can be extremely hard or virtually impossible in many cases when there is no way to determine the ideal real-world collection such as in the Web or in hidden databases. Advanced judicious sampling or probing of alternative repositories whose completeness has been established manually can give crude estimates [309]. An example could be to approximate a measure of the completeness of a computer science collection of papers on a specific topic by sampling the ACM or IEEE-CS digital libraries, DBLP, and some other commercial publishers’ on-line databases. In other cases such as for harvested or mirrored collections those estimates are easier to establish. More formally, Completeness( $C_x$ ) of a collection  $C_x$ , can be defined as the ratio between the size of  $C_x$  and the ideal real-world collection, i.e.,

$$\text{Completeness}(C_x) = \frac{|C_x|}{|\text{ideal collection}|}. \quad (3.6)$$

**Example of use.** The ACM Guide is a collection of bibliographic references and abstracts of works published by ACM and other publishers. The Guide can be considered

<sup>15</sup><http://www.ndltd.org/standards/metadata/current.html>

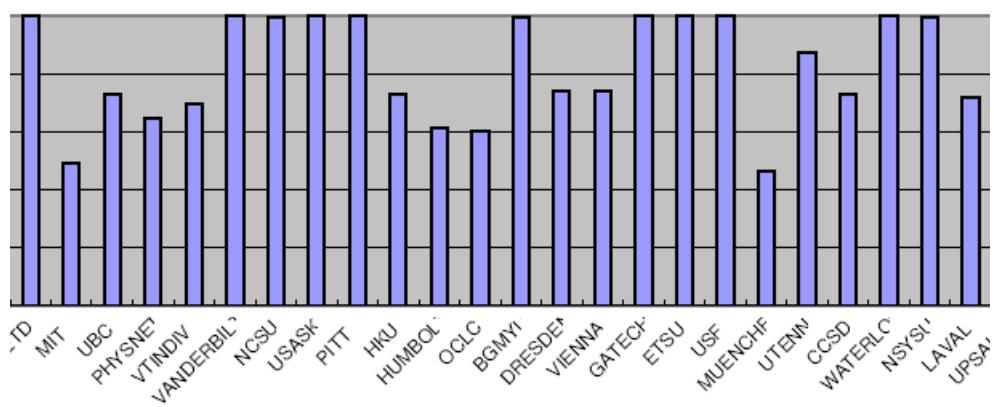


Figure 3.4: Average conformance of catalogs in NDLTD

### 108 3. EVALUATION

a good approximation of an ideal computing collection for a number of reasons including the fact that it contains most of the different types of computing-related literature and for each type it can be considered fairly complete. For example, the set of theses in the Guide comes from Proquest-UMI, which receives copies of almost all dissertations defended in the USA or Canada; the number of technical reports is close to that of NCSTRL (<http://www.ncstrl.org>), the largest repository of CS technical reports, and it contains large numbers of records from many of the most important publishers in computer science (such as ACM, IEEE, Springer, Elsevier, etc). Table ?? shows the degree of completeness of several CS-related collections<sup>16</sup> when compared with the Guide.

**Table 3.6:** Completeness of several collections

Collection	Degree of Completeness
ACM Guide	1
DBLP	0.652
CITIDEL(DBLP(partial) + ACM(partial) + NCSTRL + NDLTD-CS)	0.467
IEEE-DL	0.168
ACM-DL	0.146

#### 3.7.2 CATALOG COMPLETENESS AND CONSISTENCY

The degree of completeness of a catalog  $DM_C$  for a collection  $C$  can be defined accordingly as

$$\text{Completeness}(DM_C) = 1 - \frac{\text{no. of } do's \in C \text{ without a metadata specification}}{\text{size of the collection } C}. \quad (3.7)$$

Since each object is unique by nature (e.g., each has a unique global handle) two different objects should not have the same metadata description. A catalog in which this occurs is therefore considered inconsistent. It should be noticed, though, that an object can have more than one metadata specification (e.g., a Dublin Core and a MARC one).

Consistency, accordingly, is an indicator function defined as

- 0, if there is at least one set of metadata specifications assigned to more than one digital object;
- 1, otherwise.

**Example of Use.** In April 2004, the NDLTD Union catalog administered by OCLC tried to harvest data from the Brazilian Digital Library of Electronic Theses and Dissertations (BDTD). Because of problems in BDTD's implementation of the OAI protocol and problems with the Latin character set handling by OCLC, only 103 records were harvested

<sup>16</sup>All of which are subsets of the Guide. Size of the Guide = 735,429 (as of March, 2004)

from the repository. The BD TD collection contained 4446 records. Therefore the completeness of the harvested catalog for BD TD in OCLC would be completeness(BD TD in OCLC Union Catalog) = 1 - (4446 - 103)/4446 = 0.023. Note that completeness significantly improved by 2006.

### 3.7.3 REPOSITORY COMPLETENESS AND CONSISTENCY

A repository is complete if it contains all collections it should have. The degree of completeness of a repository  $R$  is defined as

$$\text{Completeness}(R) = \frac{\text{number of collections in the repository}}{\text{ideal number of collections}}. \quad (3.8)$$

If the repository *stores* collections with their respective metadata catalogs, its consistency can be defined in terms of these two components. Therefore, repository consistency is an indicator function defined as

- 1, if the consistency of all catalogs with respect to their described collections is 1;
- 0, otherwise.

**Example of use.** We will use the ACM Guide as the ideal collection. Not considering the Bibliography and Play subcollections of the Guide and considering each publisher as a different subcollection, the completeness of CITIDEL can be calculated as 4 (ACM + IEEE + NCTRL + ND LTD-CS) / 11 (total number of collections) or 0.36.

## 3.8 DL SERVICES

Dimensions of quality for DL services can be classified as external or internal [656]. The external view is related to information satisfaction services and is concerned with the use and perceived value of these services from the point of view of societies of end users. The internal view addresses the construction and operation necessary to attain the required functionality given a set of requirements that reflect the external view. Issues in system construction, operation, design, and implementation should be considered here.

### 3.8.1 EFFECTIVENESS AND EFFICIENCY

The two most obvious external quality indicators of DL services, as perceived by end users, are efficiency and effectiveness. Efficiency is most commonly measured in terms of speed, i.e., the difference between request and response time. More formally, let  $t(e)$  be the time of an event  $e$ , and let  $e_{ix}$  and  $e_{fx}$  be the initial and the final events of scenario  $sc_x$  in service  $Se$ . The efficiency of service  $Se$  is defined as

$$\text{Efficiency}(Se) = \frac{1}{\max_{sc_x \in Se} (t(e_{fx}) - t(e_{ix})) + 1.0}. \quad (3.9)$$

### 110 3. EVALUATION

Effectiveness is normally related to information satisfaction services and can be measured by batch experiments with test collections or through experiments with real users. Different types of information services can use different metrics, the most common ones being precision and recall [37], extensively used to assess quality of searching or filtering services.

#### 3.8.2 EXTENSIBILITY AND REUSABILITY

Regarding design and implementation of DL services, there are two main classes of quality properties: 1) those regarding composability of services; and 2) those regarding qualitative aspects of the models and implementations. The latter include issues such as completeness, consistency, correctness, and soundness. In this work we will concentrate on composability aspects but we acknowledge the importance and complexity of the latter issues.

Composability can be defined in terms of reusability and extensibility. In short, a service Y *reuses* a service X if the behavior of Y incorporates the behavior of X (in the sense that scenarios of X are also scenarios of Y). A service Y *extends* a service X if it subsumes the behavior of X and potentially includes additional conditional subflows of events (the scenarios of X are subsequences of the scenarios of Y). A composed service either extends or reuses another service. A composable service (i.e., a service that can be extended or reused) has to satisfy a number of requirements including exporting clear interfaces, providing mechanisms/protocols for connections and passing of parameters, offering gateway or mediator services to convert between disparate document formats and protocols, and satisfying circumstantial conditions such as satisfaction of any pre-condition based on the service's current state and input values to any called service. All of these make it very hard to quantify the composability of a service. However, even if an indicator of composability can be determined, a service is still only potentially reusable and extensible. One more pragmatic indicator of the actual composability is to ascertain from a set of services and service managers that run or implement those services, which managers are actually inherited from or included by others. Therefore given a set of services  $Serv$  and a set of service managers  $SM$  that run those services, two quality indicators of extensibility and reusability can be defined.

- Macro-Extensibility( $Serv$ ) =  $\frac{\sum_{Se_i \in Serv} extended(Se_i)}{|Serv|}$ , where  $Serv$  is the set of services of the DL and  $extended(Se_i)$  is an indicator function defined as

1, if  $\exists Se_j \in Serv : Se_j \text{ extends } Se_i$ ;

0, otherwise.

- Micro-Extensibility(Serv) =  $\frac{\sum_{sm_x \in SM, Se_i \in Serv} LOC(sm_x) * extended(Se_i)}{\sum_{sm \in SM} LOC(sm)}$ , where LOC corresponds to the number of lines of code of all operations of a service manager and  $sm_x$  runs  $Se_i$ .
- Since reuse/inclusion has a different semantics of extension, reusability can accordingly be defined as Macro-Reusability(Serv) =  $\frac{\sum_{Se_i \in Serv} reused(Se_i)}{|Serv|}$ , where  $reused(Se_i)$  is an indicator function defined as
  - 1, if  $\exists Se_j \in Serv : Se_j$  reuses  $Se_i$ ;
  - 0, otherwise.

- Micro-Reusability(Serv) =  $\frac{\sum_{sm_x \in SM, Se_i \in Serv} LOC(sm_x) * reused(Se_i)}{\sum_{sm \in SM} LOC(sm)}$ , where LOC corresponds to the number of lines of code of all operations of a service manager and  $sm_x$  runs  $Se_i$ .

**Example of use.** Table ?? shows the lines of code (LOC) needed to implement service managers that run several services in the ETANA archaeological digital library, in September, 2004 [537]. Let's assume a 1:1 ratio between the set of services and set of service managers. Reused services (and included service managers) are implemented as ODL components [614]. These services are searching, annotating, recommending, and (union) cataloging.

The wrapping services, the ones that really reuse and provide the services offered by the DL components, are necessary in order to deal with issues such as invoking operations, parsing results, and interfacing with other components (like the user interface). However, the additional code for those wrappers is only a very small percentage of the total lines of code required for implementing the components. In the ETANA-DL prototype (in September, 2004), only a few important services were componentized and therefore reused (Macro-Reusability(ETANA DL Services) =  $4/16 = 0.25$ ). However, Micro-Reusability =  $3630/11910 = 0.304$  makes it clear that we can re-use a very significant percentage of DL code by implementing common DL services as components. Moreover, as more service managers get componentized, more code and managers are potentially inherited from/included by more DLs.

### 3.8.3 RELIABILITY

Regarding operation, the most important quality criterion is reliability. Service reliability can be defined as the probability that the service will not fail during a given period of time

**112 3. EVALUATION**

**Table 3.7:** Analysis of ETANA DL prototype using the metric of Lines of Code

Service	Component Based	LOC for implementing service	Total LOC	LOC reused from component
Searching . Back-end	Yes	-	1650	1650
Search Wrapping	No	100	100	-
Recommending	Yes	-	700	700
Recommend Wrapping	No	200	200	-
Annotating . Back-end	Yes	50	600	600
Annotate Wrapping	No	50	50	-
Union Catalog	Yes	-	680	680
User Interface Service	No	1800	1600	-
Browsing	No	1390	1390	-
Comparing (objects)	No	650	650	-
Marking Items	No	550	550	-
Items of Interest	No	480	480	-
Recent Searches/Discussions	No	230	230	-
Collections Description	No	250	250	-
User Management	No	600	600	-
Framework Code	No	2000	2000	-
	Total	8280	11910	3630

[276]. We define the reliability of a service  $Se_x$  as

$$\text{Reliability}(Se_x) = 1 - \frac{\text{no. of failures}}{\text{no. of accesses}}. \quad (3.10)$$

A failure is characterized as an event that

1. was supposed to happen in a scenario but did not, or
2. did happen but did not execute some of its operations, or
3. did happen, where the operations were executed, but the results were not the correct ones.

**Example of use.** Table ?? shows reliability figures for the most popular services of CITIDEL, according to a log analysis done on April 1, 2004. The low reliability for the *structured searching* service can be explained by the fact that it was an experimental one, which ran only for a short period of time. However, entry points and links to this service were not removed after the experiments, and users kept using it without getting answers. This also shows how flaws in design can be found with such quality-oriented analysis.

**Table 3.8:** Reliability of CITIDEL services

CITIDEL service	No. of failures/No. of accesses	Reliability
searching	73/14370	0.994
browsing	4130/153369	0.973
requesting (getobject)	1569/318036	0.995
structured searching	214/752	0.66
contributing	0/980	1

### 3.9 CASE STUDY: 5SQUAL

Digital libraries also may present many differences when compared or analyzed over time. The available content can grow in size and diversity. The provided services may exhibit changes in their usage patterns, their internal organization may evolve, etc. However, in practice, most DL evaluations occur only when a problem or situation that requires urgent intervention occurs. Those evaluations are usually very specific, depending on the particularities of each system. Thus, in order to improve development, which in the case of DLs is generally very expensive and time-consuming [617, 614], and to promote maintenance of such dynamic systems, periodic and recurrent quality assessments of the DL components should be performed.

With this goal in mind, we designed, implemented, and evaluated 5SQual, a tool intended for *automatic quantitative evaluation* of some of the most important components of a digital library, namely, digital objects, metadata, and services. 5SQual is grounded in

### 114 3. EVALUATION

a formal quality model for digital libraries [259]. The tool helps to manage and maintain digital libraries through automatic and recurrent evaluations that can diagnose problems and suggest possible improvements in the system, as well as demonstrate its evolution over time. Due to the complexity, heterogeneity, and diversity of DLs in terms of content and services, the tool has been designed to be flexible enough to be used by many different systems.

The potential applicability and usefulness of the tool was tested by employing it for the evaluation of real DLs, such as *Virginia Tech's Digital Library of Electronic Theses and Dissertations* (VT-ETD)<sup>17</sup> and *The Brazilian Digital Library of Computing* (BDBComp)<sup>18</sup>. These evaluations generated information that, according to interviewed administrators, can be very useful to improve and maintain a DL. The tool and its evaluation serve also as a validation of the theoretical quality model for DLs presented in [259] and makes it possible for DL administrators and digital librarians to apply the model in real settings. We also performed a usability study of the 5SQual interface with usability specialists and conducted interviews with potential users (DL administrators). The results of both evaluations and the opinions expressed were in general very positive.

In sum, the main contributions of this work are: (i) the description of the design, architecture, implementation, and use of 5SQual, a tool for automatic quality assessment of digital libraries; (ii) the evaluation (with usability specialists) of its graphical interface specially designed to guide the configuration of 5SQual evaluations; and (iii) an analysis of the results of interviews discussing expectations regarding 5SQual, conducted with administrators of real DLs.

#### 3.9.1 5SQUAL OVERVIEW

The construction of 5SQual was initially based on the implementation of a subset of the quality dimensions presented in Section 3.4. These dimensions have been chosen for a first implementation of the tool because the respective numeric indicators are user independent and objective enough to allow an automatic evaluation. Other dimensions and numeric indicators can be added to the tool in the future.

##### The 5SQual Architecture

The 5SQual architecture was designed with the goal of allowing the tool to be used by a large number of diverse DLs with different goals (e.g., complete periodical evaluations, diagnosis of problems). Since these systems make the information necessary for evaluation available in many distinct ways, the architecture tries to be very flexible in several aspects including:

<sup>17</sup><http://scholar.lib.vt.edu/theses/>

<sup>18</sup><http://www.lbd.dcc.ufmg.br/bdbcomp/>

- Flexibility in *data collection*. Data for evaluation may be gathered from Web pages, from the DL repositories via the Open Archives Protocol for Metadata Harvesting [475] or from the local filesystem;
- Flexibility in *data extraction*. Since the log files of a DL may use disparate formatting rules, the 5SQual architecture allows the user to utilize internal recognizers that come with the tool, for example, for the XMLLog format [257, 260], or to indicate specific external recognizers for a particular format.
- Flexibility in *evaluation*. The tool allows the user to specify which set of dimensions she wants to evaluate.
- Flexibility in *utilization*. 5SQual receives as input an XML file with the parameters necessary for retrieving and extracting the data for the calculation of the dimensions defined in an evaluation. To facilitate the construction of this input file, a special graphical user interface was implemented to guide the user throughout this configuration process in order to generate the file and call the execution of the evaluation. It also is possible to generate only the configuration file and execute the evaluation later, via interface or via command line. The saved input file with all the configurations also may be re-used in posterior evaluations.

Figure 3.5 shows the 5SQual architecture. The necessary information for the evaluation resides in the DL and should be retrieved through the DL application layer (e.g., through an OAI interface). The 5SQual architecture is organized as follows:

- Processing Layer - In this layer we have three modules: the retrieval module, the extraction module and the calculation module.
  - Retrieval module: This module is responsible for obtaining the necessary information for evaluation on the Web or in the local file system. It collects log files that record the behavior of the DL services, its digital objects, or metadata with information about these objects. For retrieving metadata on the Web, it uses the OAI interface. Digital objects and logs can be retrieved from the Web or from the local file system, through previously indicated file paths.
  - Extraction module: 5SQual uses parsers that have been specified by the user or the ones that already come with the tool. These parsers extract data from the collected files and convert them to the 5SQual standard formats that describe the necessary information for each dimension. The set of built-in parsers includes content parsers (e.g., for PDF and PS files), specific metadata format parsers (e.g., for Dublin Core and RFC1807 formats), and specific log format parsers (e.g., for the XMLLog format [257, 260]).

116 3. EVALUATION

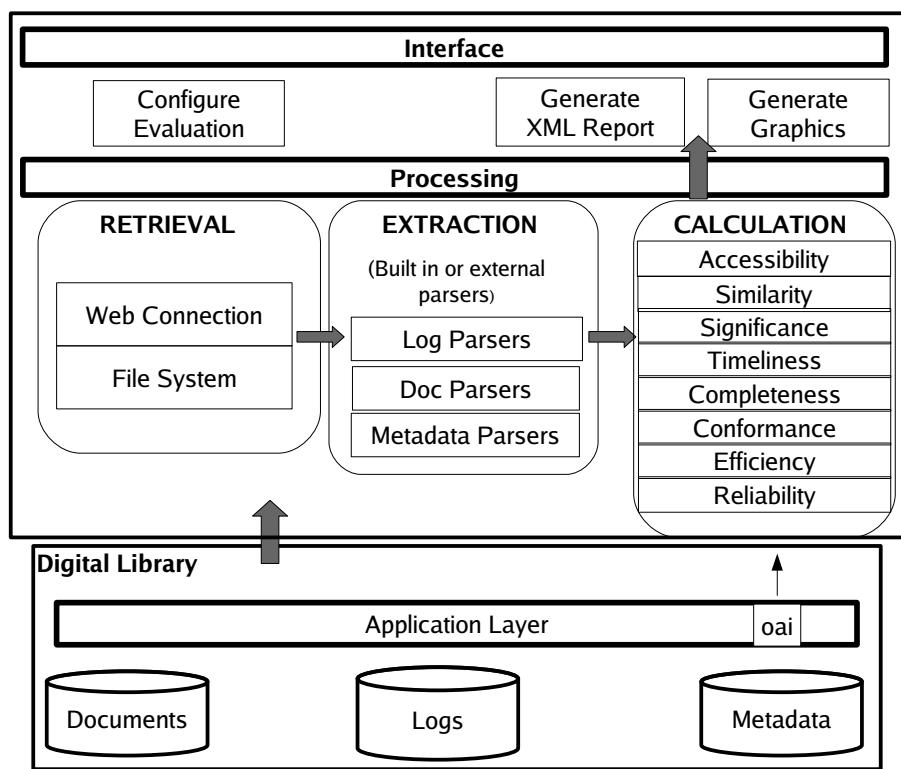


Figure 3.5: 5SQual Architecture

- The Calculation module: In this module, 5SQual implements a set of numeric indicators for each quality dimension.
- The Interface Layer - The configuration module is responsible for storing the parameters defined for the evaluation. According to the choices made by the user, 5SQual generates XML reports and charts for each dimension, considering the evaluation results.

### 5SQual Operation

Before starting with 5SQual, a user, typically the administrator of a DL, has to configure the parameters for the evaluation through an interface that was specially developed to help with this task. The 5SQual interface works like a setup wizard that guides the user through the necessary configuration steps, assuring that the mandatory parameters have been filled before undertaking an evaluation. An XML file with the configured parameters is generated and can be imported later through the same interface to repeat the evaluation, making it easier for the user to analyze the system over time.

The parameters indicate where 5SQual should find information for the evaluation and how to extract them to calculate the selected dimensions. Once the documents, metadata and any other necessary files are available, 5SQual extracts the required information. To accomplish this, 5SQual uses external programs specified by the user or the built-in parsers that come with the tool. Then, the extracted information is used to calculate numeric indicators for each dimension to be evaluated. In the following, we show a step-by-step configuration of an evaluation carried out using 5SQual.

The interface first presents to the user two options: (1) start a new evaluation from the beginning by following all the steps to configure the necessary parameters for this purpose or (2) import a previously generated file with all the parameters already specified (see Figure 3.6).

If the user chooses to fill the parameters through the interface, she is then asked to identify this evaluation by giving the name of the DL that is being assessed and adding an optional description (see Figure 3.7). This serves to facilitate re-use of this configuration in a next evaluation.

Now, the user must choose which quality dimensions to evaluate. The dimensions are selected from a set of *checkboxes* located in the left portion of the screen (see Figure 3.8). When a dimension is selected, the necessary resources for calculating the respective indicators are presented in the right portion of the screen. This is important to make the user aware of the resources that the DL must provide in order to be evaluated under that dimension. If this resource cannot be obtained, the dimension must be deselected. Another interesting aspect of the interface is the *help icons* (with the question marks) shown in front of the name of each dimension. If the user presses one of these icons, a definition of the dimension along with the explanation of its numerical indicators is presented.

**118 3. EVALUATION**



Figure 3.6: 5SQual Interface - Starting Configuration

### 3.9. CASE STUDY: 5SQUAL 119

... Identifying the Evaluation ...

Name of the Digital Library:

BDBComp

Description for this evaluation (optional):

Evaluation of the Brazilian Digital Library of Computing.  
Dimensions: Efficiency and Reliability.

previous next

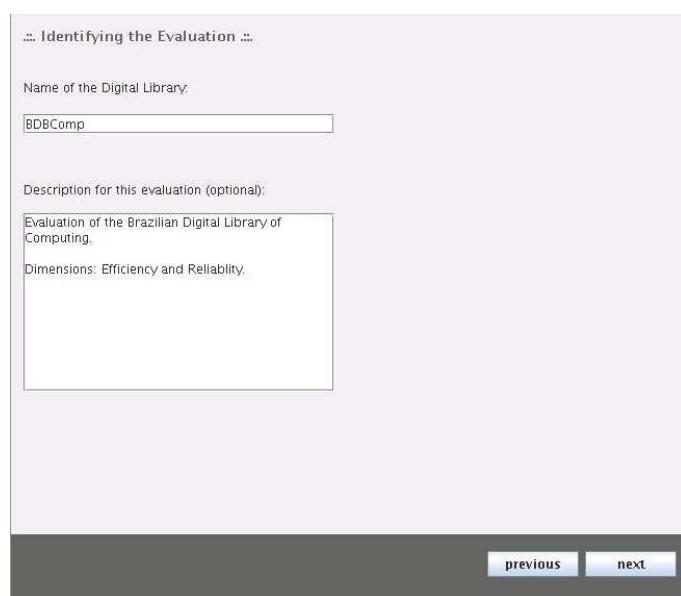


Figure 3.7: 5SQual Interface - Evaluation Identification

... Choose the Quality Dimensions to evaluate ...

Digital Objects

- Accessibility
- Significance
- Similarity
  - By content
  - By citations
- Timeliness
  - use log file
  - use metadata

Metadata

- Completeness
- Conformance

Services

- Efficiency
- Reliability

Necessary Resources

- Services executions times
- Services executions status

Resource Configuration

Inform a log file with information about the status of the services executions during a period of time:

Search for this file:  
 Locally  Remotely (web page)

Indicate the location of the file:

ok cancel

Configure Resource

previous next

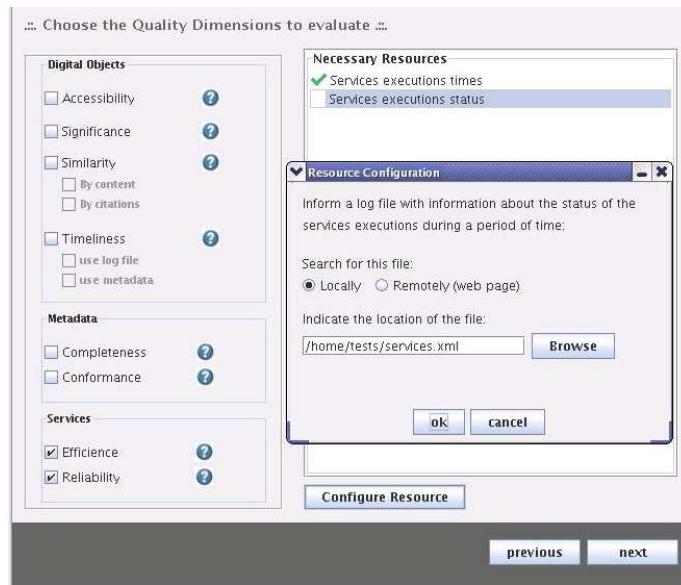


Figure 3.8: 5SQual Interface Selection of Dimensions and Indication of Resources

### 120 3. EVALUATION

After choosing all the dimensions that will be evaluated, the user must configure the resources that are shown in the list on the right. For such, she must select in the list a specific resource and press the button **Configure Resource** (see Figure 3.8). That done, a window requesting information about where to retrieve the chosen resource pops up. In Figure 3.8, a configuration window of the resource of the dimension *Reliability* requests the path for the file containing data about the status of several DL services executions, during a period of time. The user can choose to look for this file in the local file system or remotely, on the Web. This resource is mandatory and must be configured before the user advances to the next step.

The user then must specify parameters about how to extract the data from the indicated resources and how to calculate the indicators for each selected dimension. In Figure 3.9, the dimensions chosen in the previous step are shown on the left portion of the screen. Once a dimension is selected, the area on the right changes, presenting a panel that requests information about the parameters for the selected dimension. In Figure 3.9, the user chose the dimension *Reliability* and configured the necessary parameter: the recognizer program used to extract information from the log file containing data about the status of the executions of the DL services. In that case, the user specified a resource to be parsed using the 5SQual plugin for the XMLLog format.

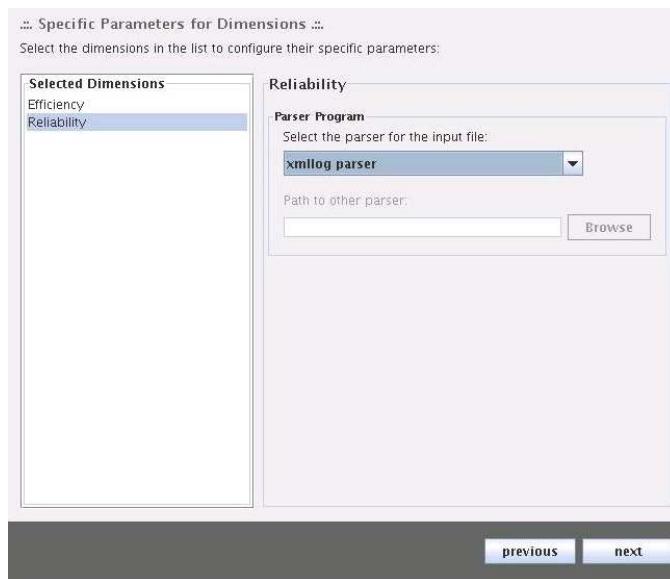


Figure 3.9: 5SQual Interface Specification of Parameters

The user now must define which files the 5SQual tool will generate and where to save them, as shown in Figure 3.10. First, the user must define where to save the configuration

### 3.9. CASE STUDY: 5SQUAL 121

file, which can be used for future evaluations. After, she chooses whether 5SQual should generate graphics and the final report to show the results of the evaluation, and defines in which directory to save them.

The screenshot shows a web-based configuration interface for 5SQual. At the top, there's a header '... Generated Files ...'. Below it, the 'File with Input Parameters' section contains a note about generating a file for the evaluation and saving it to a specified directory ('/home/barbara/5SQual/evaluations'). There's also a 'Browse' button. The 'Evaluation Results' section allows the user to choose between XML Reports and Charts, with both options checked. It also specifies a directory for saving results ('/home/barbara/5SQual/results') and includes a 'Browse' button. At the bottom of the interface, there are 'previous' and 'next' navigation buttons.

Figure 3.10: 5SQual Interface - Definition of Target for the Outputs

Before calling the configured evaluation, the user can verify a summary of the performed configuration as shown in Figure 3.11. From there, she can choose to go to a previous step and redo some configurations or confirm the current ones. In case of a confirmation, the user can execute the evaluation immediately or run it later. The configuration file is generated in either case.

#### 3.9.2 DL EVALUATIONS USING 5SQUAL

To show the functioning of 5SQual, we have performed a set of evaluations that cover all the dimensions implemented by the tool. For this, we used three different DLs, with different characteristics. We defined the set of dimensions to be evaluated on each DL, according to the availability of the resources required for each dimension to be evaluated. Below, we describe the three DLs and the chosen dimensions for each DL.

- Virginia Tech's Digital Library of Electronic Theses and Dissertations (VT-ETD)<sup>19</sup>, is a well-established DL that provides access to full-text documents with different levels of access rights. For this DL, we obtained metadata through the OAI-PMH, therefore

<sup>19</sup><http://scholar.lib.vt.edu/theses/>

**122 3. EVALUATION**

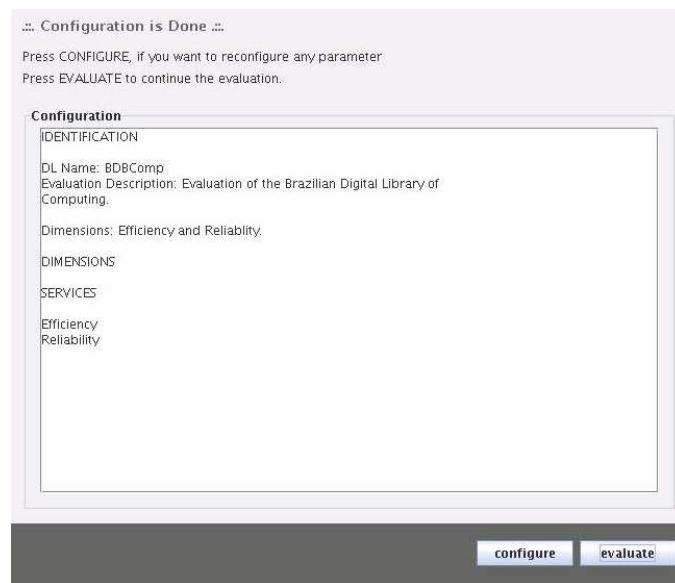


Figure 3.11: 5SQual Interface Confirmation of the Configuration

allowing to evaluate *Completeness* and *Conformance*. Since the VT-ETD metadata provides information about access policies and creation date, we also evaluated *Accessibility* and *Timeliness*.

- The Brazilian Digital Library of Computing (BDBComp)<sup>20</sup>, see [361], is a DL whose catalog has been built from several distinct sources. In this DL, we had easy access to its log files, allowing us to evaluate the *Efficiency* and *Reliability* of its services.
- The ACM 2002 collection (ACM) has 94,818 metadata records along with their internal citation relationships. This collection has been obtained in connection with the CITIDEL project, see [302, 120]. In this collection we evaluated the dimensions based on citation relationships (*Similarity by citations*), *Significance*, and *Timeliness* based on date of the last citation.

Following, we present the results obtained for each evaluated dimension, including charts and some data extracted from the evaluation report. The produced report shows the calculated indicators of the evaluated dimensions. The report covers the *evaluation date*, the *name of the DL*, the evaluations of the *selected dimensions*, and all the numerical indicators chosen in the configuration. For each {dimension, numerical indicator} pair, the report includes: *the number of evaluated items*, *the mean value* and the *standard deviation* considering all the evaluated items, as well as the maximum and minimum values. All the identifiers of the evaluated items are listed in the report in decreasing order of the numerical indicator value. This helps to identify outliers or exceptions. An excerpt of the report is shown in Figure 3.12.

#### VT-ETD Evaluation

From the VT-ETD catalog<sup>21</sup>, we harvested 8,708 metadata records on January 9, 2007 for calculating four dimensions *Accessibility*, *Completeness*, *Timeliness* and *Conformance*.

**Accessibility** The VT-ETD metadata includes the *rights* field, with information about the policy for accessing the digital objects from the DL. Objects can be restricted (available only to the VT community), unrestricted (public), or mixed (parts are public and other parts are restricted). For quantitative evaluation, we associated a value of accessibility to each one of these categories - unrestricted: 1, restricted: 0 and mixed: 0.5. To define these values, we considered the view of an actor that does not belong to the VT community.

The chart obtained from the 5SQual evaluation, shown in Figure 3.13, presents the number of objects with restricted, unrestricted, and mixed access. From the corresponding XML report, it is possible to get the identifiers of the documents for each access category.

As we can see, almost 35% of the ETDs had restricted access to those outside of the university environment, which may reveal an (largely ungrounded) apprehension from

<sup>20</sup><http://www.lbd.dcc.ufmg.br/bdbcomp/>

<sup>21</sup><http://scholar.lib.vt.edu/theses/OAI2/>

124 3. EVALUATION

```
<AboutEval>
<date>05/01/07</date>
<dlName>DLIB TESTE</dlName>
</AboutEval>
<Dimension name="Efficiency">           Dimension Identification
  <indicator name="ResponseTime(seconds)">  Dimension numeric indicator
    <numItems>60</numItems>                 Number of evaluated items
    <avgValue>1.7</avgValue>
    <stdDeviation>2.16</stdDeviation>
    <maxValue>11.0</maxValue>
    <minValue>0.0</minValue>
    <evaluations>                         Results for each evaluated item
      <evaluation value="11.0" numOfItems="1">
        <itemID>SearchByYear - 04/01/07-18:14:41/04/01/07-18:14:52</itemID>
      </evaluation>
      <evaluation value="10.0" numOfItems="1">
        <itemID>SearchByYear - 04/01/07-18:14:21/04/01/07-18:14:31</itemID>
      </evaluation>
      ... Other evaluation figures
    </evaluations>
  </indicator>
</Dimension>
```

Figure 3.12: 5SQual report excerpt

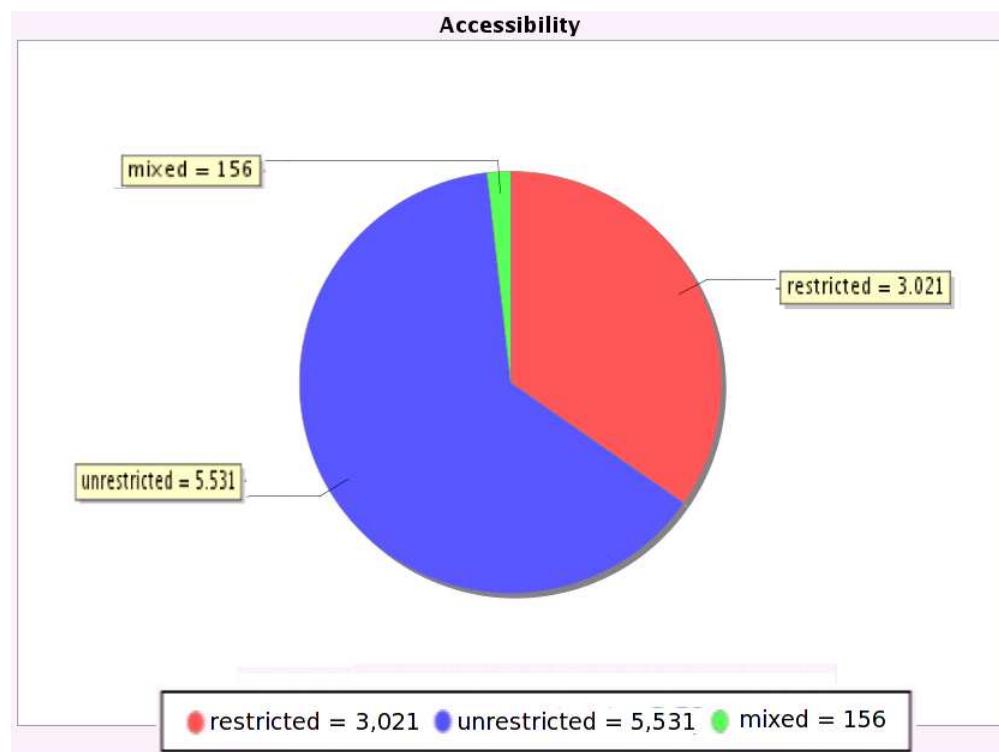


Figure 3.13: VT-ETD - Accessibility Chart

### 126 3. EVALUATION

some of the new graduates that free availability of the material might cause problems in future attempts for publishing the result of their research as scientific papers or patents. Also, the small number of mixed ETDs (less than 2%) may be due to a lack of knowledge by these graduates about the possibility of releasing only parts of the ETDs, an interesting mechanism that can at the same time protect part of the content while publicizing some of the results.

The results of this evaluation, besides revealing to the administrator the behavior of the users who ingest content in the DL, also may indicate alternatives to modify this behavior in the case it is not the desired one. A strategy to increase the accessibility of this material would be to identify the restricted ETDs (using the evaluation report) and to present to their respective authors, and also to other potential authors, the possibility of releasing only portions of their work through mixed access.

**Timeliness** The creation time of the digital objects was extracted by 5SQual from the *date* field of the VT ETD metadata records to calculate their *timeliness*, which in this case was measured in years, given by the difference between the current time and the obtained creation time.

Figure 3.14 presents the chart generated by 5SQual for *timeliness*. It shows the number of items concentrated under each of the shown timeliness values that were calculated based on the current time (date of this evaluation was January 9, 2007). The y axis shows the number of objects that were created on a specific date, and the x axis determines the date when the objects were inserted in the collection.

From this chart, we can see that objects have been continuously created in this DL over the last 10 years and that many objects (almost 100) were inserted on the same date, approximately 1.3 year ago, when scanning of the backfile was speeding up. We also notice that, in the early days of this DL, there was a very stable insertion pattern over the years, which might indicate that the insertion of new objects into the collection was related to some academic events. However, in the last three years this pattern has changed, increasing not only the number of objects per insertion, but also the frequency in which these insertions take place.

From the corresponding XML report, it is possible to find more specific information such as the age of each object, and the average object age (4.37 years) and the standard deviation (2.99 years) of the whole collection. In addition, we can see that the oldest object (identified by oai:VTETD:etd-81197-16953) is 13.76 years old and that the newest one (identified by oai:VTETD:etd-12142006-164331) was created on the date of the evaluation. This reveals that VT-ETD is a DL that keeps its content very timely, what might be due to the fact that the submission of electronic dissertations is mandatory at Virginia Tech.

**Completeness** For calculating *completeness*, we retrieved the VT ETD metadata records which follow the Dublin Core format. This format defines fifteen fields. The *Completeness*

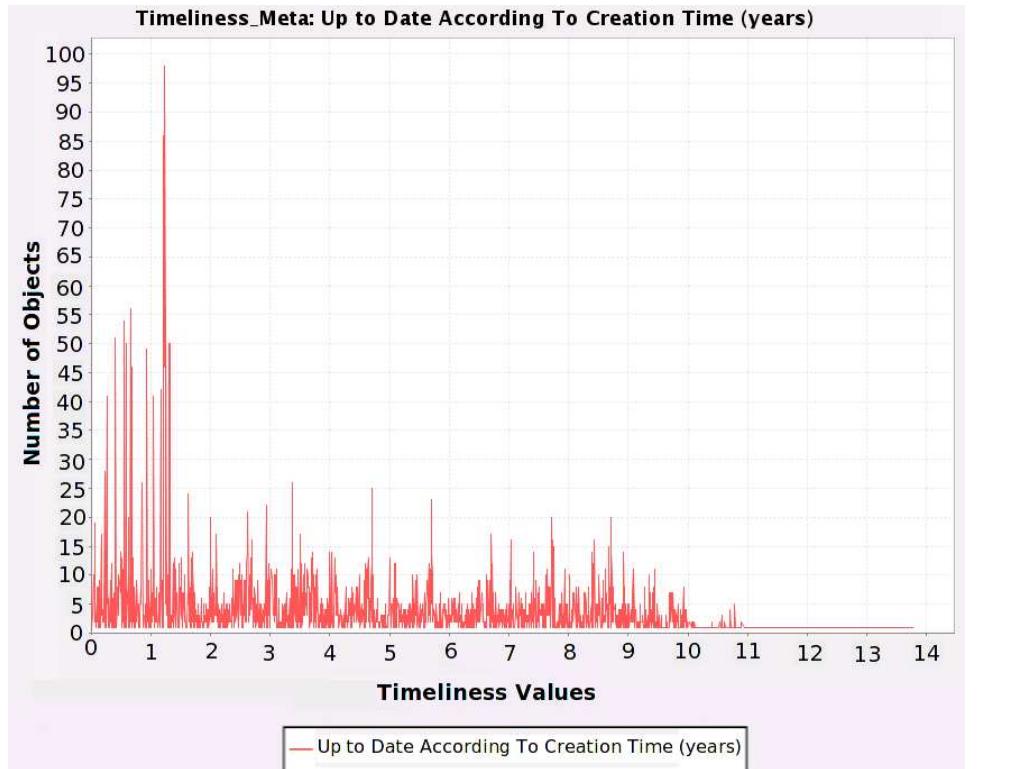


Figure 3.14: VT-ETD - Timeliness Chart

of a metadata record will be given by the number of fields present in a record among the fifteen.

The chart in Figure 3.15 shows that there are four distinct *completeness* values in the catalog. This indicates that there are four groups of records with the same number of fields. The largest group (7,470 records) presents the highest level for *completeness* in the catalog. The records of this group include 13 of the 15 fields defined by the Dublin Core format, which corresponds to a *completeness* value near to 0.87. Looking at the other groups, 24 records present *completeness* equal to 0.67, 1,162 equal to 0.73, and 52 equal to 0.80. No record is totally complete.

From the corresponding XML report, we can obtain for this dimension its average value (0.85) and the standard deviation (0.05). The high average and low standard deviation shows that the catalog of this particular DL is quite complete. Furthermore, retrieving the metadata records using their corresponding identifiers in the XML report, it is possible to check which fields are missing. For instance, for the 0.67 group, the one with the lowest level of *completeness*, we found that the missing fields were *relation*, *coverage*, *description*,

### 128 3. EVALUATION

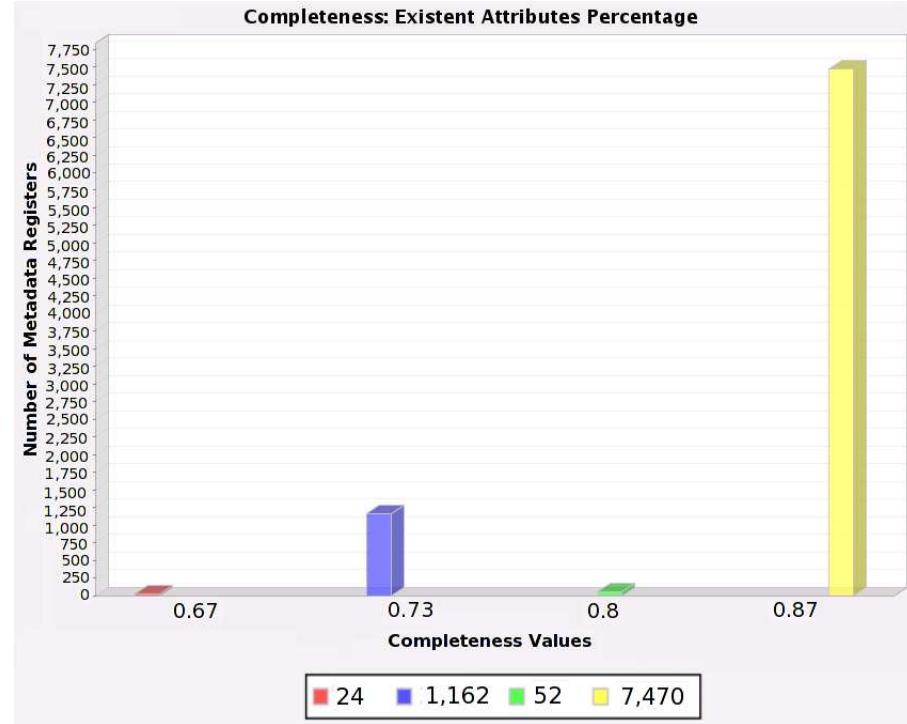


Figure 3.15: VT-ETD - Completeness Chart

*contributor* and *subject*. Analyzing this dimension, the administrator of a DL can have a clear idea of what is missing from its catalog and therefore of the required work for complementing it.

**Conformance** The Dublin Core format does not place any restriction on the minimum and maximum number of times a field should appear. To evaluate *conformance*, we have considered a specific set of Dublin Core fields (*title*, *creator*, *subject*, *publisher*, *date* and *rights*) as mandatory, i.e., we required that they should appear at least once.

The chart in Figure 3.16 shows the VT-ETD *conformance* evaluation regarding this particular set of restrictions. As we can see, all records exhibit high levels of conformance. The fact that there are just two distinct values for this dimension indicates that either the records are totally in conformance with the imposed restrictions (conformance value equal to 1.0) or that they have just one field that is not in conformance with them (conformance value equal to 0.93). The chart also shows that only 25 records were not totally in conformance with these restrictions.

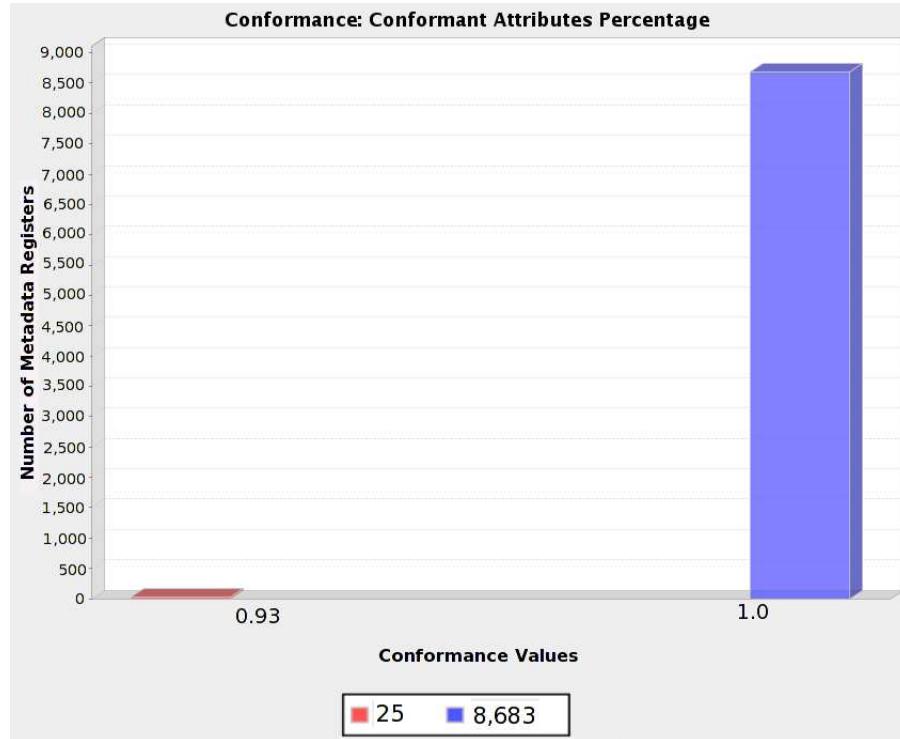


Figure 3.16: VT-ETD - Conformance Chart

Analyzing the XML report, we can identify the 25 records in the 0.93 *conformance* group. When we looked at these records, we find that 24 of them do not have the *subject* field, and that one record, identified by oai:VTETD:etd-08292003-154546, does not have the *title* field filled in.

#### BDBComp Evaluation

Due to easy access to BDBComp log files, we focused evaluation of this DL on two dimensions: *efficiency* and *reliability*. These dimensions were evaluated based on the behavior of the search and browse services. Initially, the necessary data to calculate these dimensions would be extracted from the XMLLog file [361] in use by BDBComp, but because of problems during the generation of this file, the data about the request and response times and the status of the executions were lost. Hence, to illustrate these two dimensions, we extracted information from the Apache logs for *reliability* and simulated some requests for search services (also according to Apache logs) to calculate *efficiency*. This information would be easily extracted from the XMLLog file since 5SQual already comes with a suitable parser.

### 130 3. EVALUATION

**Efficiency** To evaluate *efficiency*, we generated a series of search executions based on the most common queries according to the BDBComp Apache log file. On January 5, 2007, 60 requests were sent to five different BDBComp search services (Search By Author, By Year, By Event, By Title and By Journal), and for each execution we stored the identifier of the service along with its request and response times, specified in seconds. The generated files followed the 5SQual internal format which means that an external parser was not required.

The chart in Figure 3.17 shows the number of executions for each distinct response time. For instance, we can see that 15 of the 60 executions were processed in less than one second and that the slowest execution lasted 11 seconds.

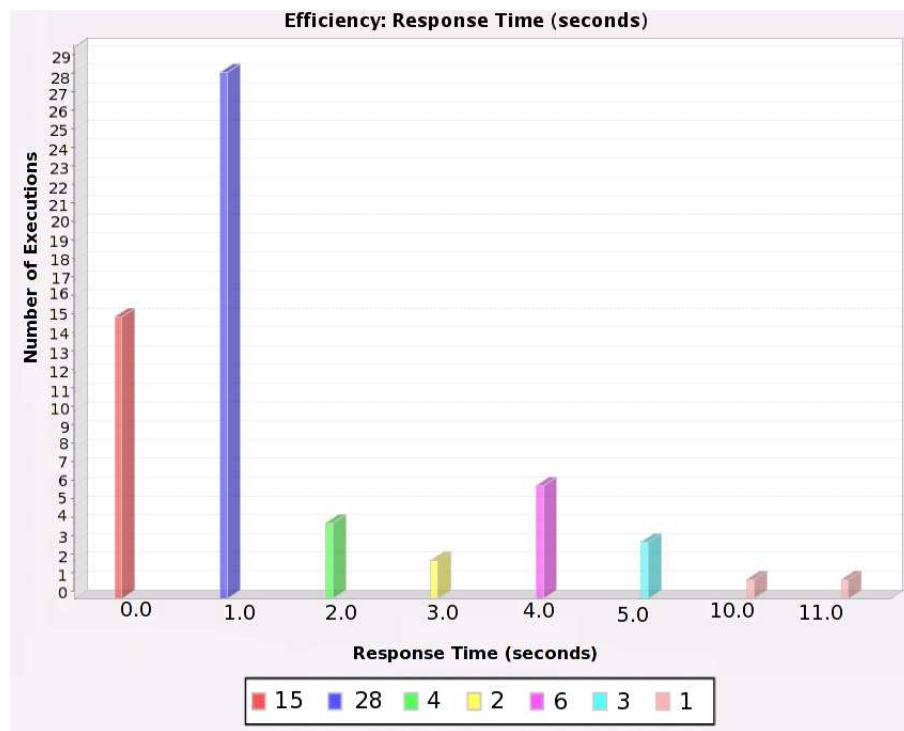


Figure 3.17: BDBComp - Efficiency Chart

From the XML report (an excerpt from it is shown in Figure 3.12), we can obtain more specific information about this evaluation. For instance, the slowest service, which took about 11 seconds to produce an answer, was Search by Year. Further investigation revealed the reasons. Due to the structure of the relational database that implements the BDBComp catalog, the SQL query processing for this kind of search yields a response set that is relatively large when compared with the other ones. Since the search processing time varies linearly with the size of the response set, this explains the poor performance

of this specific type of search. The range of the desired year period also has an impact on the query processing time, since it determines the relative size of the result set. When we analyzed the results, we noticed that the two slowest queries were the ones of type Search by Year for which the largest year ranges (1900 to 2000 and 1990 to 1998) were specified.

**Reliability** To evaluate *reliability*, we extracted data from the BDBComp Apache log files, with 5SQual employing an external parser we created. The Apache logs cover the period between April 14, 2005 and January 3, 2007. We analyzed the searching and browsing services. The chart in Figure 3.18 reflects that 634,250 executions were evaluated, where 35,657 (5.6%) ended in a failure.

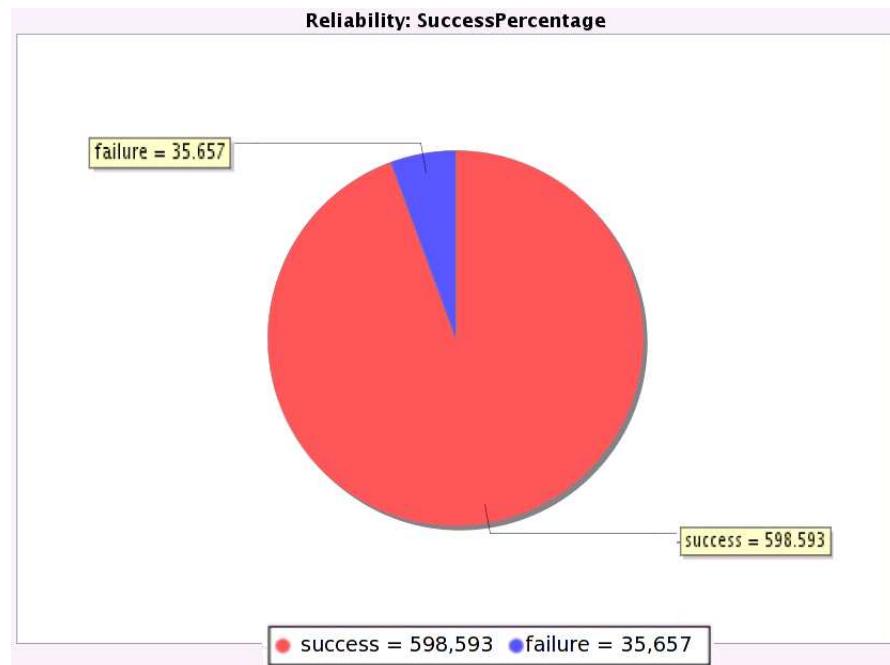


Figure 3.18: BDBComp - Reliability Chart

The corresponding XML report shows additional information. For instance, it reveals that from the failures only one corresponds to Search By Title and that all the rest of the failures are browsing services. Further investigation revealed that these failures were due to a period of instability of the server that went down many times. Additionally, considering the score for a success as 1 and the score for a failure as 0, the average value was 0.94, which means that BDBComp services were quite reliable during the analyzed period.

### 132 3. EVALUATION

#### 2002 ACM Collection Evaluation

The information about citations among digital objects is an important resource for quality evaluation, making it possible to calculate indicators for three dimensions: *similarity*, *significance*, and *timeliness*. To demonstrate the evaluation of these dimensions, we used the ACM 2002 collection of 94,919 metadata records which include citation information and publication dates for each object.

**Significance** We evaluated the *significance* of a digital object in the ACM collection according to the number of citations it receives from other objects in the collection. For this evaluation, 5SQual generated the chart in Figure 3.19. It shows that the majority of the objects has very few citations within the collection and that there are just a small number of documents with a high significance value.

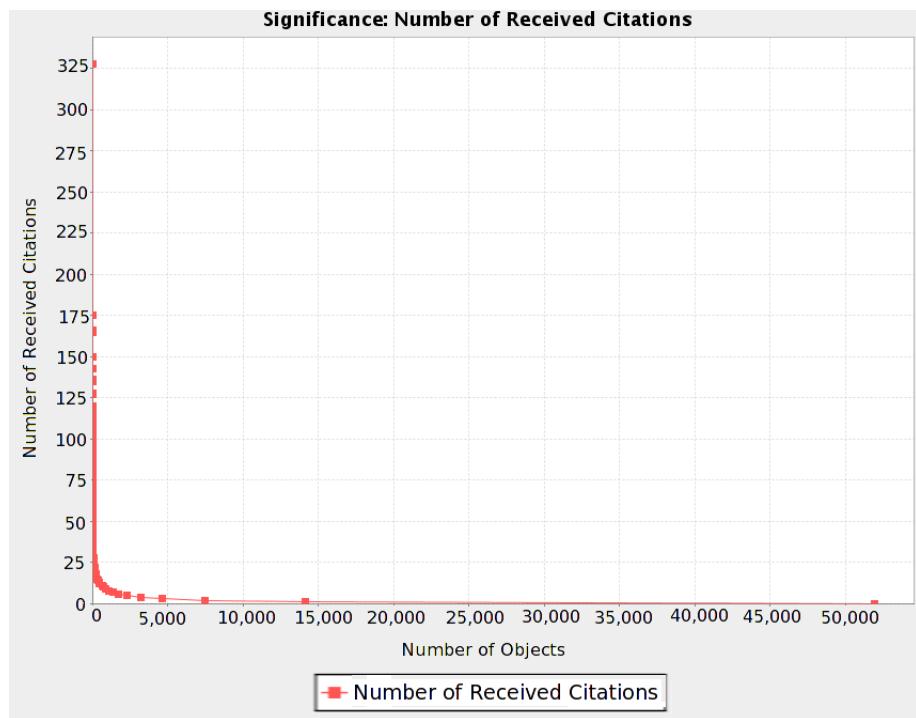


Figure 3.19: ACM - Significance Chart

The XML report details information about this evaluation, specifying for each object its respective number of citations within the collection. The maximum number of citations to a document is 328 in ACM 2002 (for the document “*Computer programming as an art*”). There are 51,925 objects without citations. The average number of citations an object receives is 2.35 (this number refers just to citations of papers that are in the ACM

collection). The high standard deviation value (6.16) shows that the number of citations has high variability.

**Similarity By Citations** To illustrate *similarity by citations*, we have chosen to compare, against the whole collection, two digital objects: the one with the most out-citations (i.e., the references that appear in a document) and the one with the most in-citations (i.e., the citations a document receives). We used two numeric indicators for these comparisons: co-citation [602], considering the document with more in-citations as the reference one, and bibliographic coupling [338] to compare the document with more out-citations against the others. Two documents are co-cited if a third one has citations to both of them (i.e., if they have in-citations in common). The more in-citations in common the more related or similar they are. Bibliographic coupling looks for common out-citations in the two objects being compared.

Bar charts for these evaluations are given in Figures 3.20 and 3.21. They show the number of objects with similarity to the reference documents inside certain intervals. The width of the intervals was obtained by dividing the size of the whole interval (given by the difference between the maximum and the minimum similarity value) by the number of bars.

Both evaluations indicate similar behavior: the majority of the objects are concentrated in the first interval, with the smallest similarity values. But when we look at the values along the horizontal axis, we can see that the bibliographic coupling measure values are more significant (almost 10 times higher) than the values obtained by co-citation. References in the documents (out-citations) contribute more to similarity than the citations they receive (in-citations). In the ACM 2002 collection we can see that there are more digital objects without in-citations (51,925) than without out-citations (46,331). This result is consistent with [131], where it was shown that for DLs containing scientific papers, measures based on bibliographic coupling are more appropriate for similarity detection.

**Timeliness** For ACM 2002, we used *timeliness* regarding the last date a specific digital object was cited, considering only the internal citations within the collection. This date spans the period of the influence of the information contained in the object. This evaluation was performed on January, 14, 2007.

As expected, since the analyzed collection is from 2002, Figure 3.22 shows that the objects more recently cited received these citations four years ago. Moreover it is possible that many objects received citations between 4.5 and 7 years ago and that there are objects that are not cited for more than 55 years.

In the report, it is possible to identify each individual object and the *Timeliness* value associated with it.

## 3.10 SUMMARY

### 134 3. EVALUATION

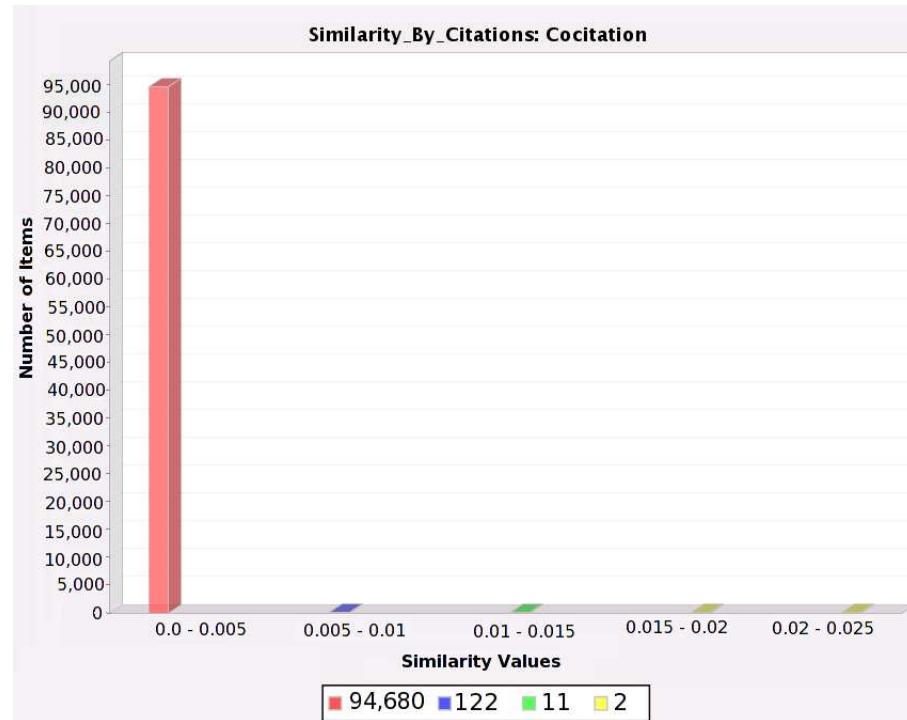


Figure 3.20: ACM - Similarity Chart - Co-citation

### 3.11 EXERCISES AND PROJECTS

1. What are some of the worst problems you have observed in working with digital libraries?
2. What evaluation approach(es) might be used to identify occurrences of such problems?

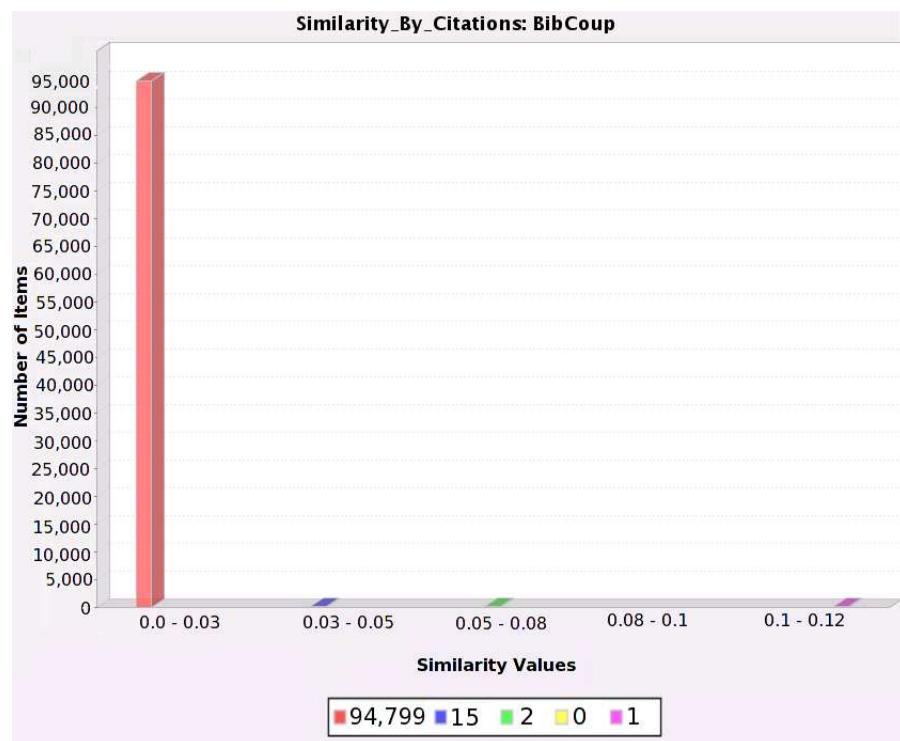


Figure 3.21: ACM - Similarity Chart - Bibliographic Coupling

136 3. EVALUATION

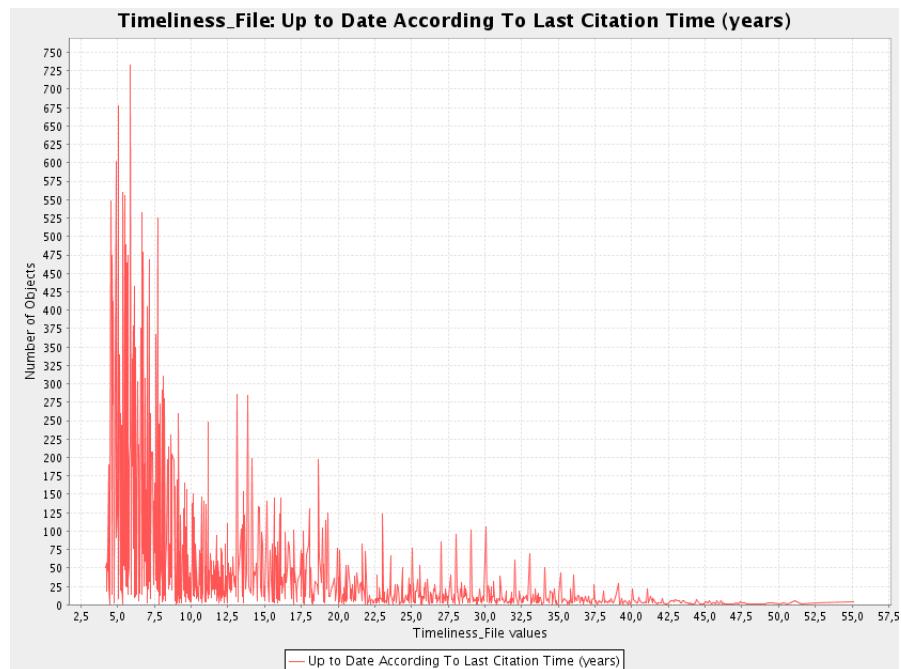


Figure 3.22: ACM - Timeliness Chart

## CHAPTER 4

# Complex Objects

by Nádia P. Kozievitch and Ricardo da Silva Torres

*Abstract:* Aggregation is a key concept in science, that has particular import in the field of digital libraries.

### 4.1 INTRODUCTION

Advances in data compression, data storage, and data transmission have facilitated the creation, storage, and distribution of digital resources. These advances led to an exponential increase in the volume and assortment of data deployed and used in many applications. In order to deal with those data, it is necessary to develop appropriate information systems to efficiently manage collections of such data.

Users involved in creation, management, and access to heterogeneous resources are often concerned with improving productivity. For this, it is important to provide developers with effective tools to reuse and aggregate content. This has been the goal of a quickly evolving research area, namely Digital Libraries.

In order to reuse and aggregate different resources, Complex Objects (COs) have been created, motivating solutions for integration and interoperability. Such objects are aggregations of different information combined together into a unique logical object [367, 461, 460]. Figure 4.1 shows the architecture for a CO-Based Digital Library. The bottom layer has the data sources, accommodating different media types, with different semantic types and formats. The data sources are aggregated in COs, which are later accessed through different services, such as processing, packaging, harvesting, browsing, and searching. These services are later used by digital library applications. Yet, these applications have faced some issues [556, 33]: (i) inadequate support by available DL software for working with COs; (ii) complicated management of COs arising from specific component particularities (such as documents' legal rights); and (iii) inadequate support for multimodal search of complex objects and all components.

Most existing solutions to deal with these issues have focused only on textual data. With the growing demand for visual data, due to the internet, new challenges have emerged. In particular, if we consider image data, significant research efforts have been spent by the Content-Based Image Retrieval (CBIR) [636] community in the development of appropriate systems to efficiently manage image collections.

## 138 4. COMPLEX OBJECTS

In spite of all the advances, there is a lack of consensus on the precise formalization involved in reusing, integrating, unifying, managing, and supporting of diverse application domains for COs and CBIR related tasks. To tackle this issue, we can take advantage of formal concepts to understand clearly and unambiguously the characteristics, structure, and behavior of complex information systems. The benefits of adopting a formal model include the abstraction of general characteristics and common features, and the definition of structures for organizing components (e.g., aggregations, collections). A precise specification of requirements also strengthens the correctness of an implementation [254]. On the other hand, formalized concepts can be used to classify, compare, and highlight the differences among components, technologies, and applications, impacting digital library researchers, designers, and developers.

In this chapter, we address the formal definitions and descriptions for COs by exploiting concepts of the 5S formal framework [254]. Later these definitions are explored in a practical case study, illustrating how CO technologies and the 5S Framework can fit together to support the description and management of COs.

### 4.1.1 WHAT IS A CO

Some authors name the integration of resources into a single digital object as *Aggregation* [672], a *Component-Based Object* [573, 574], a *Complex Object* [460], or a *Compound Object* [32]. We adopt the same definition of structuring digital objects present in [32]: atomistic, compound, and complex. The atomistic approach is when the user has a single file (whether made up from a single or multiple text files) from a preferred format. The com-

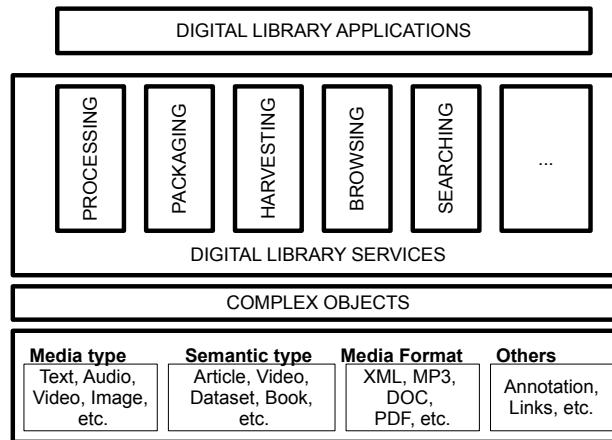


Figure 4.1: Architecture for a CO-Based Digital Library.

pound approach is made up from multiple content files, which may have different formats. A complex object is described using a network of digital objects within the repository.

According to Krafft et al., COs are single entities that are composed of multiple digital objects, each of which is an entity in and of itself [356]. Cheung et al. defined CO in the scientific context as the encapsulation of various datasets and resources, generated or utilized during a scientific experiment or discovery process, within a single unit, for publishing and exchange [115]. In other words, a complex object is an aggregation of objects, that can be grouped together and manipulated as a single object.

COs also were defined as aggregations of distinct information units that when combined form a logical whole [367]. Santanchè, on the other hand, used the idea of COs in the field of software reuse and exchange [573, 574]. Like the script concept [584], or the frame concept [431], the components in a CO are supposed to have the same behavior, respect the same rules, or represent the same concept.

#### 4.1.2 KINDS OF CO

Several complex object (CO) formats arise from different communities [461, 460, 355, 402] and can be used under different domains [348]. In scientific computing, standards arise, such as Network Common Data Form (NetCDF)<sup>1</sup>, Hierarchical Data Format (HDF)<sup>2</sup>, and Extensible File System (ELFS) [330]. HDF and NetCDF, for example, are used in multi-dimensional storage and retrieval, while ELFS is an approach to address the issue of high performance I/O by treating files as typed objects.

COs often are found in persistent database stores. They may be represented using standards from the Moving Picture Experts Group (MPEG) [85] or Metadata Encoding and Transmission Standard (METS) [389]. One example, for including digital object formats, is the Moving Picture Experts Group - 21 Digital Item Declaration Language (MPEG-21 DIDL) [310].

Even though there are a number of standards aiding in the management of COs, there is still incompatibility, motivating solutions for integration and interoperability. As each standard is specialized for a particular domain, it is hard to interoperate across contexts. Yet, it is possible to match some of them, as proposed in [160], in their comparative study of IMS Content Package (IMS CP) [89] and Reusable Asset Specification (RAS) [88].

New standards have emerged, like SQL Multimedia and Application Packages (SQL/MM) [425]. These were defined to describe storage and manipulation support for complex objects. A number of candidate multimedia domains were suggested, including full-text data, spatial data, and image data.

The Open Archival Information System (OAIS) [109] is an International Organization for Standardization (ISO) reference model, with a particular focus on digital information,

<sup>1</sup><http://www.unidata.ucar.edu/software/netcdf/>. Accessed 04 June 2011.

<sup>2</sup><http://www.hdfgroup.org/>. Accessed 04 June 2011.

## 140 4. COMPLEX OBJECTS

both as the primary form of information held and as supporting information for both digitally and physically archived materials. The objects are categorized by their content and function in the operation of an OAIS, into Content Information objects, Preservation Description Information objects, Packaging Information objects, and Descriptive Information objects.

The Open Archives Initiative (OAI) [365] is a framework for archives (e.g., institutional repositories) containing digital content (i.e., a type of digital library). The OAI technical infrastructure, specified in the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) [616], defines a mechanism for data providers to expose their metadata. This protocol mandates that individual archives map their metadata to the Dublin Core, a simple and common metadata set for this purpose.

METS [182] addresses packaging to collect digital resource metadata for submission to the repository. It is a Digital Library Federation initiative. A METS document consists of the following sections: header, descriptive metadata, administrative metadata, file section, structural map, structural links, and behavior. METS uses a structural map to outline a hierarchical structure for the digital library object, where file elements may be grouped within fileGrp elements, to provide for subdividing the files by object version. A *<fileGrp>* structure is used to comprise a single electronic version of the digital library object. *<FContent>* was created to embed the actual contents of the file within the METS document, but it is rarely used. METS provides an XML Schema designed for the purpose of:

- Creating XML document instances that express the hierarchical structure of digital library objects.
- Recording the names and locations of the files that comprise those objects.
- Recording associated metadata.

METS can, therefore, be used as a tool for modeling real world objects, such as particular document types.

SCORM [374] is a compilation of technical specifications to enable interoperability, accessibility and reusability of web-based learning content. With a Content Aggregation Model, resources described in a imsmanifest.xml file, organized in schema/definition (.xsd and .dtd) files, and placed in a zip file, are used as a content package. SCORM defines a web-based learning Content Aggregation Model and Run-Time Environment for learning objects. In SCORM, a content object is a web-deliverable learning unit. Often, a content object is just an HTML page or document that can be viewed with a web browser. A content object is the lowest level of granularity of learning resources, and can use all the same technologies a web page can use (e.g., Flash, JavaScript, frames, and images).

MPEG-21 [85] aims to define an open framework for multimedia applications, to support, for example, declaration (and identification), digital rights management, and adapta-

## 4.1. INTRODUCTION 141

tion. MPEG-21 is based on two essential concepts: the definition of a fundamental unit of distribution and transaction, which is the digital item, and the concept of users interacting with them. Within an item, an anchor binds descriptors to a fragment, which corresponds to a specific location or range within a resource. Items are grouped in a structured container using an XML-based Digital Item Declaration Language (DIDL). In addition, a W3C XML Schema definition of DIDL is provided.

Table ?? summarizes METS, SCORM, and MPEG-21 regarding basic principles available in complex objects: what is the data basic unit, how to relate a part of a document, how to identify it, and how to structure the components.

**Table 4.1:** How standards handle basic CO concepts.

Name	Unit	Internal Component	Identifier	Structure
METS	Simple object	FContent structure	OBJID	Structural Map
SCORM	Asset	Sequence rules	—	Schema/definition files
MPEG-21	Resource	Anchors and fragments	URI	XML-DIDL

Three technologies were choosed in this chapter to explore CO concepts: DCC, OAI-ORE, and Buckets. Their terms for compound information vary: DCC uses the term component; Buckets use the term logically grouped items; and OAI-ORE uses aggregation or compound object.

Digital Content Component (DCC) [572, 160, 502, 573, 574] was proposed in 2006, as a generalization format for representing compound objects. The approach derives from an analysis and comparison of content packages, and Open Complex Digital Object (OCDO) and reuse standards [572].

A DCC is composed of four distinct subdivisions (Figure 4.2):

- **content:** the content itself (data in its original format such as a PDF, Word or HTML file);
- **structure:** the declaration of a management structure that defines how components within a DCC relate to each other, in XML;
- **interface:** a specification of the DCC interfaces using open standards for interface description – WSDL and OWL-S (semantics); and
- **metadata:** metadata to describe version, functionality, applicability, and use restrictions – using OWL.

Buckets [463, 462, 464] provide an archive-independent container construct in which all related semantic and syntactic data types and objects can be logically grouped together, archived, and manipulated as a single object. Buckets are active archival objects and can communicate with each other, or arbitrary network services. Buckets are based on standard

## 142 4. COMPLEX OBJECTS

World Wide Web (WWW) capabilities to function, managed by two tools. One is the author tool, which allows the author to construct a bucket with no programming knowledge. The second one is the management tool, which provides an interface to allow site managers to configure the default settings for all authors at that site.

OAI also launched the Object Reuse and Exchange (OAI-ORE) [402] project which defines standards for the description and exchange of aggregations of Web resources, and is developing interoperable, and machine-readable mechanisms to express compound object information on the web. OAI-ORE makes it possible to reconstruct the logical boundaries of compound objects, the relationships among their internal components, and their relationships to other resources. The information is encapsulated with named graphs: a set of RDF assertions identified by a URI. Figure 4.3 highlights some concepts from the 5S framework and OAI-ORE. Note that concepts such as resource - digital object and complex object can be mutually mapped.

A named graph can be described by a resource map. OAI-ORE uses the web architecture [367], essentially consisting of:

- URIs for identifying objects;
- resources, which are items of interest;
- standard protocols, such as HTTP, that enable access;
- links via URI references;
- named graphs for encapsulating information into a compound object.

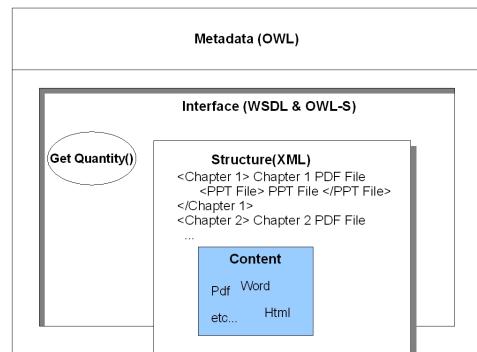


Figure 4.2: Digital Content Component Representation.

### 4.1.3 COMPARISON OF COS (DCC, BUCKETS, OAI-ORE)

DCC, Buckets, and OAI-ORE have been used with different purposes, but their focus is still aggregate resources. For example, the different advantages arise: from the space perspective, DCC works with ontologies, while from the streams perspective, the HTML-based structure in OAI-ORE facilitates integration with applications. Their operations and restrictions are different, since they manage different perspectives of the CO. The information aggregation can use several abstractions to differentiate internal parts, such as named graphs, XML files and unix directories. Different perspectives of the same entity can be explored in interfaces, methods, or named graphs.

For highlighting even more their differences, we selected other parameters related to identity, components, structure, boundary, and manipulation (shown in Table ??): (i) unique identifier; (ii) component division; (iii) how the components are composed; (iv) what is encapsulated; (v) usage; (vi) internal format and structure; (vii) implementation or access tools; (viii) advantages; (ix) how they manage software; and (x) how they handle preservation.

All DCCs and each component of a CO in OAI-ORE has an URI associated with them, thereby making them web URI-identified resources. Each bucket has its own unique id (handle). The component division is implemented by process and passive DCCs, unix directories in Buckets, and the resource maps in OAI-ORE. Each of these components can encapsulate metadata, content and processes in DCCs, metadata and content in Buckets, and description of aggregations in OAI-ORE.

In DCC, the internal CO format is divided in content, structure, interface and metadata. In Buckets, the internal CO format is divided in elements, packages and the final bucket. In OAI-ORE the resource map describes the aggregation of resources identified by URIs.

The three technologies have different implementations, but all of them allow the components reuse. They present different advantages, but all include characteristics to-

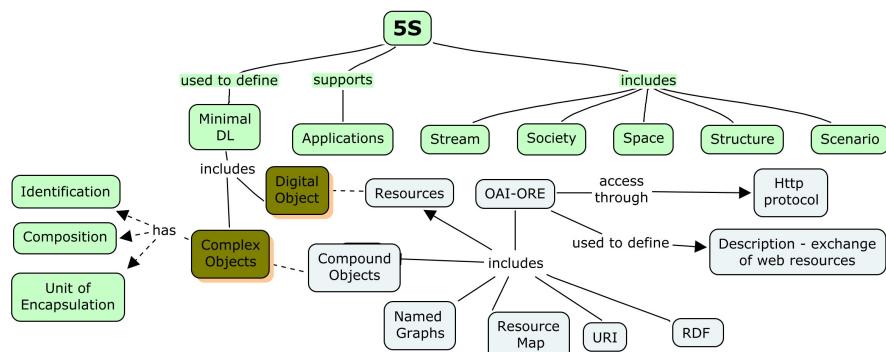


Figure 4.3: Matching the main concepts of the 5S framework and OAI-ORE [351].

## 144 4. COMPLEX OBJECTS

wards digital preservation (encapsulate content and software, directories can be zipped for archival, and description for exchange and reuse).

**Table 4.2:** Basic CO concepts from DCC, Buckets, and OAI-ORE perspective.

Description	DCC	Buckets	OAI-ORE
Unique identifier	URI	Handle	URI
Component Division	Process and passive DCCs	Unix directories	Resource map, aggregations
What is encapsulated?	Metadata, content, processes	Metadata, content	Aggregation description
Format	Content, structure, interface and metadata	Buckets, packages, and elements	Map resources, URIs, aggregation
Implementation	Jar file, extensible to other languages	Access through Author and Management Tool	Mapping resources to resource map
Compose CO	Parts accessed through relative URI, other DCCs	Packages and nested Buckets	Resource map and aggregations
Advantages	Ontology, interface, encapsulate executable content	Pointer to remote package, network or database, log	Move repository, used as standard between different systems
Manage software?	Can encapsulate content with respective SW	As a normal file	As a normal file
Preservation	Encapsulate executable and non-executable content, structure and description allows reuse	Directories can be easily zipped for archival or transport	Description allows easy transport and reuse

## 4.2 RELATED WORK

In different portions of the literature, a variety of perspectives and parameters have been presented for exploring COs and aggregations:

- ontologies: Gerber et al. in [240] specified for example, an ontology for the encapsulation of digital resources and bibliographic records;
- granularity: Fonseca et al. in [189] cited vertical navigation, where accessing a class immediately above or below, implies in a change of level of detail;
- standards for aggregations: in the context of the DELOS project a DL Manifesto [97] has been proposed, where Caldela et al. explored the completeness of the CO (measuring whether a minimal required set of elements is available). If we consider standards for aggregations, other parameters could still be included, like the number of components, types of accepted compositions, or the minimum/maximum elements that the composition should have;
- priority among components: in the context of the DELOS project [97], it was also explored the priority of one component compared to the complete set, so, if this component is copied or deleted, the other parts are copied or deleted along with it;

- portability for the CO structure: Park et al. in [499] explored the adaptation of the CO structure to different domains, such as portable devices, where some components (such as videos) might not be necessary;
- access to components: Manghi et al. in [414] suggested different access roles for the different parts, as suggested in the authentication and authorization service;
- reuse and preservation: Rehberger et al. in [544] examined the role that secondary repositories can play in the preservation and access of digital historical and cultural heritage materials;
- others: track of provenance [441], and timeline [268].

COs have also been used in preservation and harvesting [422, 558], to combine current objects to create new ones [572], or even for grouping information to respect the same permissions or operations. Depending on the aggregation, different layers can be exposed, using different information granularity, or type of media, for example.

### 4.3 FORMALIZATION

Formalizing complex objects facilitates the development, comparison, and evaluation of solutions based on distinct information resource integration; makes clear to users what a solution means; indicates how components are related; and helps users evaluate the applicability of a solution. Furthermore, it allows us to leverage special-purpose techniques for combining, aggregating, and reference the integration process. In this section, we first introduce the basic notations based on the 5S framework, followed by an overall approach for formalizing complex object, the minimum CO and the complex image object (ICO).

*Notation:* Let  $DL_1$  be a digital library; let  $\{do_1, do_2, \dots, do_n\}$  be the set of digital objects  $do$  present in  $DL_1$ ; let  $H$  be a set of universally unique handles (unique identifiers); let  $SM$  be a set of streams; and let set  $ST$  be a set of structural metadata specifications.

#### 4.3.1 CO

From a computational view, a DL, in terms of content, is mainly composed of simple components named digital objects.

A **digital object** is defined as a tuple  $do = (h, SM, ST, StructuredStreams)$ , where

1.  $h \in H$ , where  $H$  is a set of universally unique handles (labels);
2.  $SM = \{sm_1, sm_2, \dots, sm_n\}$  is a set of streams;
3.  $ST = \{st_1, st_2, \dots, st_m\}$  is a set of structural metadata specifications;

## 146 4. COMPLEX OBJECTS

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

*Streams* are sequences of elements of an arbitrary type (e.g., bits, characters, images, etc.). *Structural Metadata Specifications* correspond to the relations between the object and its parts (as chapters in a book). *Structured Streams* define the mapping of a structure to streams (how chapters, sections, introduction, etc. are organized to define a book). More details are available in [254].

**Definition 4.1** We define a **complex object** as a tuple  $cdo = (h, SCDO, S)$  where

1.  $h \in H$ , where  $H$  is a set of universally unique handles (labels);
2.  $SCDO = \{DO \cup SM\}$ , where  $DO = \{do_1, do_2, \dots, do_n\}$ , and  $do_i$  is a digital object or another complex object; and  $SM = \{sm_a, sm_b, \dots, sm_z\}$  is a set of streams;
3.  $S$  is a structure that composes the complex object  $cdo$  into its parts in  $SCDO$ .

Note that the mentioned definitions consider the object's metadata in a separate catalog [254]. The  $DO$  and  $SM$  components are finite sets, therefore the  $S$  structure is also finite, defining what belongs to the CO or not (concept referred to as a boundary).

The  $S$  structure in the **complex object** is not specified, therefore can be extended to any structure that represents parts of a whole, such as a list, a tree, or even a graph. As a practical example, we can mention the Fedora Commons approach [32], where lists represent multiple single files which were packed together, and graphs represent files which are related, creating networks of digital objects. If we consider files arranged in HTML5 [499], the  $S$  structure can be extended to a cyclic graph. Our focus is not to explore these fine-grained concepts, but to consider a high-level approach: aggregate logically and perhaps physically, distinct objects, so they can be represented as a single unit.

### 4.3.2 MINIMUM CO

We consider the minimum CO as a tuple  $cdo = (h, SCDO, S)$  where

1.  $h \in H$ , where  $H$  is a set of universally unique handles (labels);
2.  $SCDO = \{DO \cup SM\}$ , where  $DO = \{do_1\}$ , where  $do_1$  is a digital object; and  $SM = \{sm_a, sm_b, \dots, sm_z\}$  is a set of streams;
3.  $S$  is a structure that indicates  $\{do_1\}$  as a component of  $cdo$ .

Our definition considers that a CO should comprise at least one digital object. If a lower granularity is necessary, the atomistic definition [254] can be applied.

### 4.3.3 ICO

In particular, the complex image object (ICO) is a CO with the following components: the digital image object, feature vector and similarity scores (presented in Figure 4.4). If we consider the CO definition, the complex image object (ICO) has the structure  $ico = (h, SCDO, S)$ , where:

- $h$  is a unique handle that identifies  $ico$ ;
- $SCDO = \{DO \cup SM\}$ , where  $DO = \{do_1, do_2, \dots, do_{2k}, do_{31}, \dots, do_{3k}\}$ , where  $do_1$  is an **image**,  $k$  is the number of descriptors,  $do_2, \dots, do_{2k}$  is a set of **feature vector digital objects**, and  $do_{31}, \dots, do_{3k}$  is a set of **StructuredFeatureVectors** (with the similarity measures, according to a specific descriptor  $k$ ); and  $SM = \{sm_a, sm_b, \dots, sm_z\}$  is a set of streams;
- $S$  is a structure that identifies how  $do_1, do_2, \dots, do_{2k}$ , and  $do_{31}, \dots, do_{3k}$  are composed.

Note that each ICO component is a **digital object**, therefore having also its own handle. This allows users to explore the collection not only by the COs, but also by the individual components (digital objects).

An **complex image object collection**  $ImgCO$  is a tuple  $(C, S_{imgdesc})$ , where  $C$  is a collection (see Def. 17 in [254]), and  $S_{imgdesc}$  is a set of image descriptors. Function  $FV_{desc}$  defines how a feature vector was obtained, given an complex image object  $ico \in C$  and an image descriptor  $\hat{D} \in S_{imgdesc}$ .

## 4.4 CASE STUDY: FINGERPRINTS

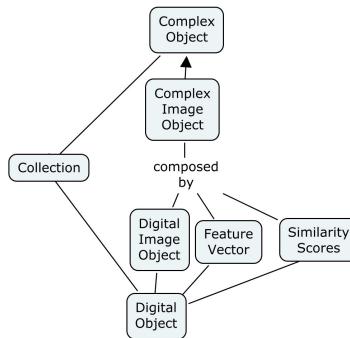


Figure 4.4: The Complex Image Object.

## 148 4. COMPLEX OBJECTS

### 4.4.1 INTRODUCTION

In this section we present a case study to provide a better understanding of how the CO concepts can fit together in real DL applications, in particular, on a Fingerprint Digital Library [353].

Consider a Fingerprint Digital Library which unifies four different digital libraries, from a compound object (CO) perspective. We offer this as an example of how database modeling approaches can be enriched with a theory-base handling of CO concepts, so as to better add requirement analysis, design and implementation of important database and/or digital library applications. Those aware of law enforcement activities will know of the first type of DL (DL1), associated with databases of stored fingerprints. Another domain relates to a project to create training materials for fingerprint examiners (DL2). A third type of DL relates to the evidence and data describing a crime scene (DL3). A fourth type of DL relates to our NIJ funded research studies supporting experimentation with fingerprint image analysis techniques, quality measures, and matching methods (DL4). Combining these four into an integrated DL, where compound objects allow us to work across these DLs, yields a very interesting and very large DL.

In DL1, information is used to identify a person. It manages large law enforcement databases which may have millions of people's prints, where each one can come with 10 fingers, 10 toes, palm, pads of feet, etc. One of the biggest biometric database and fingerprint identification system is from the Federal Bureau of Investigation (available at <http://www.fbi.gov/hq/cjis/iafis.htm>). It has at least 66 million subjects in the criminal master file, along with more than 25 million civil print images.

DL2 has a different purpose: to educate and train users. Ideally, for testing fingerprint examiners, the combination of examples identified could be used for assessment, so each case in an exam is distinct, reducing opportunities for cheating. The training modules will have examples for instruction, and yet others for exercises and examinations, taken from all of the other DLs.

In DL3, images are used for matching or excluding individuals. The evidence from a crime scene can come from thousands of people who visited a popular place, or touched an object, creating data which can be later compared with a criminal history record. Each person has ten fingers, and each finger can produce different images depending on the type of distortion, e.g., from a finger sliding. In addition, there are overlays of different prints, i.e., combinations of images from the fingers under the same substrate.

In DL4, the focus is on fingerprint algorithms, varying parameters on skin distortion and blurring. Distorted or synthetic images are created by algorithms that simulate motion and/or skin distortion. The combination of a single recorded print with the 10 parameters, for example, can synthetically generate about 10,000 images.

#### 4.4. CASE STUDY: FINGERPRINTS 149

To give a sense of scale, suppose that one image generates 100 distorted images. Multiply by 25 million possible suspects. Then try to match a crime scene image which has 55 partial fingerprints. Finally, select and link good examples for use in training.

Through the integration, the digital library unifies four different communities, allowing each one to see different perspectives, and explore the system as a whole, or focus in a determined area. In addition, we can take advantage of digital library services (e.g., browsing and searching), formalisms, and preservation solutions.

##### 4.4.2 APPROACH

According to [564], the integration process is divided into four steps: (i) discovery: systems “learn” about the existence of each other; (ii) identification: systems unambiguously identify their individual items; (iii) access: systems access their items; and (iv) utilization: systems synthesize their items. Our case study presents the first two steps.

We used COs to facilitate the aggregation abstraction (as shown in Fig. 4.5), embracing components from different domains, and unifying them with a single concept.

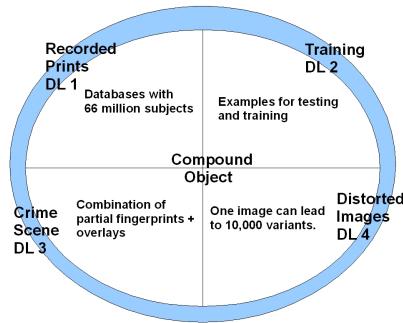


Figure 4.5: The integration of fingerprint digital libraries.

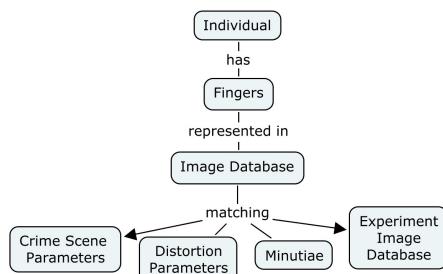


Figure 4.6: The main classes representing the Fingerprint DL.

## 150 4. COMPLEX OBJECTS

Figure 4.6 presents the concept map of the main classes, as a summary of the entity-relationship diagram [352]. Class Individual, for example, aggregates all the information from the 10 fingers, along with images, minutiae, and other metadata for a single person. Later the user can explore if the same person has images distorted by algorithms, extracted from a crime scene, or manipulated by the police station.

The integration of the four sub-systems can be exemplified by Figure 4.7. Compound Object 1 (CO1) has the following components: a fingerprint image from system A, one distorted image from system B, a crime scene image from system C, and a link to related training material, taken from system D. The components can be identified by CO1.A.1, CO1.B.1, CO1.C.1 and CO1.D.1, respectively. The CO1 structure can be represented by RDF, while the content could be packaged using OAI-ORE or DCC. The interface of CO1 can comprise the union information of its four components, along with the union of their respective vocabularies (individual, fingers, thumb, quality, distortion, parameters, etc.).

If we consider the CO formal treatment of Figure 4.7, we have  $\text{CO1} = (h, \text{SCDO}, S)$  where

1.  $h$  is a unique handle that represents CO1, and  $h \in H$ , where  $H$  is a set of universally unique handles (labels);

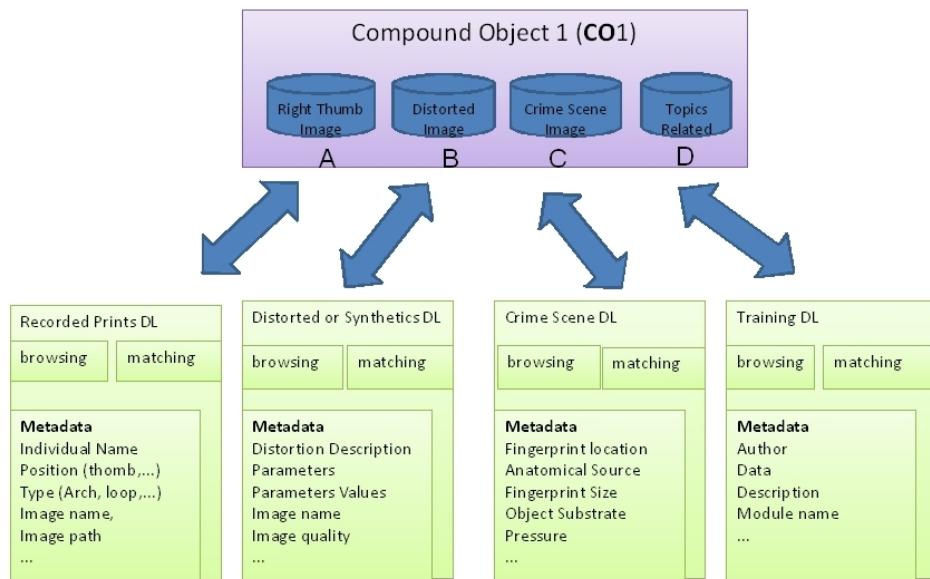


Figure 4.7: An example of compound object using four digital libraries: (A) Recorded Prints, (B) Distorted Images, (C) Crime Scene Images, and (D) Training Material.

#### 4.4. CASE STUDY: FINGERPRINTS 151

2.  $SCDO = \{DO \cup SM\}$ , where  $DO = \{A.1, B.1, C.1, D.1\}$ ; and  $SM = \{sm_a, sm_b, \dots, sm_z\}$  is a set of streams;
3.  $S$  is structured by means of an XML file, aggregating the compound object  $cdo$  into its parts in  $SCDO$ .

Examples of communities in a fingerprint digital library include criminal justice agencies, scholars, students, and researchers. Specific rules and different roles can also be used to map restrictions, such as the public non-availability of recorded prints from the police station.

Different scenarios can be defined to describe each of the four DLs and their interactions as a CO. Processes such as matching, creating distortions, training can also be described as scenarios. Softwares used for creating fingerprint distortions and matching include detailed information about parameters (such as angles, flows, plasticity, displacement, number of matches, etc.) and can also take advantage of scenarios for their description.

Examples of structures in a fingerprint digital library include the information organization (such as Figure 4.6). Each person has 10 hand fingers, and each finger can produce different images depending if it is from a police station, or distortion, or a crime scene. If they belong to the same finger, structures are used to represent this hierarchy.

Streams represent the different type of images and files managed. Users can explore not only the individual components, but also the CO as a unique digital object. As services, we can list browsing, matching, textual search, and multimodal search.

The vocabulary used for the description of the content, structure, metadata, versions, functionality, applicability, and use restrictions relates to the conceptual space.

The initial exploration of CO concepts on a project in an early development stage was important to highlight the amount of information and details needed to manage and aggregate. Further, DCC could be used to encapsulate each image, the details of each DL, the aggregation of the CO, and the software used. OAI-ORE could be used to describe the aggregations in an integrated DL service, providing *the match between latent and recorded fingerprints, or a chain of evidence to convince a jury of confidence of match*, for example. Since both technologies use URIs for identify resources, they could be integrated for further exchange and reuse of resources among the different communities. Other integrated DL services could consider *the object versions* (with the composition of distortions, for example), *correspondence of versions with provenance*, or the harvesting and matching in a DL integration process.

For the harvesting process, the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) can be used, defining a mechanism for data providers to expose their metadata. For disseminating the content in concert with a metadata harvesting protocol, some steps are necessary [422]: (i) wrap the data in a packaging format; (ii) include the metadata; (iii) encode the references to the files; and (iv) harvest the package. For this, OAI-ORE or DCC can be used, representing the objects and aggregations.

## 152 4. COMPLEX OBJECTS

The complexity of the mapping and updating in the integration 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 [598].

In the case of compound object technologies, such as DCC and OAI-ORE, the mapping process also depends on other factors, such as how the components are aggregated, what is their granularity, which vocabulary each technology is using, how the components are identified and structured, or how they are organized in a schema.

In summary, our case study explored two steps of the integration process: the “discovery” of each system, and the identification of individual items for a possible aggregation. For this we used the 5S Framework along with the CO technologies to analyze the integrated fingerprint digital library, from the identity, components, structure, and boundary perspectives. Finally, we discussed how the components can be accessed later, along with their individual metadata.

### 4.4.3 IMPLEMENTATION

Considering the large size and types of variance of the fingerprint image databases and the computational cost of fingerprint verification algorithms, for the prototype we used a pre-processing phase, using Content-Based Image Retrieval (CBIR) techniques. This phase is responsible for ranking similar images based on a texture descriptor. The objective is to reduce the number of one-to-one comparisons, seeking improvements both in terms of accuracy and retrieval speed. In this sense, we study the characterization of textural patterns that can be found in fingerprints.

This solution requires the definition of appropriate image descriptors, which are characterized by (i) an extraction algorithm (such as texture, shape, or color) to encode image features into feature vectors; and (ii) a similarity measure to compare two images based on the distance between the feature vectors. The similarity measure is a matching function (e.g., using Euclidean distance), which gives the degree of similarity for a given pair of images represented by their feature vectors. The larger the distance value, the less similar the images.

The prototype had the following phases: (i) the “discovery” and definition of each part of the compound object; (ii) the identification of the compound parts; (iii) the CBIR process; (iv) the encapsulation of the image and related metadata; and (v) the CO publishing.

**Phase 1.** The discovery and definition of each part of the CO played a key role to define the data types and different DLs of the fingerprint integration. The objective of the prototype was to aggregate data including the images, and metadata. Only two fingerprint digital libraries were selected for the prototype: the recorded prints from the police and the crime scene fingerprints.

**Phase 2.** In phase two we defined that the aggregation would comprise the “individual” concept. For the identification of the compound parts, we used the database, which



Figure 4.8: Samples of images from a Recorded Print DL from the Police.

matched the images to respective fingerprint DL, metadata, image content descriptors, and similarity distances.

**Phase 3.** In phase three, the integration of the CBIR process allowed a pre-categorization of the image, using texture comparison. For this, the Statistical Analysis of Structural Information (SASI) [5] descriptor was used. The CBIR processing of Figure 4.8 - part 11 for example, generates a feature vector, and the similarity distances for the other images on the collection. Figure 4.10 shows the ranking for Figure 4.8 - part 11 according to the texture comparison. The 10 top-down images are the most similar images compared to Figure 4.8 - part 11.

The CBIR processing of a second image (Figure 4.9 - part 3) generates a second feature vector, and another set of similarity distances. Figure 4.11 presents the image ranking for Figure 4.9 - part 3 regarding the texture comparison.

**Phase 4.** In phase four, DCC was used for the encapsulation of resources. DCC allows the recursive construction of components using composition of other components, based on a model which generalizes reuse content practices of decomposition - storage/retrieval - composition. The main characteristics of DCC are: (i) it can uniformly encapsulate both executable (programs, processes, etc.) and non-executable (data sets) content; (ii) it provides a context description for its content, using references to ontologies; (iii) it provides descriptions of interfaces to operations, also with references to ontologies; and (iv) it is independent of a platform or programming environment.

154 4. COMPLEX OBJECTS

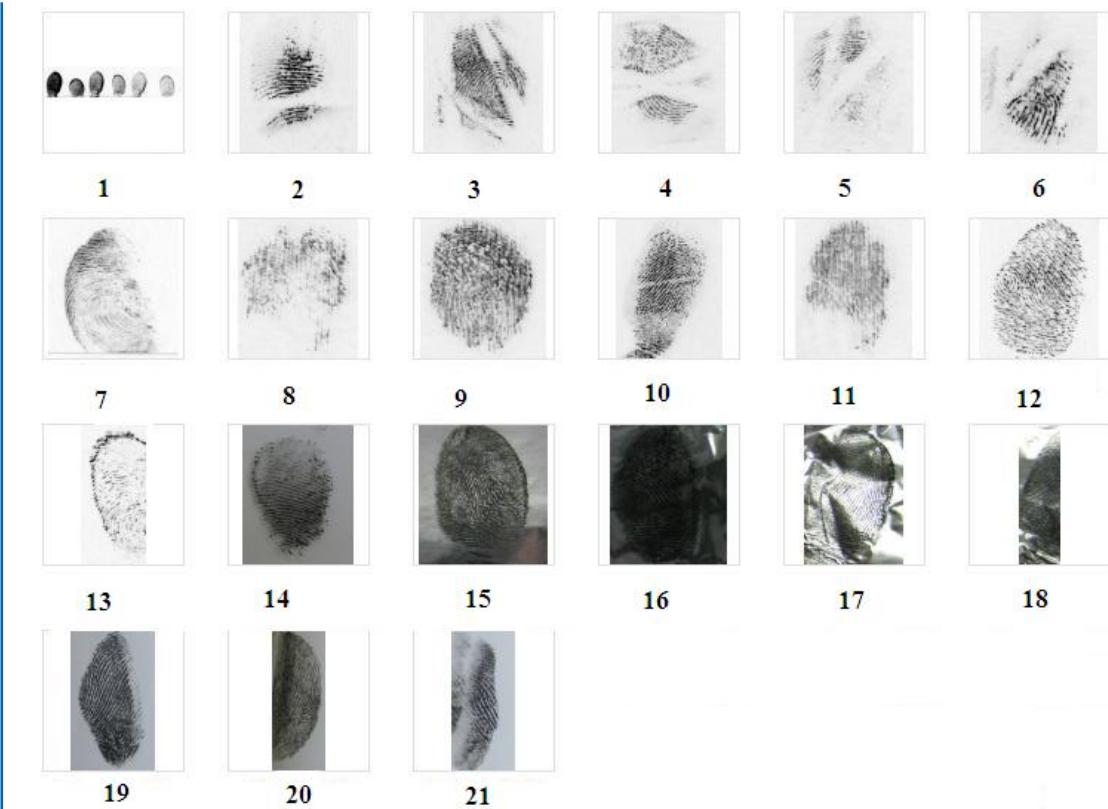


Figure 4.9: Samples of fingerprints from a DL which simulates a crime scene.

The encapsulation of resources was built in a three layer model (as shown in Figure 4.12): (i) the image compound object aggregating the CBIR and image information (encapsulated in ImageCODCC); (ii) the individual fingerprint digital library, represented by the police fingerprint DL (encapsulated in PoliceCODCC) and the crime scene DL (CrimeCODCC); and (iii) the individual compound object, aggregating all the images and metadata for a same person (encapsulated in IndividualDCC).

In the mentioned example, Figure 4.8 - part 11 and Figure 4.9 - part 3 were aggregated into two ICOs. They are represented by the ImageCODCC, which centralizes the encapsulation of the CO, concerning the JPEG images, an XML file (with metadata and similarity distance), and feature vectors for each respective image. In this case, the feature vectors are binary files. Operations available include the generation of the image CO compressed file and image CO XML. DCC metadata includes the image CO name and file location.

#### 4.4. CASE STUDY: FINGERPRINTS 155

The second layer contains the information aggregation relative to the respective fingerprint library. In this case, Figure 4.8 - part 11 belongs to an individual from the police fingerprint digital library and is encapsulated in PoliceCODCC. Figure 4.9 - part 3 belongs to the same individual, but now in the crime scene digital library, which is encapsulated in CrimeCODCC. Operations available include the generation of the CO compressed file for the respective fingerprint DL. DCC metadata include the individual name, the finger position, and the object substrate of the crime scene fingerprint.

The third layer corresponds to the Compound Object 1 presented in Figure 4.7, aggregating information from one “individual” using different fingerprint DLs. In the mentioned example this represents the aggregation of all images from Figure 4.8 and Figure 4.9 (since they represent the same individual), along with their respective feature vectors, similarity distances, and metadata. In the mentioned example, this is represented by the IndividualDCC having the name Joseph Murch. Figure 4.13 presents the XML for the individual aggregation: the initial block presents the individual metadata (name, age, sex); the second block presents the XML for the police fingerprint DL CO, and the last block presents the XML for the crime scene DL CO. Note that the second and third block have the image CO, starting with the tag <image>. Operations for the IndividualDCC include the generation of the CO with all the information from an individual. DCC metadata include the number

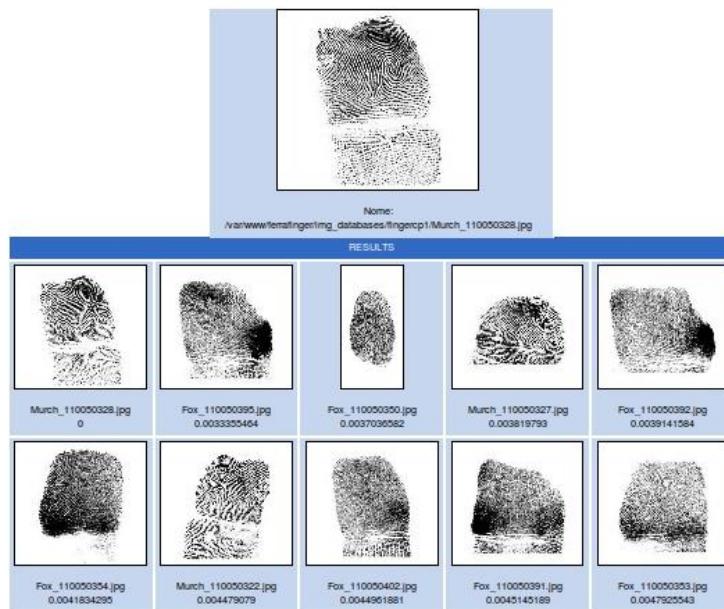


Figure 4.10: CBIR process for Figure 4.8 - part 11.

## 156 4. COMPLEX OBJECTS

of components from each DL, and the individual name (in case there is a difference between the DLs).

**Phase 5.** In phase five, the OAI-PMH protocol was used for the publishing of the individual CO metadata. It also can be used to understand which compound objects and fingerprint digital libraries are correlated to a specific individual CO. The objective is to facilitate the interchange and integration of the different fingerprint digital libraries.

Our prototype enables the installation of different image descriptors, but for the tests presented, the Statistical Analysis of Structural Information (SASI) [5] descriptor was used. The library was implemented in C, the DCCs in Java. The functions and parameters available for each DCC are described in the PostgreSQL database.

The image COs are published using the jOAI software (available at <http://www.dlese.org/dds/services/joai>). The jOAI data provider allows XML files from a file system to be exposed as items in an OAI data repository and made available for harvesting by others using the OAI-PMH.

### 4.4.4 RESULTS

The preliminary results on the fingerprint DL exploration included [353]: (i) an Entity-Relationship Diagram design; (ii) the implementation of the skin distortion model; (iii) testing of the blurring distortion; (iv) the description of NFIQ quality internal steps; and (v) an initial exploration of concepts that will be analyzed from the CO perspective; Though

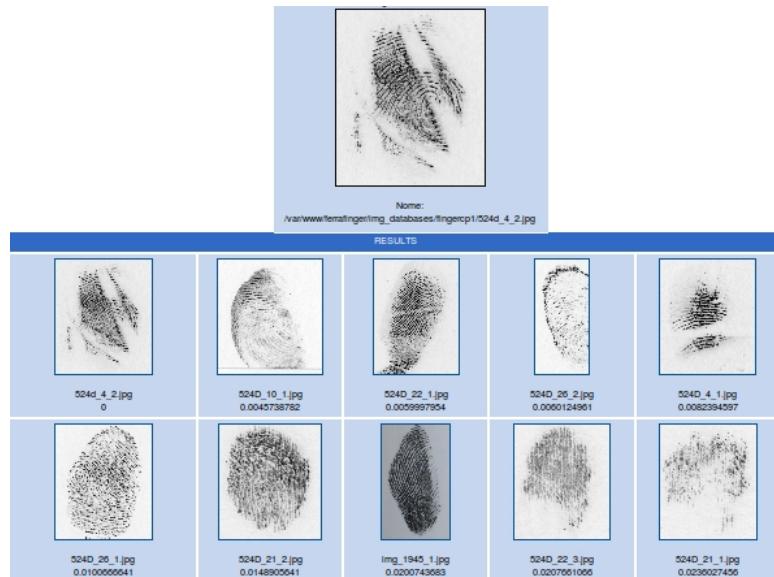


Figure 4.11: CBIR process for Figure 4.9 - part 3.

#### 4.4. CASE STUDY: FINGERPRINTS 157

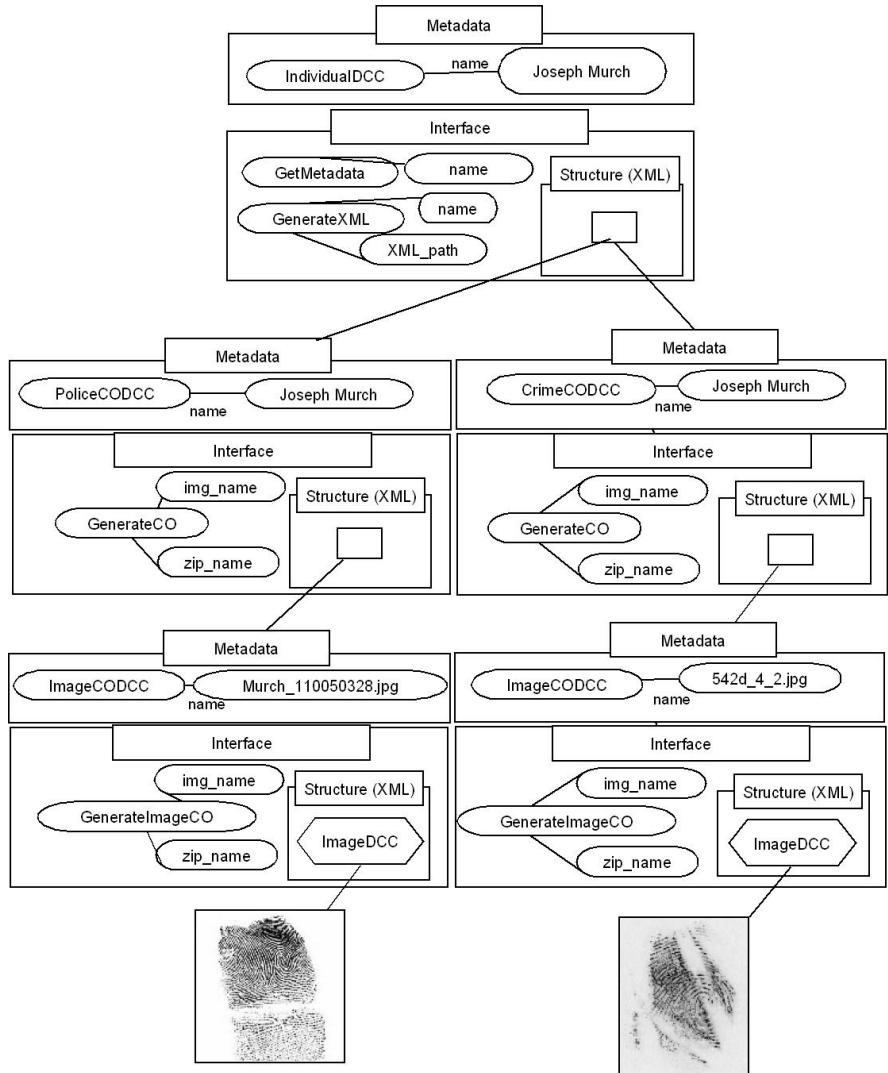


Figure 4.12: Structure for IndividualDCC.

our project is in an early development stage, these preliminary results were important to highlight the amount of information and details we need to manage, guiding us to explore the overall system by using a very large DL approach.

In summary, our case study explored the main concepts for COs using the formalization, later implemented using DCCs and OAI-PMH.

## 158 4. COMPLEX OBJECTS

```
<?xml version="1.0" encoding="UTF8"?>
<individual>Joseph Murch
  <individual_name>Joseph Murch</individual_name>
  <individual_age>22</individual_age>
  <individual_sex> M</individual_sex>
  <image_DL_indiv>Joseph Murch
    <image_DL>Police Prints Digital Library</image_DL>
    <image> Murch_110050328.jpg
      <image_name>Murch_110050328.jpg</image_name>
      <image_feature_vector_name>/home/nadiapk/data/fv/Murch_110050328.jpg.txt</image_feature_vector_name>
      <image_descriptor>SASI
        <image_name> Murch_110050328.jpg <image_dist_value>0</image_dist_value></image_name>
        <image_name>Fox_110050395.jpg<image_dist_value>0.0033</image_dist_value></image_name>
        <image_name>Fox_110050350.jpg<image_dist_value>0.0037</image_dist_value></image_name>
        <image_name>Murch_110050327.jpg<image_dist_value>0.0038</image_dist_value></image_name>
        <image_name>Fox_110050392.jpg<image_dist_value>0.0039</image_dist_value></image_name>
      </image_descriptor><image></image></image_DL_indiv>
  <image_DL_indiv>Joseph Murch
    <image_DL>Crime Scene Digital Library</image_DL>
    <image> 524d_4_2.jpg
      <image_name>524d_4_2.jpg</image_name>
      <image_feature_vector_name>/home/nadiapk/data/fv/524d_4_2.txt</image_feature_vector_name>
      <image_descriptor>SASI
        <image_name> 524d_4_2.jpg <image_dist_value>0</image_dist_value></image_name>
        <image_name>524D_10_1.jpg<image_dist_value>0.0045</image_dist_value></image_name>
        <image_name>524D_22_1.jpg<image_dist_value>0.0059</image_dist_value></image_name>
        <image_name>524D_25_2.jpg<image_dist_value>0.0060</image_dist_value></image_name>
        <image_name>524D_4_1.jpg<image_dist_value>0.0082</image_dist_value></image_name>
      </image_descriptor>
    </image>
  </image_DL_indiv><individual>
```

Figure 4.13: XML for the individual aggregation.

## 4.5 SUMMARY

Many digital library implementations and applications demand additional and advanced services to effectively reuse and aggregate different resources. Examples of commonly required services include those related to the support of newer, more complex media types such as images, multimedia objects, and related information.

In this paper we address the formal definitions and descriptions for Complex Objects. The proposed extensions for digital library functionality take advantage of formalization to understand clearly and unambiguously the characteristics, structure, and behavior of the main concepts related to components, technologies, and applications. Later these definitions are explored in a case study, to exemplify how CO concepts can be explored to define the complex image object. Our contribution relies on (i) the formalization of complex objects; (ii) the initial analysis of three CO technologies; and (iii) a case study discussion on how to handle complex image objects in applications. The set of definitions may impact future development efforts of a wide range of digital library experts since it can guide the design

and implementation of new digital library services based on complex objects, and image content. Another straightforward benefit of this work is the use of these formal definitions to construct applications, including requirements gathering, conceptual modeling, prototyping and code generation, similar to initiatives presented in [251, 698, 362]. As an example, consider the use of 5S formal theory to integrate an archaeological digital library, using applications such as 5SGraph [698]. From the implementation perspective, COs can also be used for service reuse and combination [349].

There are several research efforts that can be explored to further extend our current work. These include the study of the impact of COs on other 5S constructs, the analysis of additional features in OAI-ORE (such as proxies), the comparison and interaction with other technologies (such as the use of metadata in METS, TEI, and Dublin Core), and the use of COs in other domains (such as biodiversity information systems) and specific services (such as content-based image retrieval and annotation).

## **4.6 EXERCISES AND PROJECTS**

1. Pick your favorite digital library. Identify 3 different types of complex objects that are important in that DL.

# Integration

by Rao Shen

*Abstract:* In this chapter, 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. More specifically, we develop a minimal metamodel for archaeological DLs within the 5S theory. We implement the 5SSuite tool 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, each of which plays an important role in DL integration. We also propose an approach to integrated DLs based on the 5S formalism, which provides a systematic and functional method to design and implement DL exploring services.

## 5.1 INTRODUCTION

### 5.1.1 THE DIGITAL LIBRARY INTEGRATION 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. Over two decades 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 [284]:

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 entities);
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.

## 5.1. INTRODUCTION 161

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 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 hidden in such island-DLs. The needs for DL integration are well known, and better known than the solutions [364].

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 [212]. Hence, an integrative theory for DL is needed. [254] 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 (see Chapter 1) 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” [250].

DL integration can be done at different levels, e.g., information level and service level. Integrated information makes distributed collections of heterogeneous resources appear to be a single union collection. Integrated services provide 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.

Developing an infrastructure to address all perspectives of the DL integration problem is an ambitious task. While many efforts have looked into the DL integration problem, most developed their own approaches in an ad hoc and piecemeal fashion. In this chapter, we formalize the DL integration problem and describe 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.

## 162 5. INTEGRATION

### 5.1.2 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 a set of indicators and metrics? What are those? How well does the integration work in practice?

## 5.2 RELATED WORK

Interoperability is the most important issue when integrating heterogeneous DLs [6, 495, 532]. It has many dimensions [495, 494] and has been the subject of many initiatives. It is a broad problem domain. Typically it has been 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) [505].

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. [495] have categorized many of the prevalent approaches and have provided an informative discussion of the challenges inherent in creating interoperable DLs of 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., with Java applets).

To achieve DL interoperability requires agreement to cooperate at three levels: technical, content, and organizational [26]. 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 [497]: 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 [497]. However, establishing semantic interoperability among heterogeneous information sources from various DLs continues to be a critical issue. The NSF Post Digital Libraries Futures Workshop [369] identified it as being of primary importance in digital library research. DELOS WP5 [503] reported many issues relating to semantic interoperability in DLs. We present below related work concerning semantic interoperability in DLs.

### 5.2.1 SEMANTIC INTEROPERABILITY IN DIGITAL LIBRARIES

Semantic interoperability in DLs means the capability of different information systems to communicate information consistent with the intended meaning [503]. 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.

### 5.2.2 INTEGRATED SERVICES

There are some related works on integrating services in DLs. Some integrate searching and browsing while others integrate searching and browsing with other services. For example, CODER [196], 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 [197], an indexing, search, and retrieval system optimized for digital libraries, was developed in the 1990s; ODL [614], a system built as networks of extended open archives, was developed in 2001.

A synergy between searching and browsing is required to support users information-seeking goals [49, 50, 233, 249, 418]. Text mining and visualization techniques provide DLs additional powerful exploring services, with possible beneficial effects on searching and browsing. Thus, 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 [213][10]. CitiViz, a visual interface to CITIDEL, combines searching, browsing, clustering, and information visualization [328].

In recent years, major Web search engines have extended their services to include search on specialized subcollections or verticals focused on specific domains (e.g., news, travel, and local search) or media types (e.g., images and video). They can include summaries of relevant vertical results in web results when a user issues a query. In the research community, this is referred to as aggregated search or federation.

## 5.3 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 provide a brief review of 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.

### 5.3.1 BACKGROUND ON THE 5S FRAMEWORK

Sections 1.8 and 1.9 provide a formal framework for the DL field, summarized in Fig. 1.22. A “minimal digital library” (Def. 24 in Section 1.9, shown at the bottom right) was defined as the highest level concept. Fig. 5.1 extends this to cover digital libraries for archaeology, as discussed in Chapter 2, but omits the top layer of definitions (specified in Appendix A), regarding mathematical foundations (e.g., graphs, sequences, and functions). Fig. 5.1 begins with the 5 Ss (Streams, Structures, Spaces, Scenarios, and Societies), and key concepts of a DL (e.g., digital object, collection), and then adds in about archaeological objects and other essential concepts identified as we built the ETANA-DL [539, 537]. Arrows represent dependencies, indicating that a concept is formally defined in terms of previously defined concepts that point to it.

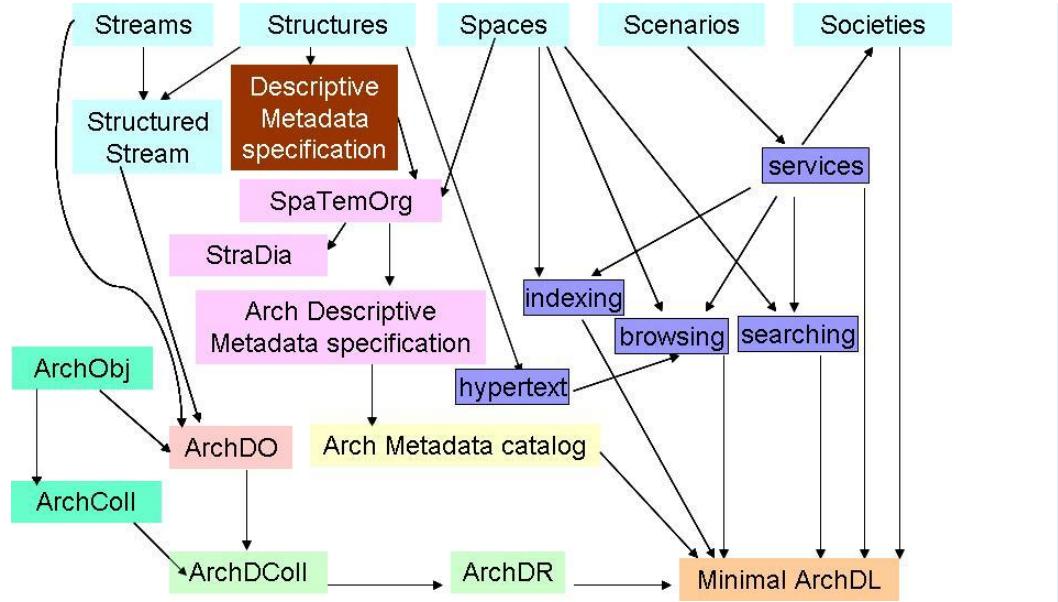


Figure 5.1: 5S definitional structure extended for archaeology

### 5.3.2 NOTATION AND DEFINITIONS

**Notation:** Let  $DL_1, DL_2, \dots, DL_i, \dots, DL_n$  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  $C_i = \bigcup_{j=1}^m C_{ij}$ , 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 [254] 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 5.1** A Union Repository (**UnionRep**) of  $n$  DLs ( $DL_1, \dots, DL_n$ ) is a DL repository ([254]) with a  $getDL\_Id$  function:  $UnionRep = (CollSet, getDL\_Id, get, store, del)$ , where

- 1)  $CollSet \subseteq 2^{\{UnionC\}}$  ;
- 2)  $getDL\_Id : UnionC \rightarrow \{Id_1, Id_2, \dots, Id_i, \dots, Id_n\}$  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  $\tilde{C} - get(h)$ ;

## 166 5. INTEGRATION

**Definition 5.2** A Union Catalog  $\mathbf{UnionCat} = DM_{UnionC}$  is a metadata catalog for  $UnionC$ .

**Definition 5.3** Minimal Union Services ( $\mathbf{MinUnionServ} = \{harvesting, mapping\} \cup (\bigcup_{i=1}^n Serv_i)$ ).

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 [254]; the mapping is defined as follows (see Def. 5.4-5.7):

**Definition 5.4** A schema is a structure [254] with a domain  $D$  of data types (e.g., strings, numbers, dates, etc.).  $\mathbf{schema} = ((V, E), L, F, D, M)$ , 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 5.5** Given a schema  $((V, E), L, F, D, M)$ , its element set  $= (v, F(v))(e, F(e))$ .

**Definition 5.6** 1-1 mapping

Let  $S$  and  $T$  be two element sets, of  $S\_Schema$  and  $T\_Schema$ , respectively. 1-1 mapping is a function:  $M_{1-1} : S \times T \rightarrow Sim$ , 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 5.7** 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 5.8** A Union Society  $\mathbf{UnionSoc} = \bigcup_{i=1}^n Soc_i$

**Definition 5.9** A Minimal Union Digital Library integrated from  $n$  DLs is given as a four-tuple:  $\mathbf{MinUnionDL} = (R_{union}, DM_{union}, Ser_{union}, Soc_{union})$ , where  $R_{union}, DM_{union}, Ser_{union}, Soc_{union}$  are Union Repository, Union Catalog, Minimal Union

Services, and Union Society. A Union DL is a superset of a **MinUnionDL**. “Integrated DL” and “Union DL” will be used interchangeably in this paper.

**Definition 5.10** DL Integration Problem Definition

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

### 5.3.3 ARCHITECTURE OF AN INTEGRATED DL

As above (Def. 5.9), 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 is harvesting. 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. The main disadvantage is that retrieval of digital objects relies on remote DLs. CITIDEL [513] 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 partially decentralized union repository may store the digital objects that will not be changed frequently. The architecture of ETANA-DL [539, 537, 538] consists of a centralized catalog and partially decentralized repository. As shown in Fig. 5.2, ETANA-DL integrates several DLs:

1. Member DLs of ETANA-DL
2. Architecture of ETANA-DL, with centralized catalog and partially decentralized repository

168 5. INTEGRATION

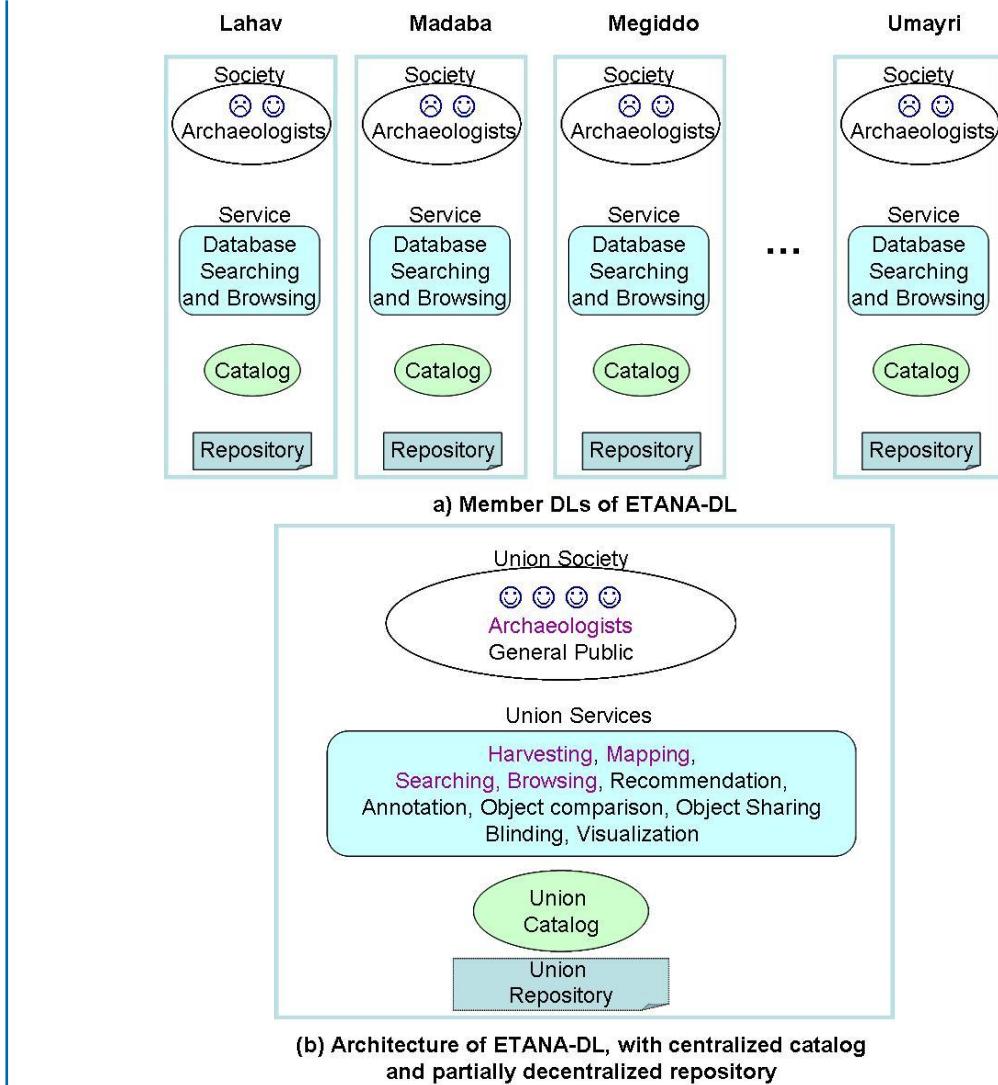


Figure 5.2: An example of an integrated DL: ETANA-DL

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 [599] is an example of such an integrated service. It provides a visual interface to ETANA-DL.

## 5.4. CASE STUDY: AN ETANA-DL EXPERIENCE 169

Search results can be classified by predefined classes. Grouped documents are displayed in several ways to help browsing.

The union services illustrated in Fig. 5.2 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.

### 5.3.4 INTEGRATION TOOLKIT: 5SSUITE

The 5S framework allows a new approach to DL development (see Fig. 5.3). 5SGraph [698, 699] supports analysis and specification, while 5SGen [332] 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 5SSuite to cover the process of union DL generation, including requirements gathering (see Fig. 5.3 step 1), conceptual modeling (see Fig. 5.3 step 2), rapid prototyping (see Fig. 5.3 step 3), and code generation (see Fig. 5.3 step 4 and Fig. 5.4). 5SSuite consists of 5SGraph, 5SGen, and SchemaMapper (described in Section 5.4.2), 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 updating of the union catalog. The complexity of the mapping and updating processes 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. Further, there may be users added who administer and/or assess the quality of the integration processes.

To integrate domain specific DLs, a metamodel for that particular domain needs to be developed based on the 5S formal theory. Section 5.4.2 describes an archaeological DL (ArchDL) metamodel and the use of 5SGraph to model ArchDLs.

## 5.4 CASE STUDY: AN ETANA-DL EXPERIENCE

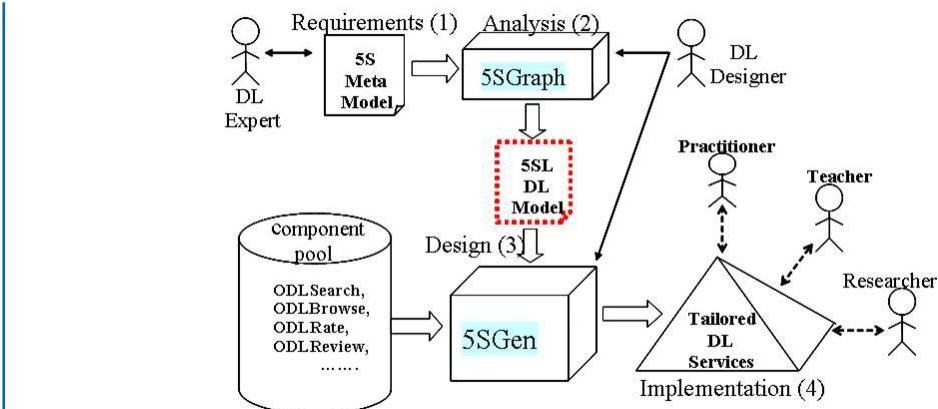


Figure 5.3: 5S related tools and their use in developing DLs [250]

#### 5.4.1 MODELING OF DOMAIN SPECIFIC DIGITAL LIBRARIES WITH THE 5S FRAMEWORK

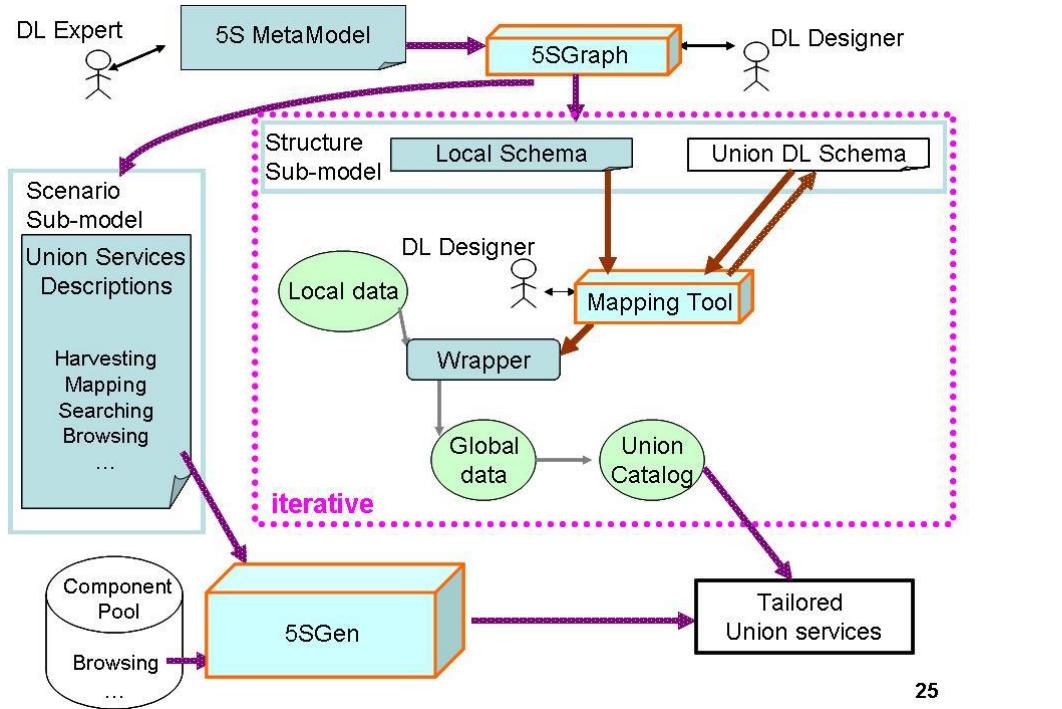
Requirements gathering and conceptual modeling are essential for the customization of digital libraries (DLs), to help attend the needs of target communities. In this section, we show how to apply the 5S (Streams, Structures, Spaces, Scenarios, and Societies) formal framework to support both tasks. The 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.

##### **Modeling of Domain Specific Digital Libraries with the 5S Framework**

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

#### 5.4. CASE STUDY: AN ETANA-DL EXPERIENCE 171



25

Figure 5.4: 5S related integration toolkit and process

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?

## 172 5. INTEGRATION

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 [254]) 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 [559]. 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

#### **5.4. CASE STUDY: AN ETANA-DL EXPERIENCE 173**

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

## 174 5. INTEGRATION

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 [187]. Spatial and stratigraphic relationships among archaeological items can be regarded as extrinsic attributes (inter-item relationships) [585]; 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.

#### Application of 5S to Archaeological DLs

With key requirements for ArchDL summarized in the previous section and in [596], 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 Chapter 1 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 (see [596]).

In this section we use the 5SGraph tool [698, 699] to specify two of the archaeological information systems of ETANA projects (<http://www.etana.org/>).

Fig. 5.5 illustrates use of 5SGraph to specify the Nimrin archaeological site, focusing on Structure, drawing upon a meta-model for archaeology that we have built for ETANA-DL [539, 537, 538]. 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. 5.6, while ETANA-DL has eight main services (see Fig. 5.7). In Section 5.4.2, we present how to integrate various structure models into the one for the union DL using the visual mapping tool.

#### 5.4.2 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

#### 5.4. CASE STUDY: AN ETANA-DL EXPERIENCE 175

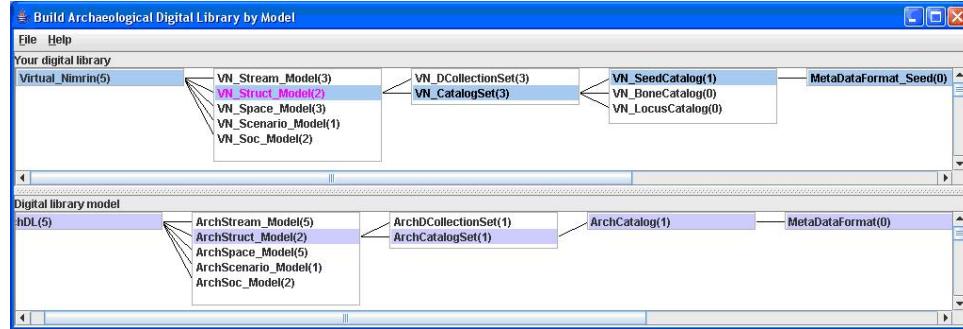


Figure 5.5: Structure model for Nimrin

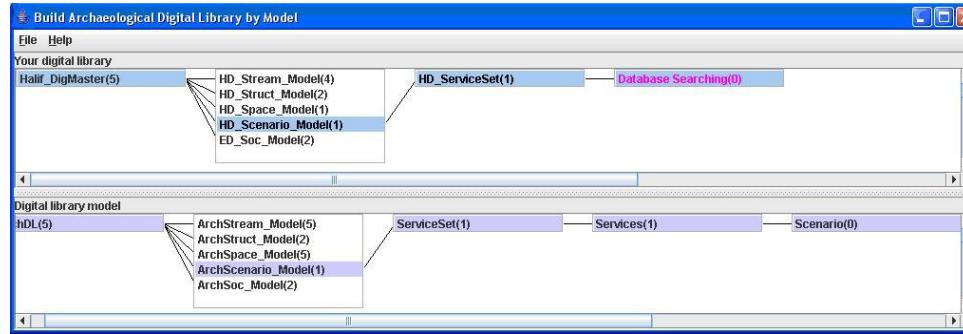


Figure 5.6: Scenario model for Halif

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 section, we present an incremental approach through intermediary- and mapping-based techniques and a visual mapping tool, SchemaMapper.

#### Features of SchemaMapper

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

1. Algorithmic perspective

## 176 5. INTEGRATION

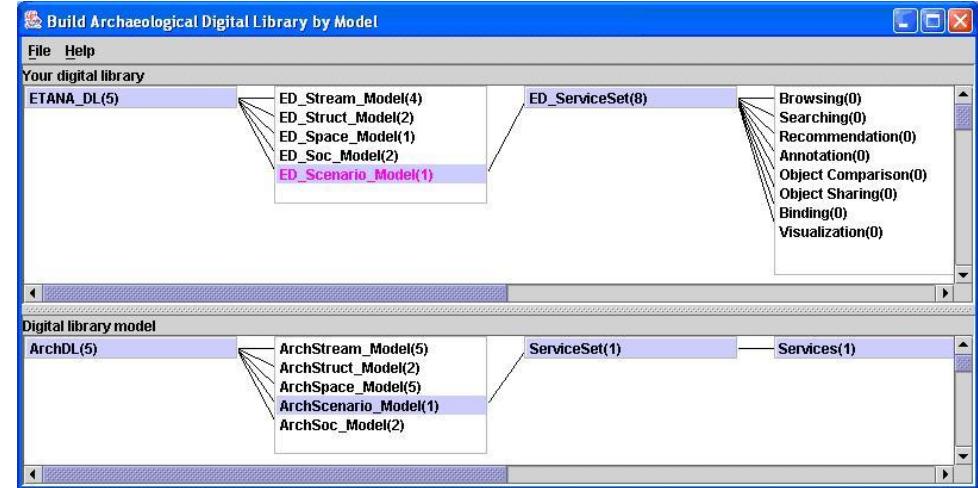


Figure 5.7: Scenario model for ETANA-DL

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 [530, 531]. 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 of Fig. 5.8). 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. 5.8. 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 ([http://www.altova.comproducts\\_mapforce.html](http://www.altova.comproducts_mapforce.html)) 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

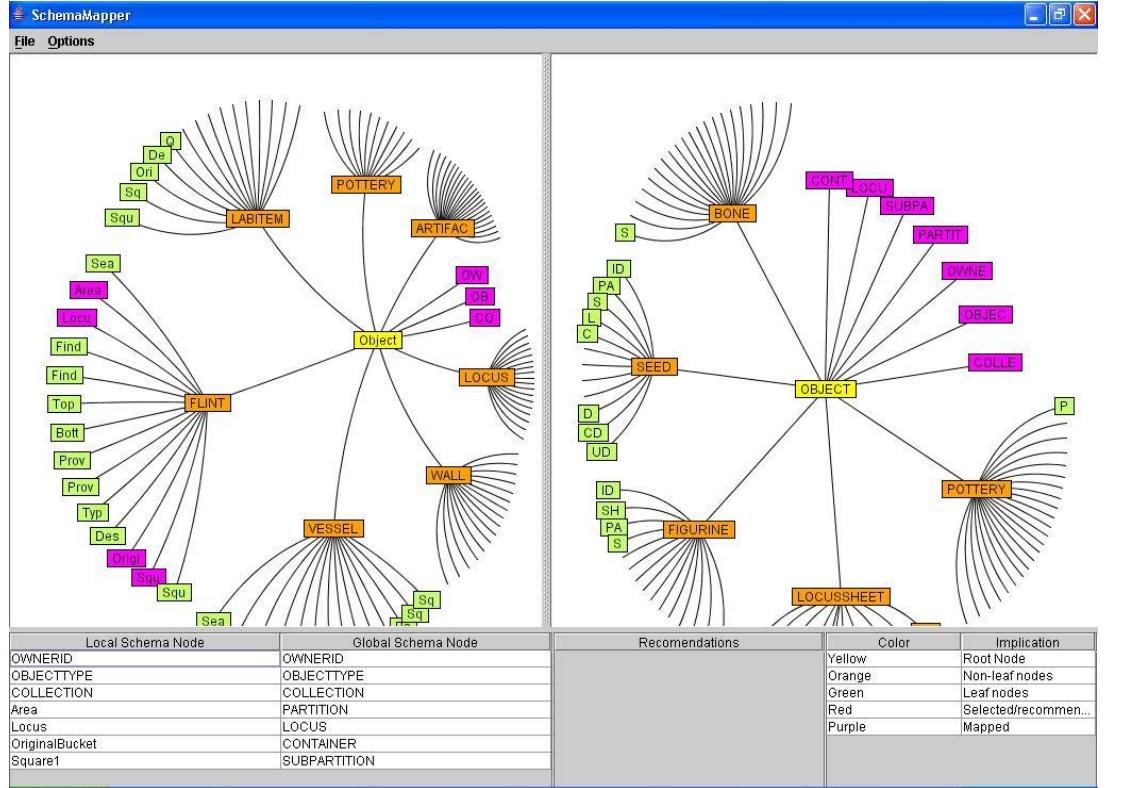


Figure 5.8: Initial set of mappings for flint tool based on rules and name-based matching

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.

### 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

## 178 5. INTEGRATION

it impossible to foresee an ultimate global schema for the description of all excavation data [187]. Accordingly, an “incremental” 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.

### Megiddo overview

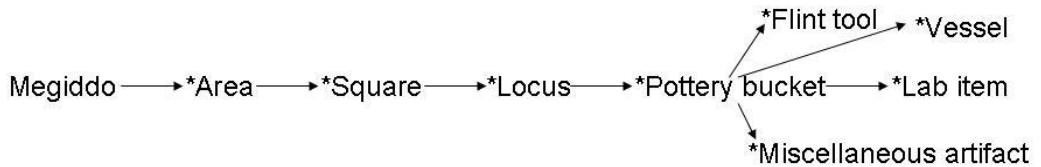


Figure 5.9: Megiddo site organization

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. 5.4) within the 5S framework. Structures represent the way archaeological information is organized along several dimensions; it is spatially organized, temporally sequenced, and highly variable. The Megiddo site organization is shown in Fig. 5.9.

### 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. 5.8 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, LOCUSHEET, 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.

#### 5.4. CASE STUDY: AN ETANA-DL EXPERIENCE 179

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.)

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. 5.8).

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. 5.10). 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. 5.10). 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.

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. 5.10). 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. 5.11 and Fig. 5.12). 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.

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 then 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

## 180 5. INTEGRATION

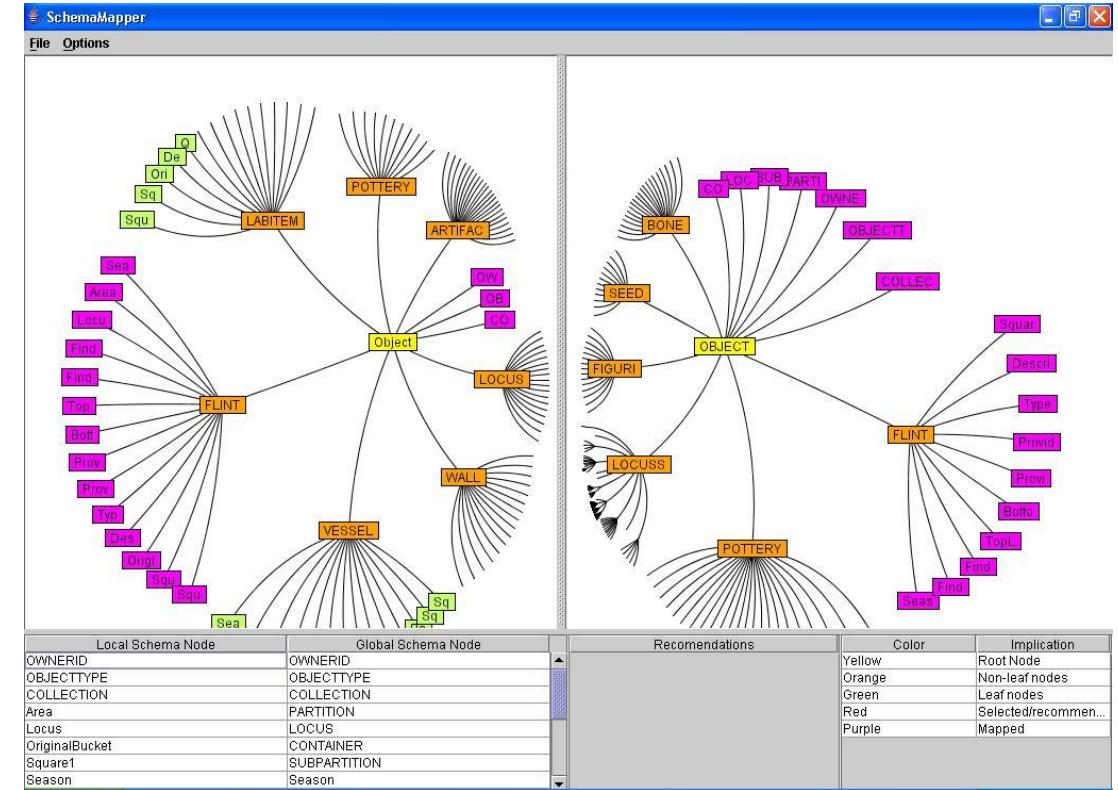


Figure 5.10: Adding FLINT sub-tree as a child of OBJECT in the global schema

consistent, Schema Mapper recommends the user to change the name of the Description node in the VESSEL sub-tree to DESCRIPTION (see Fig. 5.12). 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.

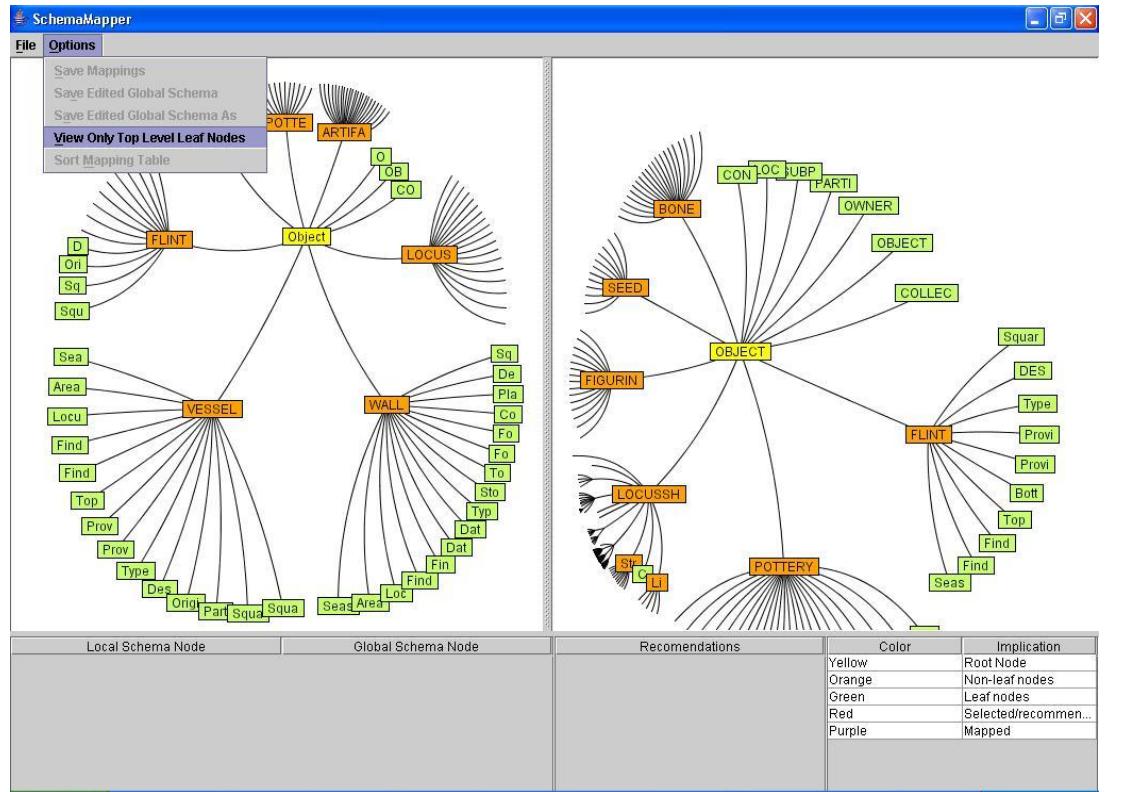


Figure 5.11: Using the View Only Top Level Leaf Nodes option mapping Vessel Collection

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.

## 5.5 SUMMARY

(ADD)

## 5.6 EXERCISES AND PROJECTS

1. Which of the integration problems described in this chapter are exclusive to digital libraries, and not commonly found when integrating databases?

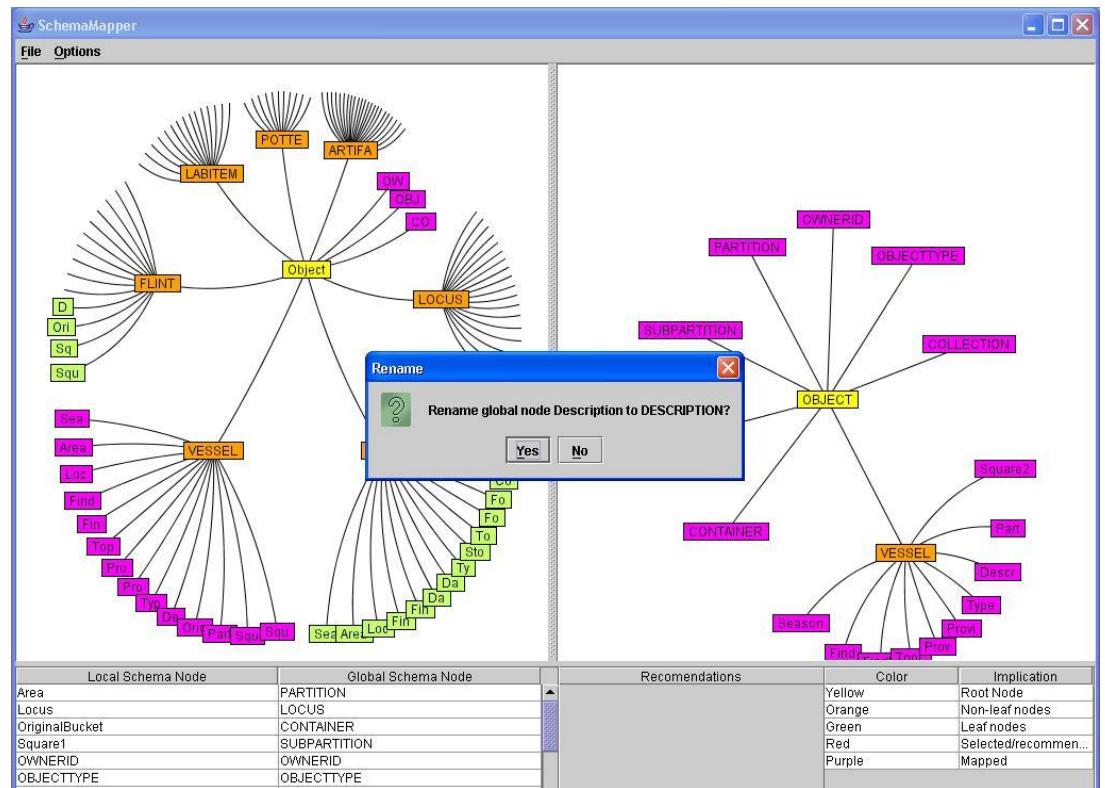


Figure 5.12: Name change recommendation based on mapping history

## CHAPTER 6

# Subdocuments

by Uma Murthy, Lois M. Delcambre, Ricardo da Silva Torres, and Nádia P. Kozievitch

*Abstract:* Subdocuments lead us in the reverse direction from complex objects.

## 6.1 INTRODUCTION

Many scholarly tasks involve working with fine-grain information or information that is part of some larger unit. Consider the following examples:

- A student reviews his notes that might refer to highlighted portions in textbooks and papers while studying for an exam.
- A doctor carefully examines her medical notes, along with marked-up X-rays of a patient's shoulder, to check for a fracture.
- A microbiologist analyzes and compares a newly found strain of bacteria with marked-up images and descriptions of similar strains in order to make deductions about it.
- A music professor develops a multimedia lecture on a musical style, combining snippets of compositions of the said style.

The student, the doctor, the microbiologist, and the musician are all working with *subdocuments*, or fine-grain information, *in situ*, or in their original information context. For the student, the subdocuments are highlighted portions of the textbooks and papers and the student views those subdocuments in the context of full text of the books and papers; for the doctor, the subdocument is the portion of the X-ray indicated by markings on the shoulder and the doctor views that subdocument in the context of the entire X-ray; for the microbiologist, the subdocuments are parts of images of the new bacteria strain as well as parts of images of previously-known strains in the context of the entire organism; and for the music professor, the subdocuments are the snippets of musical compositions (in the context of entire compositions) that the professor chooses to include in his presentation.

Working with subdocuments is an important part of such scholarly tasks, and sometimes a necessary one. For example, while comparing the bacterial strains, the microbiologist might *need to* look at the similar/different distinguishing features among the strains. The prevalence of such use of subdocuments has been noted in past studies of use of scholarly materials. For example, in a study of annotations on 150 college textbooks [420], Marshall

## 184 6. SUBDOCUMENTS

found that notes, symbols, etc. are often found near highlighted portions of text. In another study on the use of National Science Digital Library (NSDL) educational resources in creating instructional materials [540], Recker and Palmer found that teachers preferred to use resources at a smaller granularity level than which was catalogued for NSDL.

Subdocument information might be of varying content types and formats and might be distributed across locations, media, and devices. Current approaches to working with subdocuments include a combination of paper-based and digital techniques. For example, a student might have class notes, images, audio lectures, etc. in digital form and refer to textbooks, personal notes and drawings, in paper form. The student might be able to manage this fragmentation of information across various sources if the volume of information is not too large. However, the combination of large volumes of heterogeneous data in varied formats (marked-up images, notes, websites, etc.) coupled with manual information organization and retrieval, can lead to ineffective and inefficient task execution (verified by the findings in various user studies [453, 452, 450]). Two key problems are:

- Subdocuments and whole documents are dispersed across several places (paper and digital), making information management, searching, browsing, and access tedious.
- Capabilities to work with subdocuments (*in situ*) and whole documents are also dispersed across tools (storing in one place, taking notes in another, searching for information in a third place, etc.), leading to ineffective and inefficient task execution.

A digital library (DL) is an information system, with collections of documents/digital objects<sup>1</sup> and their metadata, and services including those to manage, organize, access, browse, index, and search through those collections, in order to support one or more user communities<sup>2</sup>. However, most DLs provide limited support to work with subdocuments. We use the following scenario to illustrate such a need in scholarly tasks. Consider a fisheries student, who is trying to identify the species of a fish using an image (Figure 6.1). She looks at the fish in the image and identifies the family of the fish<sup>3</sup>. Then, to help with identifying the species of the fish, she might use this image to frame a query, such as:

*Find species that are darters that have a dorsal fin that looks like Figure 6.1-A, and which is connected to another dorsal fin that looks like Figure 6.1-B , and that have an orange hue on the species' belly like Figure 6.1-C.*

Typically, support for such searches (as that mentioned above) is not present in DLs and, to the best of our knowledge, there is no facility for identifying or distinguishing subdocuments of interest from their enclosing documents. Further, there is no provision for a subdocument to have its own metadata. As a result, subdocuments are not separately accessible, searchable, or manageable in most DLs. This motivated us to define and develop

<sup>1</sup>Information in a digital library might be manifested in the form of digital objects of various content types.

<sup>2</sup>This is an informal description of a DL. There are other definitions; see Cgaot 1

<sup>3</sup>Fishes might follow a taxonomical classification, consisting of families, where each family consists of genera, and where each genus consists of species.

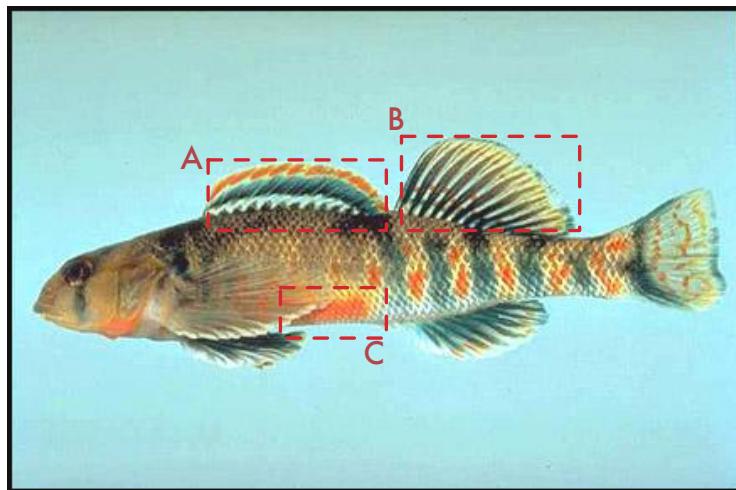


Figure 6.1: Searching on subimages and associated information (source: [313]).

a **Digital Library with Superimposed Information** (henceforth referred to as SI-DL), which combines the idea of *superimposed information* with traditional digital library services that operate in context in domains such as education.

Superimposed information (SI) refers to new information laid over subdocuments, which are part of existing information [24, 155, 408, 447]. Examples include new content such as annotations, labels, and tags; new structures/organizations such as citations, indexes, and concordances; and combination of new content and structure such as in concept maps, multimedia presentations composed from existing information, etc. A core property of an SI system is to enable working with subdocuments, while retaining its original information context, also referred to as working with information *in situ*. Thus, in an SI system, a user can select or work with information elements at subdocument level while retaining the original context.

By combining the architecture and concepts of an SI system with those in a DL, we can introduce subdocuments into a DL. This combination enables us to treat subdocuments as first-class objects in a DL, allowing DL services that apply to regular digital objects, to now apply to subdocuments. Thus, subdocuments might be managed, organized, accessed, indexed, searched for, browsed for, and used in a way similar to digital objects. This opens numerous possibilities of working with subdocuments, to support various tasks, some of which we explore in this dissertation.

The focus of this chapter is the description and definition of an SI-DL. We abstract and model the data and services in an SI-DL, by using and building upon existing abstractions and models in digital libraries and superimposed information systems. This enables

## 186 6. SUBDOCUMENTS

us to extend the notion of digital libraries in a systematic manner. We build upon DL formalization work, including that presented in Chapter 1 of this book, to extend a minimal digital library to include SI.

The rest of the chapter is organized as follows. In Section 6.2, we discuss related work in superimposed information, hypertext, annotations, and subdocuments in digital libraries. Following that, we review select definitions from Chapters 1 and 4, which we reuse and/or build upon in this chapter. In Section 6.4, we abstract the components of an SI-DL to develop an SI-DL metamodel. We formally define these components, leading to the definition of an SI-DL. Finally, in Section 6.5, we present case studies to verify the descriptive power of the SI-DL metamodel.

## 6.2 RELATED WORK

### 6.2.1 SUPERIMPOSED INFORMATION

The notion of superimposed information (SI) is foundational to this dissertation research. SI refers to new information created to reference subdocuments, or elements in existing information resources [155, 408]. SI might be created for various reasons, such as to select, highlight, reference, extend, supplement, connect, or organize subdocuments. SI can be in the form of new content (annotations or tags over few lines in a text or an object in a photograph), new structure or organization (table of contents in a book, a trailer of a movie), or new content and organization (a report, which cites information from multiple sources or a multimedia presentation developed using audio/video clips and personal notes).

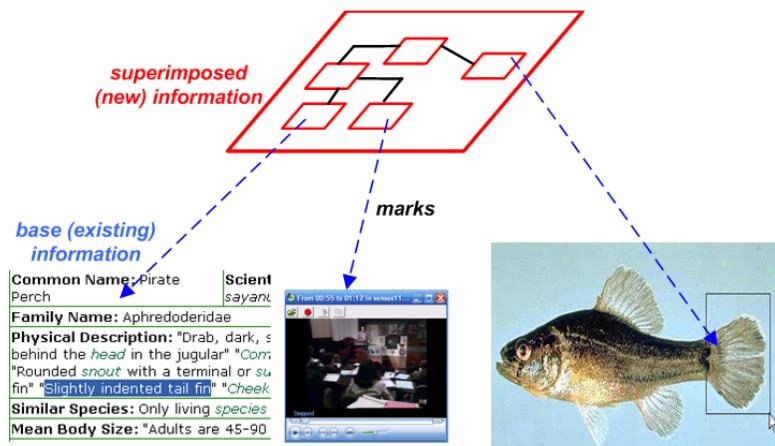


Figure 6.2: Working with information selections in situ.

Figure 6.2 shows the conceptual model of an SI system. The *base layer* consists of existing information resources and might include content in various formats and media types. A *mark* is an abstraction that specifies an addressable/reference-able region, or subdocument, in the base layer. Note that a mark is created after the creation of a base document and need not pertain to an information element as specified by the structure of the base document. New information resides in the *superimposed layer*, which consists of one or more marks, and might include new content and/or organization. A mark is created in the base layer and used in the superimposed layer, and thus, links both these layers. An important characteristic of SI systems is that they enable users to work with subdocuments *in situ*, or while keeping the original information context intact. Thus, using the notion of SI enables us to select or work with information elements at the subdocument level while retaining the original context (by referencing, not replicating, information).

Superimposed applications (SAs) allow users to lay new information over existing or base information. SAs employ marks to work with subdocuments. Prior SI work has included the development and demonstration of infrastructure for creation, resolution, and use of SI (through SAs) [153, 155, 152, 408, 447, 446, 448, 445, 449]. Notable among this work, is the Superimposed Pluggable Architecture for Contexts and Excerpts (SPARCE) [447], a middleware approach for managing superimposed information. SPARCE was used in various SAs, such as Sidepad [447], which may be used to manage and visualize marks and SIMPEL [451], a multimedia presentation editor and player.

More recently, Archer et al. defined and demonstrated an architecture for representing marks (i.e., subdocuments) as first-class objects in a DL [24]. Their work leveraged the DSpace DL system<sup>4</sup> and the Fab4 browser<sup>5</sup> [127], a derivative of the Multivalent browser [514, 515] with annotation capability. They demonstrated the same capability in the Fedora DL system<sup>6</sup>.

Most of the earlier SI work has been on development of infrastructure and systems. In this dissertation research, we augment the SI literature through multiple contributions. In general, combining the notion of SI with DLs provides a way to deal with collections of subdocuments and associated information. The SI-DL metamodel provides a formal representation of SI in a DL environment, including precise definitions of SI concepts.

### 6.2.2 SUBDOCUMENTS AND HYPERTEXT

In the Hypertext world, the notion of subdocuments is included. Typically, these are author-created parts of documents. In HTML, the #-tag is used to point to a specific part of a document [499].

<sup>4</sup><http://www.dspace.org>

<sup>5</sup><http://bodoni.lib.liv.ac.uk/fab4/>

<sup>6</sup><http://fedora-commons.org/>

## 188 6. SUBDOCUMENTS

Hypertext reference models such as the Dexter model [274] make explicit the concepts (such as linking) in hypertext systems. Hypermedia models such as the Amsterdam model [278] extend the hypertext notion of links to time-based media and compositions of different media. Obviously, SI is rooted in these ideas. The main additional capability it brings is being able to work with subdocument information *in situ*. There is limited support in hypertext models and systems to work with information *in situ*. In standards such as XLink and XPath, subdocuments may typically be referenced as long as they have been predefined by the author, or if they are encompassed within XML tags [629, 630]. SI enables linking information at varying document granularities after a document has been created, thus supporting reader-created subdocuments along with author-created subdocuments.

Ted Nelson's Xanadu system presented two ideas – deep content links and transclusion. Both of these, he felt, describe his vision of hypertext (connected, networked documents), more than what the World Wide Web implemented [465]. Marks being viewed in their context are very similar to the idea of transclusion, where quotations and annotations may be connected to subdocuments in their original context.

Work by Kerne et al., on recombinant information and hypersigns, focuses on developing compositions for visual semiotics (to construct and to understand new meanings) supporting personal expression to promote the creative process and information discovery [335, 336, 337]. This can be considered an application or type of SI. In the SI-DL metamodel, we developed a representation for such SI in a DL environment.

### 6.2.3 SUBDOCUMENTS AND SUPERIMPOSED INFORMATION IN DIGITAL LIBRARIES

Many DL systems have annotation capabilities, usually focusing on annotations of complete documents. DiLAS is a DL annotation service that brings together multiple DL annotation systems to create a framework for a decentralized annotation service [11]. Agosti and Ferro worked on various annotation projects, including conducting a comprehensive review of digital libraries and other information systems with annotation capabilities [13] and integrating annotation and search capabilities into digital libraries to annotate images of historical manuscripts [15], to develop a formal model for annotations in a DL, which provides precise specifications for annotations in a DL [14]. The focus of our work is to incorporate subdocuments and superimposed information in a digital library. Our SI-DL metamodel does not explicitly include an annotation as a component. However, an annotation might be represented as superimposed information in the metamodel.

SI relates also to the idea of secondary repositories, where users may compose structured collections of complex digital objects [544]. These objects point back to the primary digital objects (similar to base information) from which they are produced. The focus of this project [544] is to examine the role of secondary repositories in access and preservation. Secondary repositories are similar to the idea of collections of aggregate works [84],

where Buchanan, et al. discuss challenges of representing aggregate works, such as an encyclopedia, an anthology of poems, etc., in a digital library. In both works [544, 84], a unit of information is a document, as structured and specified by the document's author. The aggregate work is a new complex object composed of documents. Our work might be considered an extension in the opposite direction, where a user might create and use a subdocument, a unit of information, which is smaller than a document. Our goal is to represent such fine-grain information inside a digital library. Once we are able to treat these subdocuments as digital objects, they might be combined in the same ways as other digital objects, to form complex digital objects, or secondary works, or aggregates.

#### 6.2.4 SUBDOCUMENTS AND ANNOTATIONS

There has been considerable work done on annotation standards [325] and frameworks [40, 514, 515]. The Open Annotation Collaboration (OAC) project [474, 283], launched in 2009, is working on an annotation model to enable interoperability and communication across annotations on web-based content. The OAC annotation model focuses on enabling various annotation use cases dealing with multimedia content of heterogeneous formats.

### 6.3 REVIEW OF SELECT DEFINITIONS

We review select definitions in the 5S framework and build upon these and other DL and set theory concepts to yield SI-DL concept definitions. First, we review the definitions of a stream, structure, system state, event, and scenario. These definitions are used in subsequent definitions in the 5S framework and in the definition of SI-DL concepts. Next, we review the definitions of a digital object, descriptive metadata specification, metadata format, hypertext, and a minimal digital library. For further details of these definitions and for other definitions, readers are referred to Chapters 1 and 4 of this book.

**Definition 6.1** A **stream** is a *sequence* whose codomain is a nonempty set. Examples of a stream might be sequences of bits, characters, and numbers.

**Definition 6.2** A **structure** is a tuple  $(G, L, \mathcal{F})$ , where  $G = (V, E)$  is a directed graph with vertex set  $V$  and edge set  $E$ ,  $L$  is a set of label values, and  $\mathcal{F}$  is a labeling function  $\mathcal{F} : (V \cup E) \rightarrow L$ . Thus, a structure might be considered as a graph, with the vertices and edges labeled.

**Definition 6.3** A **system state** (from now on, just state) is a function  $s : L \rightarrow V$ , from labels  $L$  to values  $V$ . A **state set**  $S$  consists of a set of state functions  $s : L \rightarrow V$ .

## 190 6. SUBDOCUMENTS

**Definition 6.4** A **transition event** (or simply **event**) on a state set  $S$  is an element  $e = (s_i, s_j) \in (S \times S)$  of a binary relation on state set  $S$  that signifies the transition from one state to another. An event  $e$  is defined by a *condition* function  $c(s_i)$  which evaluates a Boolean function in state  $s_i$  and by an *action* function  $p$ .

For example, consider two system states. In the first one,  $s_i$ , there is an image collection with  $k$  images. An event,  $e$ , might be the uploading of a new image to the collection, yielding a new system state,  $s_j$ . The condition function,  $c(s_i)$ , might be a test for the adequate availability of space in the collection for the new image. The action function,  $p$ , is to upload the new image in the collection. A variable (or logical *location*),  $X$ , such as the size of the collection, might be different in the two states. It might be represented as  $s_i(X)$  and  $s_j(X)$  in the two states, with values  $k$  and  $k + 1$ , respectively.

**Definition 6.5** A **scenario** is a sequence of related transition events  $\langle e_1, e_2, \dots, e_n \rangle$  on state set  $S$  such that  $e_k = (s_k, s_{k+1})$ , for  $1 \leq k \leq n$ .

Considering the aforementioned example of uploading an image to a collection, a scenario might involve uploading an image, adding metadata about the image, and publishing the image to a blog or a website.

### Structural metadata specification

The structural metadata specifies the internal structure of a digital object and its component parts. This usually refers to author-defined parts of the digital object. For example, the author of a paper specifies the components of the paper, including paper title, sections, section titles, figures, tables, and references. The structure in this case is linear, for the most part, with elements of hypertext linking via cross-referencing.

### Descriptive metadata specification

**Definition 6.6** Let  $\mathcal{L} = \bigcup D_k$  be a set of literals defined as the union of domains  $D_k$  of simple datatypes (e.g., strings, numbers, dates, etc.). Let also  $\mathcal{R}$  and  $\mathcal{P}$  represent sets of labels for resources and properties respectively. A **descriptive metadata specification** is a structure  $(G, \mathcal{R} \cup \mathcal{L} \cup \mathcal{P}, \mathcal{F})$ , where:

1.  $\mathcal{F} : (V \cup E) \rightarrow (\mathcal{R} \cup \mathcal{L} \cup \mathcal{P})$  can assign general labels  $\mathcal{R} \cup \mathcal{P}$  and literals from  $\mathcal{L}$  to nodes of the graph structure;
2. for each directed edge  $e = (v_i, v_j)$  of  $G$ ,  $\mathcal{F}(v_i) \in \mathcal{R} \cup \mathcal{L}$ ,  $\mathcal{F}(v_j) \in \mathcal{R} \cup \mathcal{L}$  and  $\mathcal{F}(e) \in \mathcal{P}$ ;
3.  $\mathcal{F}(v_k) \in \mathcal{L}$  if and only if node  $v_k$  has outdegree 0.

The triple  $st = (\mathcal{F}(v_i), \mathcal{F}(e), \mathcal{F}(v_j))$  is called a **statement** (derived from the descriptive metadata specification), meaning that the resource labeled  $\mathcal{F}(v_i)$  has property  $\mathcal{F}(e)$  with value  $\mathcal{F}(v_j)$  (which can be designated as another resource or literal).

A descriptive metadata specification is used to describe digital objects. It might be considered as a labeled graph, specifying the relationships between resources and between a resource and its properties. The aforementioned definition of a statement is related to Resource Description Framework (RDF) [96] statements in the Semantic Web [54]. It makes use of triples to describe properties of resources, where each property of a resource is associated with a value. For example, the descriptive metadata about a journal paper might include statements such as (Journal1, ‘editor’, ‘Edward A. Fox’), (Journal1, ‘format’, ‘PDF’), (Paper1, ‘journal’, ‘Journal1’), and (Paper1, ‘creation-date’, ‘18 February 2011’).

### Metadata format

**Definition 6.7** Let  $D_{\mathcal{L}_{MF}} = \{D_1, D_2, \dots, D_i\}$  be the set of domains that make up a set of literals  $\mathcal{L}_{MF} = \bigcup_{j=1}^i D_j$ . As for metadata specifications, let  $\mathcal{R}_{MF}$  and  $\mathcal{P}_{MF}$  represent sets of labels for resources and properties, respectively. A **metadata format** for descriptive metadata specifications is a tuple  $MF = (V_{MF}, \text{def}_{MF})$  with  $V_{MF} = \{\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_k\} \subset 2^{\mathcal{R}_{MF}}$  a family of subsets of the resource labels  $\mathcal{R}_{MF}$  and  $\text{def}_{MF} : V_{MF} \times \mathcal{P}_{MF} \rightarrow V_{MF} \cup D_{\mathcal{L}_{MF}}$  is a property definition function.

A metadata format uses the property definition function,  $\text{def}_{MF}$ , to constrain the types of resources that might be associated together in statements. For example,  $\text{def}_{DC}(\mathcal{R}, ‘creation – date’) = Date$  and  $\text{def}_{DC}(\mathcal{R}, ‘format’) = AllowedTextFormats$ . In a digital library, a descriptive metadata specification conforms to a metadata format. This conformance is represented by the following definition.

**Definition 6.8** A descriptive metadata specification  $MS = (G_{MS}, \mathcal{R}_{MS} \cup \mathcal{L}_{MS} \cup \mathcal{P}_{MS}, \mathcal{F}_{MS})$  **conforms with** a metadata format  $MF = (V_{MF}, \text{def}_{MF})$  if  $\mathcal{R}_{MS} \subseteq \mathcal{R}_{MF}$ ,  $\mathcal{L}_{MS} \subseteq \mathcal{L}_{MF}$ ,  $\mathcal{P}_{MS} \subseteq \mathcal{P}_{MF}$ , and for every statement  $st = (r, p, l)$  derived from  $MS$ ,  $r \in \mathcal{R}_k$  for some  $\mathcal{R}_k \in V_{MF}$  and  $p \in \mathcal{P}_{MS}$  implies  $l \in \text{def}_{MF}(\mathcal{R}_k, p)$ .

### Digital object

**Definition 6.9** A **digital object** is a tuple  $do = (h, SM, ST, StructuredStreams)$  where

1.  $h \in H$ , where  $H$  is a set of universally unique handles (labels);
2.  $SM = \{sm_1, sm_2, \dots, sm_n\}$  is a set of streams;

## 192 6. SUBDOCUMENTS

3.  $ST = \{st_1, st_2, \dots, st_m\}$  is a set of structural metadata specifications;
4.  $StructuredStreams = \{sts_m_1, sts_m_2, \dots, sts_m_p\}$  is a set of StructuredStream functions defined from the streams in the  $SM$  set (the second component) of the digital object and from the structures in the  $ST$  set (the third component). A Structured-Stream defines a mapping from nodes of a structure to segments of a stream.

Consider the example of a journal paper as a digital object. In this case, the streams in the journal paper digital object are the text stream and the image streams. The set of structural metadata specifications represents the organizational structure of the journal paper. For example, it might contain the specifications for the title, each section, figures and captions, tables and captions, and specifications for the bibliography. The structured streams set specifies how the text and image components of the paper map to each item in the organizational structure of the paper. For example, it might specify the text stream that constitutes the title and the text and image streams that constitute a particular section.

### Hypertext

**Definition 6.10** Let  $H = ((V_H, E_H), L_H, \mathcal{F}_H)$  be a structure and  $C$  be a collection. A **hypertext**  $HT = (H, Contents, \mathcal{P})$  is a triple such that:

1.  $Contents \subseteq C \cup AllSubStreams \cup AllSubStructuredStreams$  is a set of contents that can include digital objects of a collection  $C$ , all of their streams (and substreams) and all possible *restrictions* of the StructuredStream functions of digital objects.
2.  $\mathcal{P} : V_H \rightarrow Contents$  is a function which associates a node of the hypertext with the node content.

Consider a hypertext generated by a set of digital articles on the Web, which are linked together via references and citations. The nodes of this hypertext would be all the articles, and all the *SubStreams* and the *SubStructuredStreams* within each of the articles, as defined by the author of an article. For example, a *SubStream* could be the stream of bits in an image or the sequence of characters in a paragraph or a word. Examples of *StructuredStreams* are sections, title, tables, etc. A hyperlink is a directed edge in the hypertext graph from one node to another. An examples of an anchor (source node of a hyperlink) is a phrase, which is linked to a citation or an article elaborating on that phrase.

### A minimal digital library

**Definition 6.11** A **digital library** is a 4-tuple  $(\mathcal{R}, Cat, Serv, Soc)$ , where

- $\mathcal{R}$  is a repository;
- $Cat = \{DM_{C_1}, DM_{C_2}, \dots, DM_{C_K}\}$  is a set of metadata catalogs for all collections  $\{C_1, C_2, \dots, C_K\}$  in the repository;
- Serv is a set of services containing at least services for indexing, searching, and browsing;
- Soc is a society.

Considering the Flickr<sup>7</sup> photo sharing web application as an example of a digital library, the components of this definition might be described as follows. The collections in Flickr's repository include the collection of images (different for each user or group), collection of user profiles, and the collection of group profiles. A metadata catalog associated with the image collection has a metadata record for each image, which includes information such as image title, description, tags, comments, and image notes (subimages and associated annotations). Services within Flickr include image management (adding, deleting, and other functions to manage and organize images in a user account), user management, indexing (image information), browsing, text-based search, image annotation, and tagging. Societies within Flickr include image owners, image commenters, group administrators, and group moderators. Thus, Flickr includes all the required components and can be considered a digital library.

### 6.3.1 COMPLEX OBJECTS

The notion of a complex object, defined by Kozievitch et al. [348, 355] (Chapter 4), can be considered as an addition to the original set of 5S definitions. Complex objects are single entities that are composed of multiple digital objects, each of which is an entity in and of itself [356, 366]. A complex digital object is a (simple) digital object or a recursive composition of other complex objects, as shown in Figure 6.3.

A complex digital object can be a digital object or an organization of other complex objects, therefore needing a structure to organize its components. An example of a complex object is a dissertation, which is composed of chapters, figures, and tables, where each chapter, figure, and table is a digital object.

**Definition 6.12** A complex digital object is defined as a tuple  $cdo = (h, SCDO, S)$  where

1.  $h \in H$ , where  $H$  is a set of universally unique handles (labels);
2.  $SCDO = \{DO \cup SM\}$ , where  $DO = \{do_1, do_2, \dots, do_n\}$ , and  $do_i$  is a digital object or another complex object; and  $SM = \{sm_a, sm_b, \dots, sm_z\}$  is a set of streams;

<sup>7</sup><http://flickr.com>

## 194 6. SUBDOCUMENTS

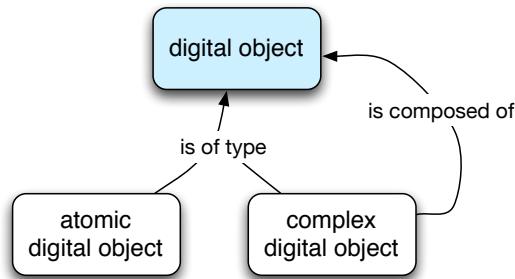


Figure 6.3: A concept map for complex object composition. (Source: [355])

3.  $S$  is a structure that composes the complex object  $cdo$  into its parts in  $SCDO$ .

A complex object is a simple digital object or a composition of other complex objects. The composition of its sub-parts (as seen in Figure 6.3) is represented by the component  $S$ .

## 6.4 FORMALIZATION AND APPROACH TO A DIGITAL LIBRARY WITH SUPERIMPOSED INFORMATION (SI-DL)

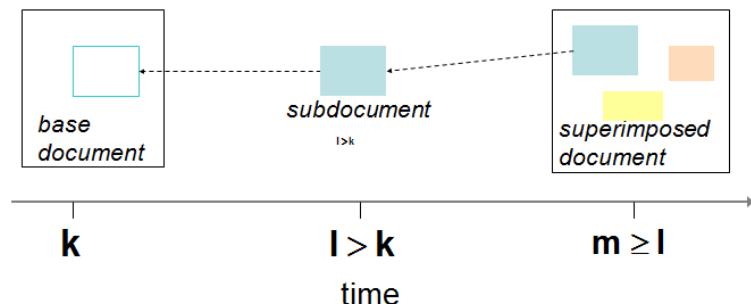


Figure 6.4: Temporal relationship among digital objects in an SI-DL.

We extend the 5S minimal digital library framework to include support for subdocuments, superimposed documents, and relevant services. In terms of content, we distinguish among three types of digital objects: 1) base document – information existing as whole documents for which subdocuments have been defined; 2) subdocument – part of a base document referenced by an address, which indicates a range or span, in the base document; and 3) superimposed document – a document, which consists of one or more subdocuments

#### 6.4. FORMALIZATION AND APPROACH TO AN SI-DL 195

and other associated information. It is important to highlight the temporal ordering that exists among the aforementioned types of digital objects, as depicted in Figure 6.4. The ordering relationship is similar to the temporal dimension of digital objects described by Agosti and Ferro in their formal model of annotations [14]. The temporal ordering states that a base document exists before a subdocument might be (marked and) created in it. Further, a subdocument exists before the creation of a superimposed document that uses this subdocument. This might be expressed by the following ordering:

$$t_{BD} < t_{sd} \leq t_{sidoc}, \text{ where:}$$

- $t_{BD}$  represents the time of creation of a base document,  $BD$ .
- $t_{sd}$  represents the time of creation of a subdocument,  $sd$ , in the base document  $BD$ .
- $t_{sidoc}$  represents the time of creation of a superimposed document  $sidoc$ , which contains the subdocument,  $sd$ .

Base documents, subdocuments, and superimposed documents have all of the ordinary properties of a digital object as well, such as having metadata associated with it and being part of one or more collections. The content of each of these digital objects and their associated metadata can be browsed, indexed, and searched, as with any other digital object. In addition to existing services, we need a new service to deal with the referencing and presentation of a subdocument in situ. We refer to this service as *view in context*. The view in context service enables a subdocument to be viewed in the original context of its containing base document.

We assume that subdocuments and all kinds of superimposed information exist in the DL along with ordinary digital objects<sup>8</sup>. The activity of creation/composition is outside the scope of these definitions just as the authoring of digital objects is generally supported by tools that are outside of the DL. Thus, creating a subdocument, annotating a subdocument or another digital object, and creating/composing a superimposed document, such as a concept map or a strand map, are all outside of the scope of our model. We are only concerned with how this information is represented in a DL and what new services will be added to access, retrieve, and facilitate viewing of information once it has been added to the DL. Note that specific superimposed applications are responsible for viewing superimposed documents and the SI-DL formalization is not concerned with those applications<sup>9</sup>.

We need to make a comment about *annotation* here. In the SI-DL metamodel, we treat a subdocument as a digital object and provide its definition. An annotation is an important part of an SI-DL since it is supplemental information associated with a subdocument. However, an annotation might be associated with any kind of digital object and is not restricted to subdocuments. In the formal model of annotation, Agosti and Ferro

<sup>8</sup>Ordinary digital objects need not be any of: a base document, a subdocument, or a superimposed document.

<sup>9</sup>In a similar way, we are not concerned about display of base documents.

## 196 6. SUBDOCUMENTS

[14] extensively explore and define the idea of annotation on digital content, of which the basic unit is a digital object. Thus, we do not formally define annotation in our metamodel. Note that in an SI-DL, an annotation might be represented as a superimposed document consisting of text or other content comprising the annotation (or link to a digital object comprising the annotation) that references a subdocument or other document, i.e., the original material in a base document that is being annotated.

Table ?? provides examples of the 5 S's in a DL and in an SI-DL. The new concepts added to a DL are as shown in Figure 6.5. The figure also shows the connection between a superimposed document and a complex object. In the remaining part of this section, we formally define the components of an SI-DL.

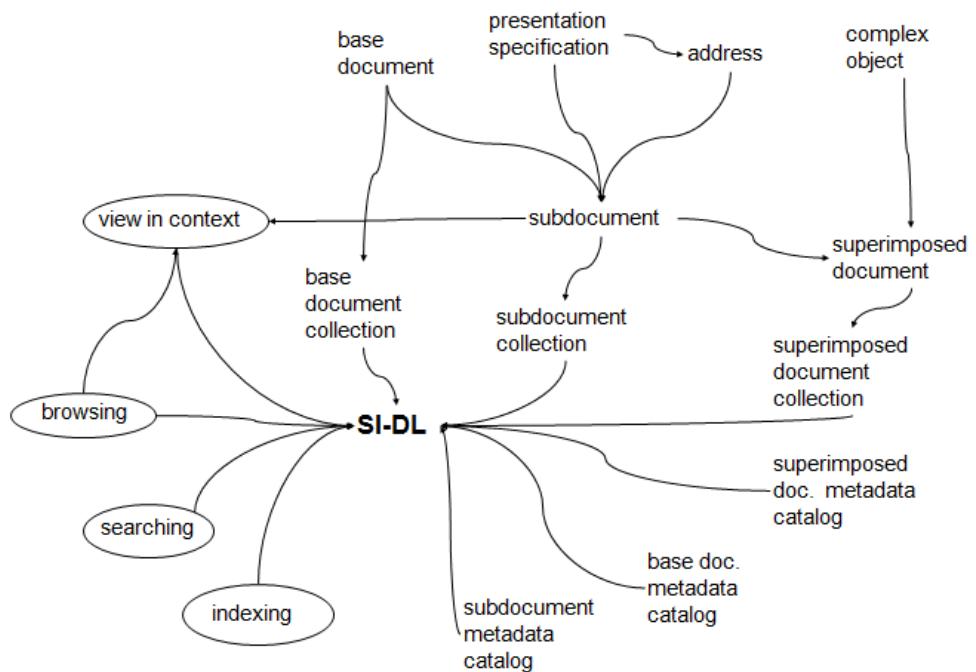


Figure 6.5: Definitional dependencies among concepts in an SI-DL.

### 6.4.1 5S EXTENSIONS

In this section, we define new concepts, which, along with the set of concepts in the 5S framework for minimal digital libraries and complex objects, are used to define and describe a digital library with superimposed information.

**Table 6.1:** Examples of the 5 S's in a DL and in an SI-DL.

Ss	Examples	Objectives	Examples in an SI-DL
Streams	Text, video, audio, and image	Describes properties of the DL content such as encoding and language for textual material or particular forms of multimedia data	(Multimedia) base document, superimposed document, subdocument, metadata about each of the aforementioned documents.
Structures	Collection, catalog, hypertext, document, and metadata	Specifies organizational aspects of the DL content	Structure of a superimposed document, structure of metadata formats, extended hypertext defined by links between a superimposed document and the subdocuments that it contains, and between a subdocument and the base document of which it is a part.
Spaces	Measurable, topological, vector, and probabilistic	Defines logical and presentational views of several DL components	Span of a subdocument in a base document, view (display) of a subdocument in the context of its containing base document.
Scenarios	Creating, searching, browsing, indexing, annotating, and recommending.	Details the behavior of DL services	Traditional DL services acting upon base documents, subdocuments, and superimposed documents and view in context.
Societies	Service managers, learners, and teachers.	Defines managers, responsible for running DL services; actors, that use those services; and relationships among them	Creator of subdocuments and superimposed documents, annotator, user of subdocuments, and administrators.

## 198 6. SUBDOCUMENTS

### Base document

A **base document**  $BD$  is a digital object for which a subdocument exists. Any digital object can thus become a  $BD$ , upon creation of the first subdocument. In Section 6.4, we described the temporal relationship among base documents, subdocuments, and superimposed documents. In the temporal relation, we represent the time of creation of a base document as  $t_{BD}$ . However, it is important to point out that a digital object becomes a base document only at  $t_{sd}$ . Before that time, it is considered as a regular digital object (as defined in the 5S framework).

### Presentation specification, address, and subdocument

In this section, we define all concepts associated with a subdocument. We build upon the definition of a substream in the 5S framework (Chapter 1) and the definition of a segment in the formal annotation model [14], to define a subdocument. According to Goncalves et al., a substream is associated with a pair of natural numbers  $(a, b), a < b$ , corresponding to a contiguous subsequence  $[S_a, S_b]$  of stream  $S$ . Or, we can say  $sm_t[i, j] = \langle a_0, a_1, \dots, a_n \rangle$ ,  $0 \leq i \leq j \leq n$  is a substream or segment of stream  $S$ . According to Agosti and Ferro, given a stream  $sm: I = \{1, 2, \dots, n\} \rightarrow \Sigma$ , where  $\Sigma$  is the alphabet of symbols and  $n \in N$ ,  $sm \in SM$ , a segment is a pair:  $st_{sm} = (a, b)$  such that  $1 \leq a \leq b \leq n$ , where  $a, b \in N$ .

In addition to getting the content of the base document that comprises the subdocument, we need to retain the base document context of the subdocument (to allow tools to view or present it *in situ*). We do so by extending the aforementioned definitions of substream and segment to include *presentation specification* and *address*. Also, we store other associated information with a subdocument including properties (such as its creator and a timestamp of its creation) and semantic attributes (such as annotations and tags) as part of promoting the subdocument to be a first-class concept within a digital library.

A presentation specification provides information about how a subdocument was defined in a base document. This notion is borrowed from the hypertext/hypermedia world, where it refers to the runtime behavior of information units presented to the user [274, 278]. In the hypertext/hypermedia literature, presentation specification refers to the encoding information and the mechanism that is used to present a component (or network of components) to the user. A software application/tool uses the presentation specification to display the contents of a digital object. A presentation specification is a descriptive metadata specification conforming to a presentation-based metadata format. A presentation specification is used to specify how the content in a digital object translates into a particular view/presentation. A presentation specification includes information such as the content type of the base document (text, image, audio, video, etc.), the format of the base document (.PDF, .DOC, .JPEG, .AVI, etc.), and the specific software tool used to view/present the base document (Adobe Acrobat, Microsoft Word, Microsoft image viewer, etc.), used when the subdocument was created.

**Definition 6.13** A presentation specification,  $PS = (G_{PS}, \mathcal{R}_{PS} \cup \mathcal{L}_{PS} \cup \mathcal{P}_{PS}, \mathcal{F}_{PS})$  conforms with

a presentation-based metadata format  $MFPS = (V_{MFPS}, \text{def}_{MFPS})$  with the following constraints:

1.  $V_{MFPS} = \{\mathcal{R}_{PS1}, \mathcal{R}_{PS2}, \dots, \mathcal{R}_{PSk}\} \subset 2^{\mathcal{R}_{PS}}$  a family of subsets of the resource labels  $\mathcal{R}_{MFPS}$
2.  $\text{def}_{MFPS} : V_{MFPS} \times \mathcal{P}_{MFPS} \rightarrow V_{MFPS} \cup D_{\mathcal{L}_{MFPS}}$  is a property definition function, where:
  - $\mathcal{P}_{MFPS}$  represent sets of labels for properties
  - $V_{MFPS}$  represents the nodes of a graph structure
  - $D_{\mathcal{L}_{MFPS}}$  is the set of domains that make up the set of literals  $\mathcal{L}_{MFPS}$
3.  $\mathcal{R}_{PS} \subseteq \mathcal{R}_{MFPS}$ ,
4.  $\mathcal{L}_{PS} \subseteq \mathcal{L}_{MFPS}$ ,
5.  $\mathcal{P}_{PS} \subseteq \mathcal{P}_{MFPS}$ , and
6. for every statement  $st = (r, p, l)$  derived from  $PS$ ,  $r \in \mathcal{R}_k$  for some  $\mathcal{R}_k \in V_{MFPS}$  and  $p \in \mathcal{P}_{PS}$  implies  $l \in \text{def}_{MFPS}(\mathcal{R}_k, p)$ .

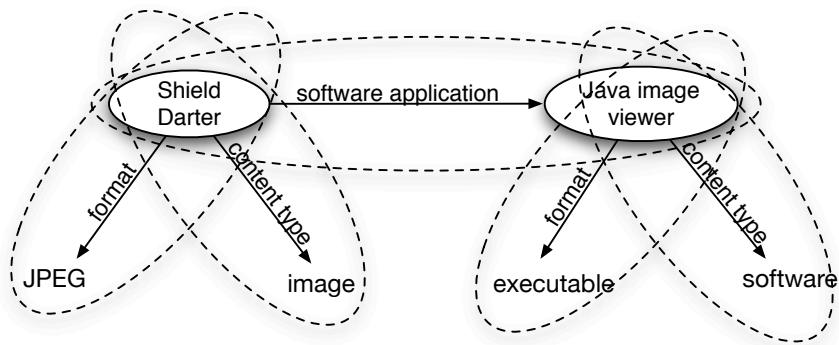


Figure 6.6: Example of a presentation specification.

## 200 6. SUBDOCUMENTS

Examples of resources could be academic papers, images, software applications, etc. Examples of properties include format, content type, software application to view, etc. Consider the example shown in Figure 6.6. Here the object “Shield Darter” is an “image” of “JPEG” format and makes use of the “Java image viewer” software application. Another example is from the Dublin Core metadata format. For any set of labels  $\mathcal{R}$  for resources, the Dublin Core metadata format defines that  $\text{def}_{DC}(\mathcal{R}, \text{'format'}) = \text{String}$  and  $\text{def}_{DC}(\mathcal{R}, \text{'format.mimetype'}) = \text{MIME}$  where  $\text{MIME}$  is a finite set of labels for Resources corresponding to mime types.

A presentation specification of a subdocument consists of all information required to interpret the *address* of the span/region of the subdocument within the base document. The address is used by an appropriate software application to navigate to and view the subdocument in the context of its originating base document. Consider the example of an academic paper, which might have mixed content including text and images. It could be a PDF document presented/viewed using Adobe Acrobat. The address of a segment or substream in this case might be different than if the same content were in a .DOC document presented/viewed using Microsoft Word since the navigation/addressing schemes within each of these tools is different. Adobe Acrobat uses a word-based scheme whereas Microsoft Word uses a character-based scheme. Another example is the address of a subdocument within an image document (or a subimage), which might vary depending on the format, resolution, and software used to view/present the image. Archer et al. extended upon previous SI work to include subdocuments in the DSpace<sup>10</sup> DL software [24]. They implemented a feature for Microsoft Word (and OpenOffice) that allows for creation of subdocuments, which have been stored in an instance of the Fedora DL<sup>11</sup>. Also, it accepts an address for a subdocument with a Microsoft Word (and OpenOffice) base document and displays it highlighted.

**Definition 6.14** Given base document  $BD$ , a **subdocument**  $sd$  is a digital object with the following extensions and constraints:

- $sd$  is a *digital object*  $= (h, SM, ST, StrStreams, PS, addr)$ , where
  1.  $h \in H$ , where  $H$  is a set of universally unique handles (labels);
  2.  $SM_{sd} = \{sm_{sd}[i, j]\} \in SM$ , where  $sm_{sd}[i, j] = \langle a_i, \dots, a_j \rangle$ ,  $0 \leq i \leq j \leq n$ .  $sm_{sd}[i, j]$  refers to substreams of a base document  $BD$ .
  3.  $ST = \{st_1, st_2, \dots, st_m\}$  is a set of structural metadata specifications associated with the base document  $BD$ ;

<sup>10</sup><http://www.dspace.org/>

<sup>11</sup><http://fedora-commons.org/>

#### 6.4. FORMALIZATION AND APPROACH TO AN SI-DL 201

4.  $StrStreams = \{stD_1, stD_2, \dots, stD_m\}$  is a set of StructuredStream functions defined from the base document substreams in the  $SM_{sd}$  set (the second component) of the subdocument and from the structures in the  $ST$  set (the third component).
5.  $PS$  is a *presentation specification*.
6.  $addr$  is the function from the  $SM_{sd}$  set (the second component) of the subdocument and from the presentation specification  $PS$  of the base document.

Note that the subdocument contains the *structures* and the contiguous *streams* of its parent base document that exist within the span defined by the address of the subdocument. It inherits all the descriptive and structural metadata specifications associated with the span defined by the address. Figure 6.7 shows an example of a subdocument with its components, including the substreams and substructures associated with it, as inherited from the containing base document. The left part of the figure shows the base document ( $BD$ ), with a handle ( $h_{BD}$ ), a title (`enhance_cmaps.pdf`), and a highlighted subdocument, which is associated with a presentation specification ( $PS$ ).  $PS$  might contain presentation information, such as content-type, format, and software application. The right part of the figure shows details of components of a subdocument ( $sd$ ), including a handle ( $h_{sd}$ ), a substream ( $sm_{sd}[i,j]$ ), a substructure ( $st_{sd}$ ), and an address ( $addr$ ).  $sm_{sd}[i,j]$  is the sequence of characters in the  $sd$ , where  $i$  and  $j$  indicate the character numbers of the first and the last character of that substream within  $BD$ . In some cases, there might be multiple substreams, such as in a subdocument that includes text and images.  $st_{sd}$  shows the mapping between the structural metadata specification of  $BD$  and the streams within  $sd$ .  $addr$  indicates pointers to the beginning and ending of  $sd$ , considering the presentation specification ( $PS$ ) of  $BD$ .

Since a subdocument is a digital object, it has its own metadata. This could include properties of subdocument creation such as information about the subdocument creator, the timestamp of creation, etc. Also, as with an ordinary digital object, a subdocument could be associated with semantic information such as annotations and tags. Like other digital objects, a subdocument may have many manifestations. For example, consider a subdocument within a text-based PDF document. One manifestation of the subdocument might be the textual excerpt of the subdocument. Another might be an image transformation of a portion of the base PDF document with the highlighted subdocument.

At this point, it is important to describe the relationship between a mark, as defined in the SI literature (Section 6.2.1), and a subdocument. A mark is a reference to a subdocument. Considering the aforementioned definition, a mark necessarily consists of  $h$ ,  $PS$ , and  $addr$ . In the SI literature, text-based marks have been known to store the *excerpt*, which is the content of the marked region (or subdocument). However, an excerpt is not necessary to render a mark or reference the marked region (subdocument) and need not be stored.

## 202 6. SUBDOCUMENTS

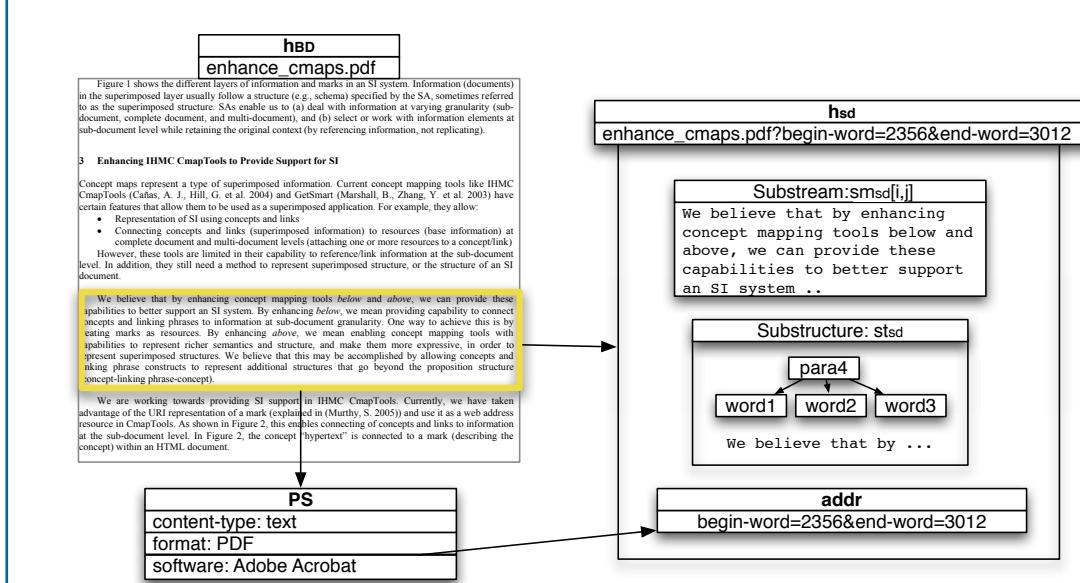


Figure 6.7: Example of a subdocument and its components.

### Superimposed document

A superimposed document can be represented as a complex object (as defined in Section 6.3.1), where at least one of its constituent digital objects is a subdocument.

**Definition 6.15** A superimposed document is a complex digital object, defined as a tuple  $sidoc = (h, SCDO, ST)$ , where

1.  $h \in H$ , where  $H$  is a set of universally unique handles (labels);
2.  $SCDO = \{DO \cup SM\}$ , where  $DO = \{do_1, do_2, \dots, do_n\}$ , and  $do_i$  is a digital object, such that  $\exists$  at least one  $do_i = sd$ , for  $i = 1, 2, \dots, n$ , where  $sd$  is a subdocument and  $SM = \{sm_a, sm_b, \dots, sm_z\}$  is a set of streams;
3.  $S$  is a structure that composes the complex object  $sidoc$  into its parts in  $SCDO$ .

This is consistent with earlier work in SI, where the references to subdocuments (i.e., marks) could be incorporated into a variety of superimposed documents structured according to various data models [445]. A superimposed document can be of different types. For example, it may consist of subdocument references (i.e., marks) interspersed with other digital content, such as in a textual document that has citations to specific portions of other

documents. Another example is a time-ordered arrangement of audio/video clips merged with textual content from web pages [451]. A concept map [455] and a strandmap [154], where the resources point to subdocuments, are other examples.

#### 6.4.2 COLLECTIONS AND CATALOGS

An important component of a digital library with SI support is the ability to deal with various (base document, subdocument, and superimposed document) collections and corresponding metadata catalogs. Here, we define collections and catalogs for the three types of digital objects that we have introduced.

**Definition 6.16** A **base document collection**

$C_{BD} = \{bd_1, bd_2, \dots, bd_l\}$  is a set of base documents.

**Definition 6.17** A **subdocument collection**

$C_{sd} = \{sd_1, sd_2, \dots, sd_m\}$  is a set of subdocuments.

**Definition 6.18** A **superimposed document collection**

$C_{sidoc} = \{sidoc_1, sidoc_2, \dots, sidoc_n\}$  is a set of superimposed documents.

**Definition 6.19** Let  $C_{BD}$  be a collection of  $l$  base documents, with  $l$  handles in  $H$ , such that there is a unique handle for each base document in  $C_{BD}$ . A **base document metadata catalog**  $DM_{C_{BD}}$  for  $C_{BD}$  is a set of pairs

$\{(h, \{dm_{BD_1}, \dots, dm_{BD_{l_h}}\})\}$ , where  $h \in H$  and the  $dm_{BD_i}$  are descriptive metadata specifications for  $BD$ , the base document.

**Definition 6.20** Let  $C_{sd}$  be a collection of  $m$  subdocuments, with  $m$  handles in  $H$ , such that there is a unique handle for each subdocument in  $C_{sd}$ . A **subdocument metadata catalog**  $DM_{C_{sd}}$  for  $C_{sd}$  is a set of pairs

$\{(h, \{dm_{sd_1}, \dots, dm_{sd_{m_{h_{sd}}}}\})\}$ , where  $h_{sd} \in H_{sd}$  and the  $dm_{sd_i}$  are descriptive metadata specifications for the subdocument,  $sd$ .

**Definition 6.21** Let  $C_{sidoc}$  be a collection of  $n$  superimposed documents, with  $n$  handles in  $H$ , such that there is a unique handle for each superimposed document in  $C_{sidoc}$ . A **superimposed document metadata catalog**  $DM_{C_{sidoc}}$  for  $C_{sidoc}$  is a set of pairs  $\{(h, \{dm_{sidoc_1}, \dots, dm_{sidoc_{n_h}}\})\}$ , where  $h \in H$  and the  $dm_{sidoc_i}$  are descriptive metadata specifications for the superimposed document,  $sidoc$ .

## 204 6. SUBDOCUMENTS

### 6.4.3 SERVICES

In an SI-DL, traditional services such as browsing, indexing, and searching now act upon different types of digital objects including base documents, subdocuments, superimposed documents, as well as metadata associated with each of these. For example, using the search service on subdocuments, the query specification can contain subdocument-related information and the results can include subdocuments. In addition, advanced searches on components of superimposed documents and base documents might be possible. For example, one could retrieve all subdocuments within a particular base document. Another example is to retrieve all base documents that contain subdocuments, which are referenced in a particular superimposed document.

In addition to traditional digital library services, we add a new service, *view in context*, to the digital library to support access for viewing/presentation of subdocuments in the context of their parent base document. This can be considered an extension of the browsing services as defined in the 5S framework, which acts upon the extended hypertext that now includes subdocuments and links between base documents and subdocuments as well as those between superimposed documents and subdocuments. This creates new referential hyperlinks between a subdocument and its parent base document as well as those between a superimposed document and its constituent subdocuments. In addition, we now need to make use of services, for example plugins that can be invoked by the digital library based on the presentation specification of the base document which contains a subdocument.

**Definition 6.22** A *view in context* service is a set of scenarios  $\{sc_1, \dots, sc_n\}$  over an extended hypertext where events are defined by edges of the hypertext graph  $(V_{HE}, E_{HE})$ , where  $V_{HE}$  includes the union of base documents, subdocuments, and superimposed documents and  $E_{HE}$  includes the links between a subdocument and base document, such that the subdocument–base document link events  $e_i$  are associated with a function  $ViewInContext : V_{HE} \times E_{HE} \rightarrow Contents$ , which, given a subdocument, instantiates the service that is required to present/view the base document (facilitated through information in the presentation specification,  $PS$ ), retrieves the content of the base document and uses the aforementioned service for the base document’s presentation with the subdocument highlighted within the base document, i.e.,  
 $ViewInContext(v_{k_{sd}}, e_{k_i}) = P(v_{t_{sd}})$  for  $e_{k_i} = (v_{k_{sd}}, v_{t_{sd}}) \in E_{HE}$ . Here,  $v_{k_{sd}}$  is a reference to the subdocument in a superimposed document and  $v_{t_{sd}}$  is a reference to the subdocument in its original context, or in its parent base document.

An example of the view in context service is shown in Figure 6.8. Here, the subdocument, which is used in a superimposed document (e.g., a concept map), is created in a base document (i.e., a Microsoft Word document), with a plugin that allows subdocument creation and viewing. On instantiating the view in context service from this subdocument, an instance of Microsoft Word (an application to work with the base document) is launched,

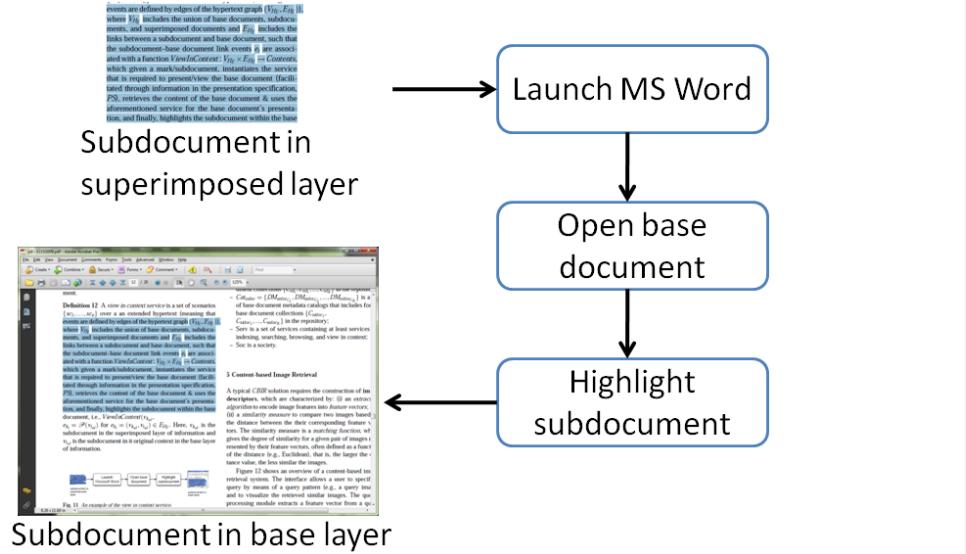


Figure 6.8: An example of the view in context service.

the base document containing the subdocument is opened and presented in the Microsoft Word application, and the subdocument is highlighted in this base document.

When subdocuments are part of a DL, we might consider extracting collections of subdocuments, generated from search results, browse criteria, etc. In such cases, it might be useful to have a view in context service, where a subdocument might be *viewed in context of a superimposed document*, of which it is a part. For example, consider the subdocument presented in Figure 6.8. When this subdocument is presented as part of a search result set, a user might want to view the superimposed documents, where this subdocument is used. Upon activating the view in context of superimposed document, the DL might display a concept map, highlighting a resource (pointing to the subdocument) attached to a concept.

#### 6.4.4 SI-DL

A digital library with superimposed information, or an SI-DL, is a digital library that manages a repository of collections of base documents, subdocuments, and superimposed documents; and is associated with services, including indexing, searching, browsing, and view in context.

**Definition 6.23** A **superimposed information supported digital library** is a 4-tuple  $(\mathcal{R}, DM, Serv, Soc)$ , where

- $\mathcal{R}$  is a repository;

## 206 6. SUBDOCUMENTS

- $DM = DM_{BD} \cup DM_{sd} \cup DM_{sidoc} \cup DM_{do}$ ,
- $DM_{BD} = \{DM_{BD_{C_1}}, DM_{BD_{C_2}}, \dots, DM_{BD_{C_L}}\}$  is a set of base document metadata catalogs for all base document collections  $\{C_{BD_1}, C_{BD_2}, \dots, C_{BD_L}\}$  in the repository;
- $DM_{sd} = \{DM_{sd_{C_1}}, DM_{sd_{C_2}}, \dots, DM_{sd_{C_M}}\}$  is a set of subdocument metadata catalogs for all subdocument collections  $\{C_{sd_1}, C_{sd_2}, \dots, C_{sd_M}\}$  in the repository;
- $DM_{sidoc} = \{DM_{sidoc_{C_1}}, DM_{sidoc_{C_2}}, \dots, DM_{sidoc_{C_N}}\}$  is a set of base document metadata catalogs for all base document collections  $\{C_{sidoc_1}, C_{sidoc_2}, \dots, C_{sidoc_N}\}$  in the repository;
- $DM_{do}$  is a set of metadata catalogs for all collections  $\{C_{do_1}, C_{do_2}, \dots, C_{do_K}\}$  in the repository, that are not in the sets of base document, subdocument, and superimposed document collections;
- Serv is a set of services containing at least services for indexing, searching, browsing, and view in context;
- Soc is a society.

## 6.5 CASE STUDIES

### 6.5.1 USING THE SI-DL METAMODEL TO DESCRIBE SUPERIDR

In this case study, we use the metamodel for an SI-DL to define and analyze content and behavior of SuperIDR, an image description and retrieval application [453, 452, 450].

#### SuperIDR

SuperIDR might be considered a prototype SI-DL, which enables users to work with subimages and associated information. It was designed and developed with the aim of supporting scholarly tasks that involve working with subimages. In fisheries sciences people work with subimages to identify species of fishes [452, 450]. The SuperIDR version described in this section was customized to include images, descriptions, and taxonomical information for species of freshwater fishes of Virginia [313]. The use of subimages might apply to other domains as well, which have tasks that involve working with images with a significant number of details, such as analyzing paintings in art history, reviewing plans in architecture, or studying X-rays of the human body in medicine. In another customization of SuperIDR, it was seeded with parasite images and information to evaluate its applicability in the Zooparasitology domain [354].

SuperIDR brings together SI and *content-based image retrieval* (CBIR) in a personal image-based digital library environment. It incorporates the idea of SI and working with

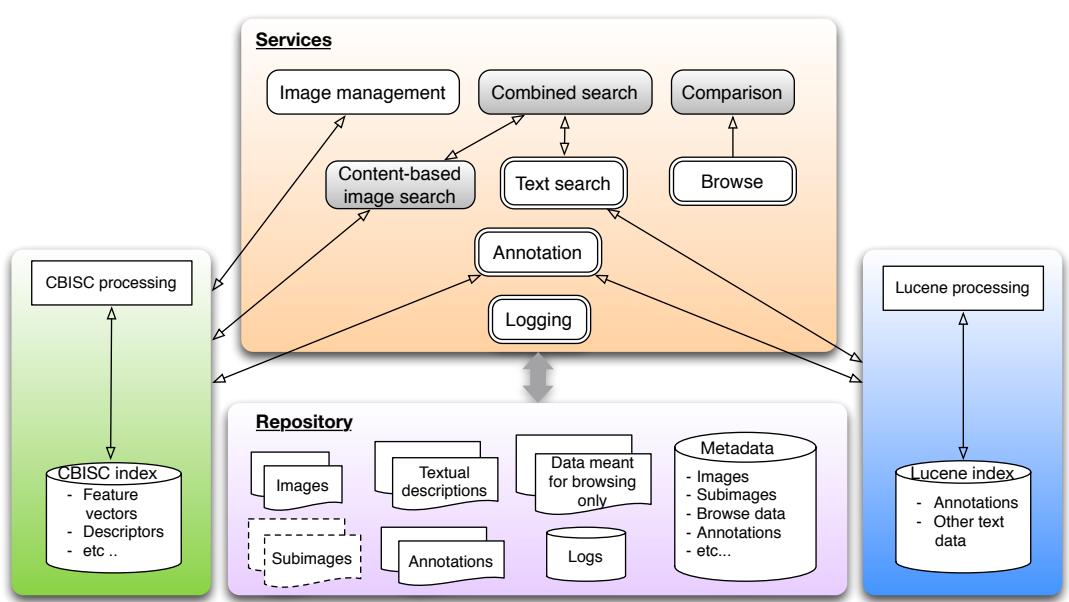


Figure 6.9: Software architecture of SuperIDR shows repository with collections, and services. The CBIRC and Lucene components are used to index and retrieve image and text data, respectively.

parts of images *in situ*, to enable users to select, annotate, explore, retrieve, and compare parts of images in the context of the original image and other associated data. SuperIDR uses CBIR to index and retrieve the visual content of images and subimages. In addition to providing traditional digital library services, such as browsing, searching, and indexing, the digital library environment is also responsible for managing images, subimages, and all other associated data (such as descriptions, metadata about images, subimages, etc.).

Figure 6.9 outlines the software architecture of SuperIDR. Data collections consist of images, subimages (derived from images), textual descriptions, annotations, taxonomical classification data that might be used for browsing only (such as family- or genera-level images and information), log data, and metadata related to all of the aforementioned data. There are three kinds of services in SuperIDR:

- Traditional DL services that have not changed with the addition of subimages - image management.
- Traditional DL services that now work with subimages and related data - browse, text search, annotation, and logging.

## 208 6. SUBDOCUMENTS

- New DL services that work with subimages and related data - browsing subimages and associated annotations, content-based image search, combined search, and comparison.

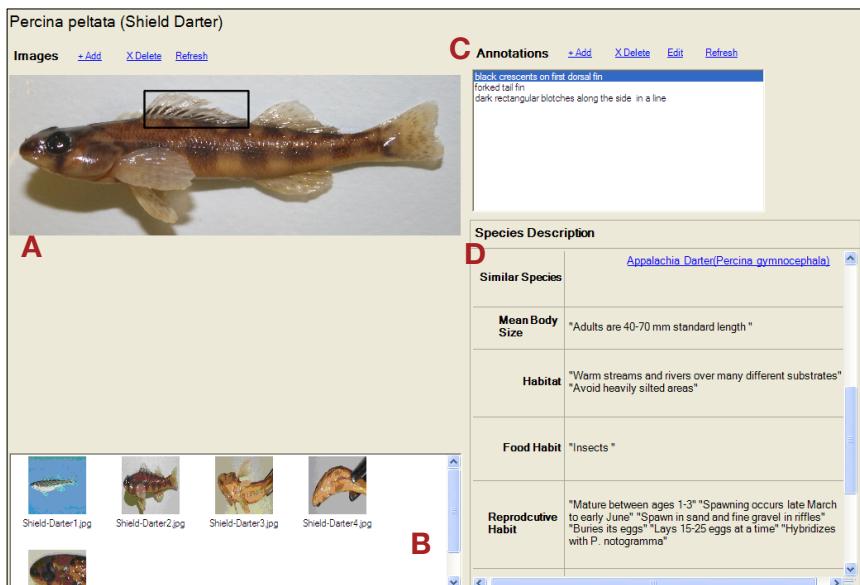


Figure 6.10: Species description interface in SuperIDR: A) focus image, showing marked region associated with a selected annotation; B) list of images in the species; C) annotation menu and list of annotations on the focus image; and D) physical description, habitat, and other information about the species.

SuperIDR uses the Content-Based Image Search Component, CBISC, for indexing and retrieving the visual content of images and subimages. CBISC is an Open Archives Initiative (OAI)-compliant component that provides an easy-to-install search engine to query images by content [636]. Lucene<sup>12</sup>, a open-source text retrieval package, is responsible for full-text and field-wise indexing and search of all text data in the SuperIDR collections, including textual description (of images) and annotations.

We now briefly describe the services of SuperIDR. For further details on these services and on SuperIDR, the reader is referred to Chapter 5 of the Ph.D. dissertation [450].

**Annotation:** The annotation service enables a user to select subimages within images and associate them with text annotations. Also, a user can edit and delete annotations.

**Image management:** A user can add images to and delete (user-added) images from a species using the image management service.

<sup>12</sup><http://lucene.apache.org/>

## 6.5. CASE STUDIES 209

**Text search:** SuperIDR uses the full-text and/or field-wise search in the Lucene component to match keywords entered by the user against one of the following: 1) annotations; 2) species descriptions; or 3) both – annotations and species descriptions. Depending on the user’s selection, a separate ranked list of results is processed and displayed for species descriptions (Figure 6.11-B) and for annotations (Figure 6.11-A).

**Content-based image search:** The content-based image search service takes as input a query, which is either an image or a subimage. It then sends this query to CBISC, which in turn matches the visual content of this query image or subimage against the visual content of all images and subimages in the CBISC index. A ranked list of results is produced, which could contain images and/or subimages. Search results are displayed separately for images and subimages.

**Combined search:** The purpose of including a combined search service is to give the user an idea of how image and text content might be combined in search, especially focusing on how subimages might be combined with text content. The combined search service takes, as input, a query, which is a combination of an image or subimage and keywords. In addition, the user can specify an image-weight and a text-weight, which indicate the relative importance given by the user to each component (image and text) of the query. The text and image search results are combined<sup>13</sup> to produce a single list for each of species/images and of annotations/subimages. The combined search results are displayed similar to text search results, as shown in Figure 6.11-A,B.

**Browse:** The browse service enables multiple ways of browsing through species images, descriptions, subimages, and annotations, including:

- Browsing through images and description of a species
- Browsing through annotations, and for each image, viewing the associated subimage in the context of its base image.
- Browsing through the taxonomical classification, including families, genera, and species of fishes using: 1) column-wise organization or 2) tree-based organization.
- Browsing through a digital version of the identification key guide of freshwater fishes of Virginia [313].

**Comparison:** The comparison service is a special case of the browsing service. It enables a user to view two images side-by-side. A user is able to choose from images of the same or different species. While viewing the images side-by-side, a user can browse through annotations and view the associated subimages in the context of their base images. The goal of the comparison service is to enable the user to manually analyze two images side by side.

**Logging:** The logging service is used to log all user interactions with the tool.

<sup>13</sup>The combination method is explained in [450].

## 210 6. SUBDOCUMENTS

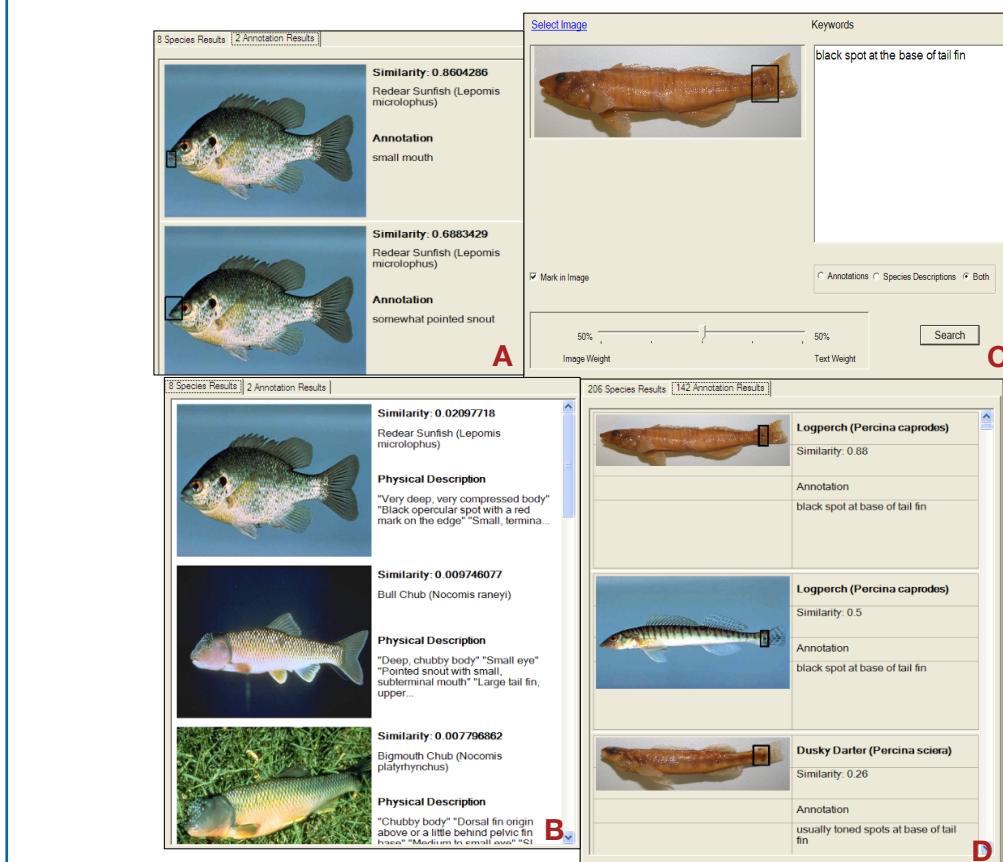


Figure 6.11: Search in SuperIDR: A) annotation search results for the text query – “red mark” “small mouth” “pointed snout” “no spots”; B) species description results for the same query; C) combined search query interface; and D) annotation/subimage combined search results.

### Analyzing and describing SuperIDR

SuperIDR might be considered to be an extension of the minimal digital library as defined in the 5S framework (Chapter 1). Figure 6.12 shows the components of SuperIDR. We extended the definition of a digital object to include an image digital object, a subimage (or image subdocument), an image complex object, and a superimposed image complex object. In addition, SuperIDR has other digital objects, such as annotation and image complex object description. These conform to the digital object definition as mentioned in the 5S framework (Chapter 1). Each of the aforementioned digital object belongs to respective collections and is associated with a metadata catalog. In addition, SuperIDR

has the view in context and CBIR services. The rest of this section describes and analyzes the components of SuperIDR<sup>14</sup>.

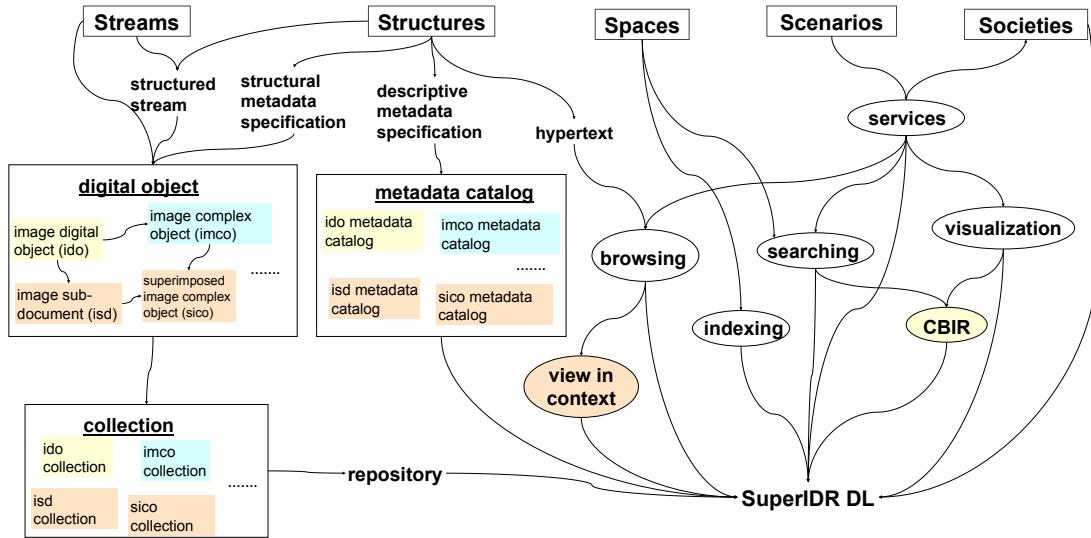


Figure 6.12: Definitional dependencies among concepts in an SuperIDR digital library, showing connections among concepts in the 5S framework and the extensions defined.

Figure 6.13 shows the information components within SuperIDR and relationships among them. Here, an image complex object consists of (at least) a description and a collection of image digital objects (or, images). When at least one of the image digital objects is marked up and annotated, an image subdocument (or, a subimage) is created and added to the image complex object. Also, the associated annotation object is added to the image complex object. The addition of an image subdocument makes the image complex object into a superimposed image complex object. Each of the aforementioned digital objects, image digital object, image subdocument, annotation, image complex object description, and image complex object, has an associated metadata record. Each type of digital object is also part of a collection of the same type. For the remainder of this section, we use the notation mentioned in Table ?? to refer to each of these digital objects. In the case of SuperIDR customized for fish data (as discussed in most of this dissertation research), the aforementioned digital objects and associated metadata correspond to fish-related data, as mentioned in Table ??.

An image complex object in this case is a species. When SuperIDR is customized to other, similar, image-based domains, such as species

<sup>14</sup>Images and CBIR services are important components of SuperIDR. We will assume the definitions of these concepts, as presented in Chapter 9.

## 212 6. SUBDOCUMENTS

**Table 6.2:** Digital objects in SuperIDR, notations used in the case study, and examples from SuperIDR customized for fish-related data

Digital object	Notation	Example from SuperIDR customized for fish-related data
Image complex object	<i>imco</i>	A fish species, consisting of a description and a set of images.
Image complex object description	<i>desc</i>	Description of a fish species, including details, such as physical description, mean size, and habitat.
Image digital object	<i>ido</i>	Fish image, such as an image of a trout.
Image subdocument	<i>isd</i>	Part of a fish image, such as a fin, mouth, or tail.
Annotation	<i>ann</i>	Textual description of a part, such as large mouth or orange dorsal fin.
Superimposed image complex object	<i>sico</i>	A fish species, consisting of a description, a set of images, with one or more marked-up images, and annotations associated with marked-up regions in images.
Base document	<i>bd</i>	An image digital object before it has been marked up or a species before it contains a marked-up image and associated annotations.

of trees with accompanying images and descriptions and genres of paintings that include images and descriptions, an image complex object might take other forms.

Note that an image digital object and an image complex object are candidate base documents. When an image subdocument is created on an image digital object, the image digital object becomes a base document. Similarly, at first, an image complex object consists of a description and a set of images. When an image subdocument is created (and hence, an associated annotation) in at least one of the image digital objects of an image complex object, this image complex object becomes a superimposed image complex object. Thus, a single digital object (in this case, an image or an image complex object) might play multiple roles and as a result, might be part of multiple digital object collections.

We can define a SuperIDR digital library as a 4-tuple,  
 $SuperIDR\_DL =$   
 $(SuperIDR\_{\mathcal{R}}, SuperIDR\_{DM}, SuperIDR\_{Serv}, SuperIDR\_{Soc})$ , where

- $SuperIDR\_{\mathcal{R}}$  is a repository, having collections  $C_{ido}$ ,  $C_{isd}$ ,  $C_{ann}$ ,  $C_{desc}$ ,  $C_{imco}$ ,  $C_{sico}$ , and  $C_{bd}$ , where

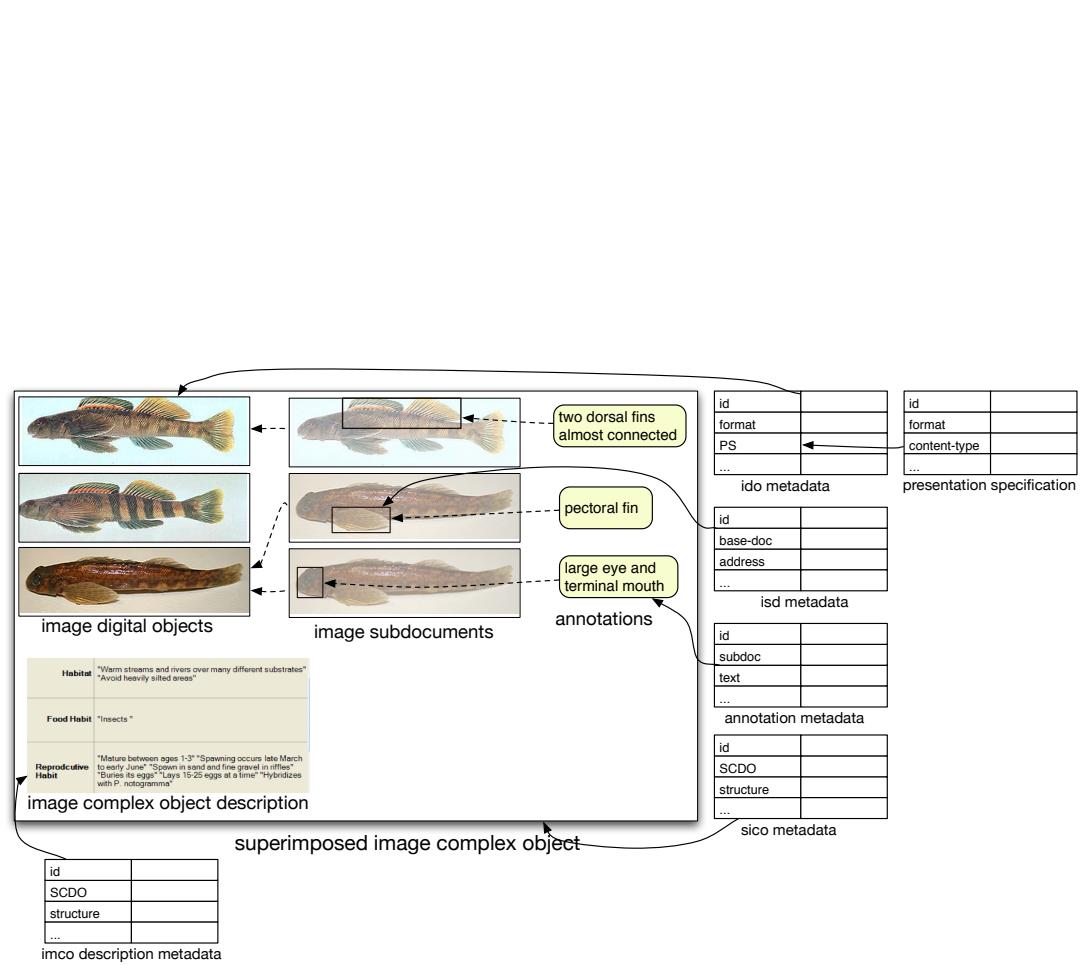


Figure 6.13: A superimposed image complex object, its components, associated metadata, and relationships among all of the above.

## 214 6. SUBDOCUMENTS

- $C_{ido}$  is a collection of image digital objects,
  - $C_{isd}$  is a collection of image subdocuments,
  - $C_{ann}$  is a collection of annotations
  - $C_{desc}$  is a collection of image complex object descriptions,
  - $C_{imco}$  is a collection of image complex objects,
  - $C_{sico}$  is a collection of superimposed image complex objects,
  - $C_{bd}$  is a collection of base documents,
- $SuperIDR\_DM = \{DM_{ido}, DM_{isd}, DM_{ann}, DM_{desc}, DM_{imco}, DM_{sico}, DM_{bd}\}$  is a set of descriptive metadata specifications, where
    - $DM_{ido}$  is a metadata catalog for the collection of image digital objects,
    - $DM_{isd}$  is a metadata catalog for the collection of image subdocuments,
    - $DM_{ann}$  is a metadata catalog for the collection of annotations,
    - $DM_{desc}$  is a metadata catalog for the collection of image complex object descriptions,
    - $DM_{imco}$  is a metadata catalog for the collection of image complex objects,
    - $DM_{sico}$  is metadata catalog for the a collection of superimposed image complex objects,
    - $DM_{bd}$  is a metadata catalog for the collection of base documents,
  - $SuperIDR\_Serv$  is a set of services containing services for indexing, searching, browsing, CBIR and view in context;
  - $SuperIDR\_Soc$  of  $SuperIDR\_DL$  is a society including {Patron, Student, Faculty, Researchers, Practitioner, Amateur, SuperIDR\_Admin, ... }.

We now describe the contents of some of these components further. The set of streams in  $SuperIDR\_DL$  consists of image and text streams. The union set of handles of various digital objects in collections  $C_{ido}$ ,  $C_{isd}$ ,  $C_{ann}$ ,  $C_{desc}$ ,  $C_{imco}$ ,  $C_{sico}$ , and  $C_{bd}$  will compose  $SuperIDR\_DLIDs$ , the set of handles in  $SuperIDR\_DL$ . Examples of content of each metadata specification are described here.

1.  $DM_{ido} = \{\text{'id'}, \text{'image name'}, \text{'format'}, \text{'size'}, \text{'location'}, \dots\};$
2.  $DM_{desc} = \{\text{'id'}, \text{'author'}, \text{'source'}, \dots\};$
3.  $DM_{imco} = \{\text{'id'}, \text{'author'}, \text{'structure'}, \dots\};$

4.  $DM_{bd} = \{\text{'id'}, \text{'name'}, \text{'format'}, \text{'size'}, \dots\}$ .
5.  $DM_{isd} = \{\text{'id'}, \text{'base document'}, \text{'address'}, \text{'presentation\_specification'}, \dots\}$ ;
6.  $DM_{ann} = \{\text{'id'}, \text{'subdocument'}, \text{'text'}, \dots\}$ ;
7.  $DM_{sico} = \{\text{'id'}, \text{'author'}, \text{'structure'}, \dots\}$ .

Items 4, 5, and 6 are added to  $SuperIDR\_DL$ , when at least one of the images within the image complex object is marked and annotated. Then, the image complex object is modified into a superimposed image complex object as it now contains subdocuments.

Using  $SuperIDR\_DL$ , we will formally describe two scenarios, each of which involves one or more services of the extensions mentioned in this paper.

### 1. *AddImageSubdocumentAndAnnotation*

**Informal description:** This scenario is part of creating and adding an annotation into DLSuperIDR. We focus on what happens in a DLSuperIDR before, during, and after a subdocument is created. Given an image, which is associated with an image complex object, an address referencing a part of the image, and an associated text annotation, a subdocument and an annotation object are created. In addition, the newly created subdocument and annotation are added to the species complex object. If this is the first subdocument added to a species, it changes from being an image complex object to a superimposed image complex object.

**Goal:** Given an image, which is part of an image complex object, an address of a part of that image, and an associated text annotation, create a subdocument and annotation object and add those to the aforementioned image complex object. This adds a new subdocument to the  $DLSuperIDR$  and makes the image complex object a superimposed image complex object.

**Scenario:**

$\langle e_1 : p = AddImageSubdocumentAndAnnotation$

$(ido_j, imco_i, ps_k, addr_l, ann_m), e_2 : p = response (sico_i, isd_o) \rangle$ , where the following constraints apply:

- (a)  $ido_j$  is an image digital object, such that  $ido_j \in imco_i$  and  $ido_j \in C_{ido}$  and  $imco_i \in C_{imco}$ , where  $imco_i$  is an image complex object that consists of images and species descriptions,  $C_{ido}$  is a collection of image digital objects in  $SuperIDR\_DL$ , and  $C_{imco}$  is a collection of image complex objects in  $SuperIDR\_DL$ .
- (b)  $addr_l$  is an address, specifying a region/span within the image digital object  $ido_j$ , and is associated with a presentation specification  $ps_k$ .

## 216 6. SUBDOCUMENTS

- (c)  $ann_m$  is an annotation digital object, such that  $ann_m \in sico_i'$  and  $ann_m \in C_{ann}$ , where  $C_{ann}$  is a collection of annotations in  $SuperIDR\_DL$ .
- (d)  $isd_o$  is a newly created subdocument, such that  $isd_o \in sico_i'$  and  $isd_o \in C_{isd}$ , where  $C_{isd}$  is a collection of image subdocuments in  $SuperIDR\_DL$ .
- (e)  $imco_i$  is modified into  $sico_i'$ , a superimposed image complex object, such that  $ido_j$  and other digital objects in  $imco_i$  are now in  $sico_i'$ , and  $sico_i' \in C_{sico}$ , where  $C_{sico}$  is a collection of superimposed image complex objects in  $SuperIDR\_DL$ .
- (f)  $C_{imco}' = C_{imco} - imco_i$ , where  $C_{imco}'$  is the modified collection of image complex objects in  $SuperIDR\_DL$ , which does not contain the image complex object,  $imco_i$ .

### 2. *DisplayImageSubdocumentList*

**Informal description:** This scenario might take place in case of browsing search results which include image subdocuments and associated information as result items (see Figure 6.11) or in case of browsing through annotations associated with image subdocuments within an image complex object (see Figure 6.10). Given a list of image subdocuments and associated information, clicking on an image subdocument in that list will cause the system to display the image subdocument in its original context or the context of its containing base document. In a sense, a hyperlink is being traversed from the subdocument in the list (superimposed layer) to the subdocument in its original context (base layer).

**Goal:** Given a list of image subdocuments, display them in the context of the original base document.

**Scenario:**  $\langle e_1 : p = DisplayImageSubdocumentList(isd_1, isd_2, \dots, isd_n), e_2 : p = response(\mathcal{P}(v_{t_{isd_1}}), \mathcal{P}(v_{t_{isd_2}}), \dots, \mathcal{P}(v_{t_{isd_n}}))$ , such that  $\mathcal{P}(v_{t_{isd_i}})$ ,  $1 \leq i \leq n$  is the response to the service  $ViewInContext(v_{k_{isd_i}}, e_{k_i})$ , with the following constraints:

- (a)  $isd_i$ ,  $1 \leq i \leq n$  are image subdocuments
- (b)  $e_{k_i} = (v_{k_{isd_i}}, v_{t_{isd_i}}) \in E_{HE}$ , where  $E_{HE}$  is the extended hypertext formed by the network of image base documents, image subdocuments, and superimposed image complex objects.
- (c)  $v_{k_{isd_i}}$  is a reference of the image subdocument in the superimposed image complex object
- (d)  $v_{t_{isd_i}}$  is the image subdocument in its original context of its containing parent image digital object

### 6.5.2 FLICKR (PLANNED)

### 6.5.3 USING THE METAMODEL TO MAP SUBDOCUMENTS/ANNOTATIONS BETWEEN TWO SYSTEMS (PLANNED)

## 6.6 SUMMARY

We developed the metamodel presented in this chapter to abstract and model the data and services in an SI-DL and to formally define the components of an SI-DL. The SI-DL metamodel builds upon the 5S framework for minimal digital libraries to provide support for working with subdocuments. The main additions to a minimal DL, are the notions of a subdocument, a base document, a superimposed document, and a view-in-context service. In essence, by treating a subdocument as a first-class object in a DL, we are now able to organize, index, search, browse, view, and use subdocuments. We verified the descriptive power of an SI-DL through a case study, where we used the SI-DL metamodel to describe SuperIDR, an image description and retrieval application.

## 6.7 EXERCISES AND PROJECTS

1. What is an example of the use of subdocuments that is distinct from annotation?
2. Describe the components of a subdocument on a video in YouTube considering the definition of a subdocument presented in this chapter. Note that there are at least three streams involved in this subdocument - a space-based image stream and a time-based audio stream and a time-based video frame. List/describe associated metadata and other digital objects associated with this subdocument.
3. Map subdocuments and associated information from System-A to System-B.

## CHAPTER 7

# Ontologies

by Seungwon Yang and Mohamed Magdy

*Abstract:* While many digital libraries (DLs) support taxonomic and /or category systems, few support ontologies in the most general sense. To incorporate the benefits of using ontologies in DLs, it is necessary to understand clearly what ontologies are, how they are developed, current practices of ontology use in DLs, and potential applications in various aspects of DLs. We start from the basics of ontologies such as how humans perceive environment and develop concepts, various definitions, and their components. Then, the three types of ontologies, namely upper ontologies, linguistic ontologies, and domain ontologies, are presented with several examples. We then present a spectrum of ontology examples based on their formality and expressiveness. The ontology engineering section includes methodologies for development and tools to support the development process. In the ontology applications section, three application areas such as DLs, semantic Web, and focused crawling are described followed by ontology evaluation methodologies. A case study of developing a Crisis, Tragedy, and Recovery (CTR) ontology is in the last section.

## 7.1 INTRODUCTION

In this chapter, we provide an overview of what an ontology is and its related topics such as ontology components, kinds of ontologies, development tools and approaches, applications, and methodologies to evaluate ontologies. We present upper ontologies, which attempt to capture general world knowledge as well as domain ontologies, which represent knowledge in a specific domain. We use the term ontology in its broadest sense. Therefore, we include taxonomies, thesauri, classification systems, and lexical databases in the boundaries of ontology.

This section presents various definitions, types, and languages of ontologies. Related studies are introduced in Section 2. Section 3 presents formal definitions for the components of ontologies. Ontology engineering processes, development methodologies, and tools are explained in Section 4. Section 5 is about several application areas such as digital libraries, classification, semantic web, and focused crawling. Section 6 presents methodologies to evaluate the quality and validity of ontologies. A case study of initial ontology development in the field of disaster management is introduced in the final section of 7.

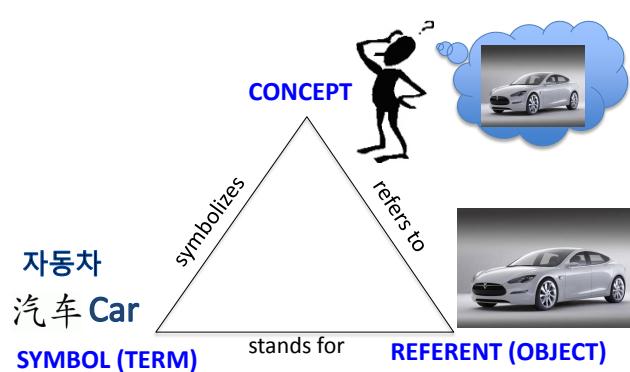


Figure 7.1: The meaning triangle

### 7.1.1 WHAT IS AN ONTOLOGY

We perceive objects in the surrounding environments and map them into certain concepts in our mind. These intrinsic concepts are expressed as symbols (i.e., terms). Ogden et al. [486] explains these relationships, among objects, concepts, and corresponding terms, as the *meaning triangle* (see Figure 7.1). For example, we see cars on the street. They have various designs, colors, materials, and so on. They also have different performance ratings such as top speed and horsepower. The object that we call a *car* is mapped as a concept in our minds. When we think of a car, we have general knowledge about what it looks like, which parts it consist of, how to drive it, etc. The symbol (i.e., term) *car* represents the concept residing in our minds, and it is useful when we share this concept with others. A concept does not have to have its corresponding object. An example of this is the abstract concept *emotions*. Such concepts and their relationships together form knowledge in our mind. Some knowledge can be specific to a domain or specialty, such as computer science, hence the development of specialized ontologies for those fields (see Fig. 7.2).

#### Ontology definitions

Various researchers in the Ontology Engineering (OE) and Artificial Intelligence (AI) field have provided definitions for *ontology*. Among these definitions, the most quoted one is from Gruber et al. [267] : An ontology is an explicit specification of a conceptualization. It was slightly modified by Borst et al.[77] as: Ontologies are defined as a formal specification of a shared conceptualization.

Studer et al. [612] provide a more complete and detailed definition, which covers both definitions above: An ontology is a formal, explicit specification of a shared conceptualization. *Conceptualization* refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon. Explicit means that the type

## 220 7. ONTOLOGIES

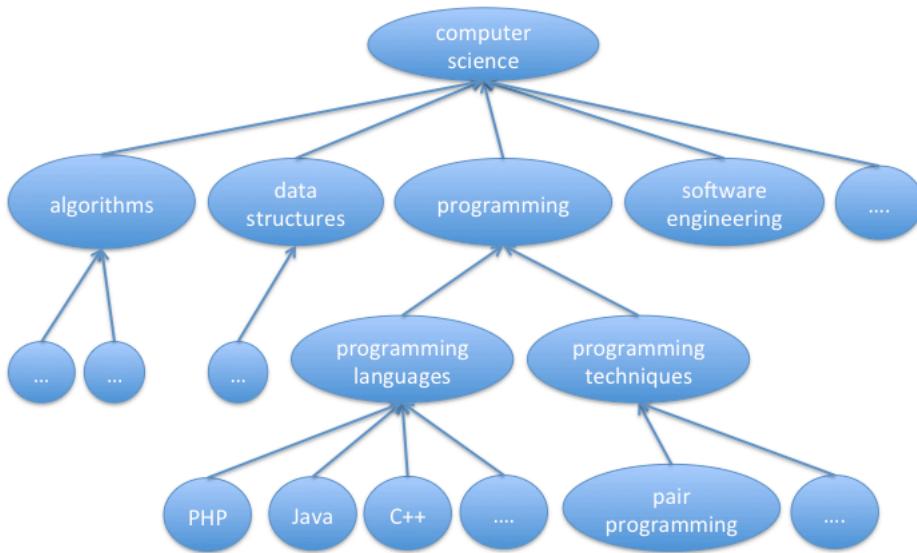


Figure 7.2: A portion of a Computer Science ontology

Table 7.1: Common ontology components and examples

Name	Example
Classes (i.e., concepts)	Programming language
Instances (i.e., objects)	Python
Properties (i.e., attributes)	Latest version
Values	3.2.2
Relationships (i.e., relations)	Python <i>is-a</i> programming language

of concepts used, and the constraints on their use, are explicitly defined. Formal refers to the fact that the ontology should be machine-readable. Shared reflects the notion that an ontology captures consensual knowledge, that is, it is not private to some individual, but accepted by a group.

### Ontology components

Ontologies consist of common components. Classes or concepts are kinds of things or types of objects. Instances or objects are specific things. They can be either concrete (e.g., people, automobiles, animals) or abstract (e.g., numbers). For example, *Python* is an instance of a class *programming language*. Properties or attributes add characteristics and values to

classes or instances. For example, ‘Latest version’ attribute of the ‘Python’ instance has a value ‘3.2.2’. Relationships specify in what sense an object or a class is related to another. Two major relationships are *is-a* (subsumption) and *is-part-of* (meronymy). The example sentence in the table 7.1 shows a subsumption relationship between ‘Python’ and ‘programming language’. In ‘a keyboard *is-part-of* a laptop’, an instance ‘keyboard’ is a component of another instance ‘laptop’.

The combinations of relationships provide semantics to the ontology. Other ontology components include axioms (i.e., assertions), which represent knowledge that is considered true. Based on axioms and rules, theorem can be deduced. Multiple axioms and (deduced) theorems comprise a theory. Other components of ontologies include events, restrictions, and function terms.

### 7.1.2 KINDS OF ONTOLOGIES

One way to classify ontologies is whether the ontology concepts are general or specific for a certain field. Upper ontologies have high-level and general concepts that are applicable across a wide range of domains, or worlds. Domain ontologies attempt to represent knowledge using specific terms in a certain knowledge domain. Specific classes and instances from domain ontologies can be mapped to general classes in an upper ontology. Another kind of ontology is a linguistic ontology. However, some consider this type of ontology not as an ontology but as a lexical database.

#### Upper ontologies

A formal upper ontology is also called a foundation ontology. The terms in the upper ontologies are not specific to a certain domain. Therefore, upper ontologies can cover a wide range of concepts in multiple domains. They might have limitations in representing specific knowledge details because the same terms might be translated differently depending on the knowledge domains under an upper ontology. Selected examples of standardized upper ontologies are:

**Cyc:** Cycorp ([www.cyc.com](http://www.cyc.com)) built the Cyc ontology with a vision to create the world’s first true artificial intelligence that has commonsense knowledge and ability to reason about it. Cyc includes about 3,000 terms that are organized into 43 topical areas such as fundamentals, times and dates, living things, actors and actions, etc. Under these terms, there exist over 1 million assertions that were manually implemented in the CycL language.

**The Standard Upper Ontology (SUO):** IEEE initiated a joint effort to develop a large, general, and formal ontology [508]. The participants were from academia, industry, and government in several countries. From this effort, two ontologies emerged. They are Information Flow Framework (IFF) Foundation Ontology and Suggested Upper Merged

## 222 7. ONTOLOGIES

Ontology (SUMO). Not only these two ontologies but also OpenCyc, 4D ontology, and Multi-Source Ontology (MSO) are competing to be a foundation of the standard.

**Suggested Upper Merged Ontology (SUMO):** It is a formal ontology stated using an ontology language called SUO-KIF (cite: Pease 2003 book page 86). It includes an upper-level ontology that has about 1000 terms, a Mid-Level Ontology (MLO) with about 2000 terms, and a dozen domain ontologies. In total, SUMO includes over 20,000 terms and 70,000 axioms. It is mapped to a lexical database, WordNet, and expanded using Wikipedia.

Instead of modeling a specific domain, linguistic ontologies describe semantic constructs by using words as grammatical units. They are built with different purposes. For example, WordNet is used as an online lexical database, and SENSUS as an ontology for machine translation.

**WordNet:** It is a large English lexical database created at Princeton University based on Psycholinguistic theories [429] [428]. It organizes words into synonym sets called synsets, which represent underlying concepts. It also provides general and brief definitions for the synsets as well as a representation of the semantic relations among them. The types of relations include synonymy, antonymy, hypernymy (subclass-of), hyponymy (superclass-of), meronymy (part-of), and holonymy (has-a). WordNet is widely used in natural language processing and ontology enrichment.

**SENSUS:** The Natural Language group at ISI developed SENSUS to provide a broad conceptual structure for machine translation tasks. The content of this ontology was obtained from various electronic knowledge sources, and then organized into three regions. The upper region, that is also called the Ontology Base, contains general and essential items for linguistic processing. WordNet and an English dictionary are merged into the Ontology Base. The items in the middle region provide word senses in English. The lower regions contain more specific items, which are anchor points for different languages.

### Domain ontologies

Domain ontologies (Mizoguchi et al. 1995) represent specific concepts and relationships in a certain domain. The concept vocabularies are reusable in the same domain when developing another domain ontology. Numerous domain ontologies exist in areas such as science, e-commerce, enterprise, medicine, etc. Well known examples include:

**The United Nations Standard Products and Services Codes (UNSPSC):** The United Nations Development Program (UNDP) and Dun & Bradstreet developed the UNSPSC as a global commodity code standard, which organizes products and services in different segments. The coding system has a five-level taxonomy of products, which are Segment, Family, Class, Commodity and Business Function. Each level has a two-digit number and a description.

**Unified Medical Language System (UMLS) medical ontology:** The United States National Library of Medicine developed this as a large database to integrate massive numbers of biomedical terms from various sources. It has three parts. The Metathesaurus part contains biomedical information for each of a couple of million terms. The Semantic Network part is a top-level ontology with biomedical concepts and relations. The Specialist Lexicon has syntactic information about biomedical terms to support natural language processing.

**Chemicals Ontology:** It is composed of two ontologies, Chemical Elements and Chemical Crystals. The Chemical Elements ontology presents knowledge of the chemical elements in the periodic table such as chemical names and their properties. The Chemical Crystals ontology models crystalline structures of the chemical elements. The Chemicals Ontology is used for education and scientific discoveries.

#### Ontology Examples by Formality

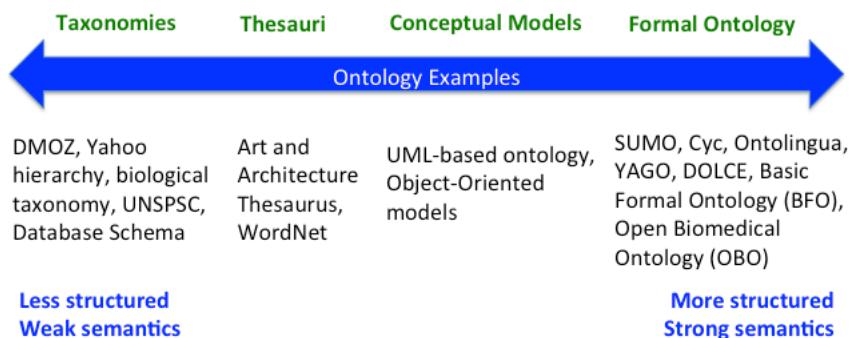


Figure 7.3: Ontology examples by their formality

A spectrum of ontology examples is presented in Figure 7.3. The ones located on the right side are more formal or heavyweight, and the ones on the left are more informal and lightweight. For example, formal ontologies such as SUMO and Cyc are based on logics. They contain axioms, rules, and various relationships among concepts; therefore, reasoning is possible. Informal ontologies have limitations regarding inference since they are not based on a formal logic and only contain simple relationships such as *is-a* or *part-of*.

#### 7.1.3 ONTOLOGY LANGUAGES

Ontology languages are used in constructing and reasoning ontologies. Various languages are shown in Figure 7.4. They were built from different disciplines. For example, a language

## 224 7. ONTOLOGIES

based on the first order logic (e.g., KIF) was developed from the Artificial Intelligence field from the early 1990s. It has formal rules, axioms, and theorems, which allow deducing a new theorem based on what we know already. General Knowledge Representation languages and systems also were used in ontology construction. In the Software Engineering field, ontologies were built using the Unified Modeling Language (UML), which is not as expressive as logic-based languages, but is able to show important entities and their relationships. Entity-Relationship diagrams are used to construct ontologies in the Database field. These languages are useful in addressing the different needs of disciplines.

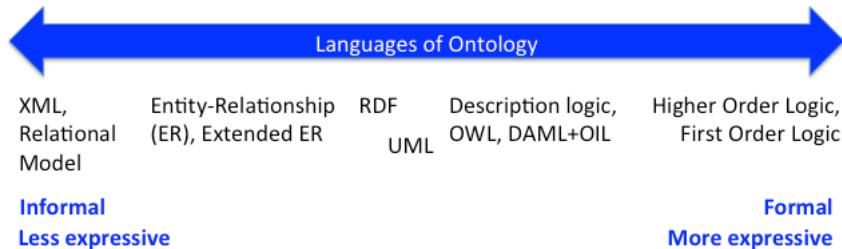


Figure 7.4: Ontology language examples based on their formality and expressivity

Selected examples of ontology languages are described below:

**Knowledge Interchange Format (KIF):** It was developed for the ARPA Knowledge Sharing Effort [459]. It is based on first order logic [239]. One of the related ontology languages is Ontolingua (Farquhar et al., 1997), which was built on top of KIF to address the problem of difficulty in creating ontologies using KIF. Ontolingua is based on frames and first order logic. It has LISP style syntax and had wide acceptance in 1990s. Figure 7.5 presents an example of KIF. It asserts that the number obtained by raising a real number  $?x$  with an even number exponent  $?n$  is greater than zero.

**Resource Description Framework (RDF):** The World Wide Web Consortium (W3C) developed RDF (<http://www.w3.org/TR/rdf-primer/>) to represent information in the Web [373]. It consists of three components: resources, properties, and statements. Resources are described with RDF expressions. A Uniform Resource Identifier (URI) and optional anchor identifiers refer to a unique resource in the Web. Properties define attributes and relations, which describe resources. Statements are composed of triples in RDF terminology, which have subjects, properties (i.e., predicates), and objects. The subjects are resources, and properties represent relationships or aspects of the resources between subjects and objects. For example, we can represent a sentence, *this coffee tastes bitter*, as

```
(=> (and (real-number ?x)
           (even-number ?n))
      (> (expt ?x ?n) 0))
```

Figure 7.5: An example of KIF representation

RDF triples. The subject is *this coffee*, the predicate is *tastes*, and the object is *bitter*. A collection of RDF statements forms a directed multi-graph. By using a query language such as SPARQL, we can infer specific knowledge from this RDF graph. Due to the need to define relationships between resources and properties, RDF Schema or RDFS was built. The combination of RDF and RDFS is referred to as RDF(S).

**OWL:** Building upon RDF(S), the OWL language was created to publish and share ontologies in the Web (Dean and Schreiber 2003). Like its predecessors, it has a layered structure: OWL Lite, OWL DL, and OWL Full. OWL Lite is intended to support users who need a classification hierarchy and simple constraints. OWL DL provides the maximum expressiveness and allows reasoning because of its correspondence with the description logic. OWL Full provides more flexibility to represent ontologies than OWL DL. It allows ontologies to augment the meaning of the pre-defined vocabulary. A class ‘Flight’ might be defined as a subclass of the class ‘Travel’ using OWL (Figure 7.6). Thus, attribute ‘flightNumber’ can have only one integer value, and the value for ‘transportMeans’ is ‘plane’.

## 7.2 LITERATURE REVIEW

### 7.2.1 ONTOLOGY ENGINEERING

More and more ontologies are being created and used in academia, e-commerce, businesses, government, and so on. Due the ability of ontologies to convey semantics, intelligent information systems and digital libraries that are based on ontologies might provide accurate information that users request. Uschold and Gruninger (2004) presents why ontologies and semantics-based technologies will play an important role in achieving seamless connectivity. They argue that getting the right information to the right people at the right time is the challenge of IT. For this, connecting people, software agents, and various IT systems is necessary. Researchers and businesses have been focusing on technologies for physical and syntactic connectivity. However, due to the lack of semantics in the data streams, the connection between people, software agents, and IT systems may not be effective. In addition, the physical coupling among the systems becomes less flexible. Incorporating semantics in an ontology might allow people, software agents, and IT systems to connect with each other in a deeper and more conceptual manner, thus achieving seamless connectivity.

## 226 7. ONTOLOGIES

```
<owl:Class rdf:ID="Flight">
  <rdfs:comment>A journey by plane</rdfs:comment>
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#Travel"/>
    <owl:Restriction owl:cardinality="1">
      <owl:onProperty rdf:resource="#flightNumber"/>
      <owl:allValuesFrom rdf:resource="xsd:integer"/>
    </owl:Restriction>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#transportMeans"/>
      <owl:hasValue rdf:datatype="xsd:string">plane</owl:hasValue>
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>
```

Figure 7.6: An OWL definition of the class ‘Flight’

Once we know the importance of ontologies, a next question to consider is how we can develop an ontology to meet our needs. It is important that a domain ontology has an accurate and comprehensive coverage of that domain. To ensure high quality, most ontology development works are conducted either manually or in a semi-automatic way (Gómez-Pérez & Manzano-Macho, 2004; Uschold & Gruninger, 2004). One of well-explained ontology development method is by (Noy and McGuinness, 2001). They present an iterative methodology of ontology development using Protg-2000 ontology editing environment in wine and food domain as an example. The emphasis is on its three fundamental rules: there is no one correct way to model a domain; ontology development is an iterative process; and concepts in the ontology should be close to objects and relationships in the domain of interest. In addition, their seven-step process explains the development details. The authors conclude with the remark that we can assess the quality of our ontology by using it in applications for which we design it.

Different development methods are sometimes merged together. Brusa et al. merged the iterative method by Moy and McGuinness as well as Methontology by Gomez-Perez et al. to develop a government budgetary ontology (Brusa, Caliusco, & Chiotti, 2006). They divide their ontology development into three phases. In the specification phase, the ontology goal and scope are determined. The domain of the ontology also is described. Motivating scenarios and competency questions are prepared. Then ontology granularity and type are decided. In the conceptualization phase, a domain conceptual model is defined. The authors used a Unified Modeling Language (UML) diagram to elaborate on the

main relations among defined concepts, which is common practice in the Software Engineering field. Relationships among classes and class attributes are identified. Instances are created. The last phase is implementation. The importance of modularizing the ontology for extensibility and reuse was discussed. The Protégé 3.1 ontology editor was used in their study.

To address the problem of time- and effort-intensive manual methods, Sanchez and Moreno developed a semi-automatic methodology to extract information from Web documents to construct an ontology (Sanchez and Moreno, 2004). The procedure involves finding keywords that are representative of the domain of interest, fetching of the resulting Web documents using a publicly available search engine (e.g., Google), and filtering duplicate documents. Then an exhaustive analysis is performed to extract information from each document. After stemming and stop-word removal procedures, a statistical analysis helps to select the most representative ones. The whole procedure iterates with a new key phrases that is constructed by joining the found concepts and the original concept. The ontology is finally refined to obtain a more compact taxonomy and avoid redundancy. Other methods that are based on extracting information from text are also provided by Balakrishna (2010) and Storey (2005).

Instead of creating ontologies from scratch, we may reuse already developed ontologies. One method is to merge more than one ontology into a single ontology to expand its coverage of a domain. Stumme and Maedche [613] introduce the method, FCA-MERGE, which is used to develop federated and autonomous Web system by merging specific ontologies. They apply techniques such as natural language processing and formal concept analysis to derive a lattice of concepts as a result of FCA-MERGE. Humans then explore and transform these concepts into a merged ontology. The other method is an ontology mapping. In a distributed environment like the Semantic Web, several applications need to access multiple ontologies that have been developed. These ontologies might be mapped to a common layer, where information can be shared in semantically sound ways. In a survey paper, Kalfoglou and Schorlemmer (Kalfoglou and Schorlemmer 2005) present a formal mathematical definition of ontology mapping, along with a review of 35 works such as MAFRA, OIS, OntoMapO, and Information Flow (IF)-Map with some example cases. However, among these methods there is not a single method that is fully automatic. This is pointed out as one of the biggest challenges for ontology mapping when we consider the proliferation of ontologies and agent technologies.

### 7.2.2 ONTOLOGY AND DIGITAL LIBRARIES

The Digital Libraries (DL) field has been adopting ontologies. (Liu et al. 2008) present their study on intelligent information retrieval services in DLs. Based on their ontology of scientific documents, user queries are expanded to include semantically relevant terms for better search results. Another study in this line of research is by (Xu et al. 2008). As an

## 228 7. ONTOLOGIES

attempt to go beyond keyword matching-based retrieval in DLs, they provide algorithms to incorporate the WordNet lexical database in constructing an ontology as well as in expanding queries.

Other studies of semantic searching are conducted by (Ding et al., 2004; Lei et al., 2006; Shah and Joshi, 2002). In JeromeDL, community-oriented services as well as semantic empowered services augment the traditional DL services (Kruk et al., 2008). The metadata of the DL resources are semantically ‘ontologized’, and the resources are tagged and filtered by the community of people to support semantic and social searching. CallimachusDL is another semantics-based DL, which provides faceted search. It also integrates both social media and multimedia elements in a semantically-annotated repository (Garca-Crespo et al., 2010).

### 7.3 FORMALIZATION

#### 7.3.1 THE BIG PICTURE

(This section is largely adopted from Guarino et al. 2008 ‘What is an ontology?’ It will be revised and updated.)

Ontologies attempt to capture what we perceive and conceptualize from surrounding real world phenomena using abstract and concrete entities as well as their relationships. This is summarized in Figure 7.7. It shows that phenomena occur in real world. Their presentation patterns are perceived at different times, and then they are constructed as an abstracted conceptualization. This conceptualization is interpreted as models using a language. Good ontology models approximate the intended model as much as possible.

Guarino et al. (2008) provide formal definitions of the three aspects of an ontology as considered by Studer et al. (1998):

- What is a conceptualization?
- What is a proper, formal, explicit specification?
- Why is ‘shared’ of importance?

#### Conceptualization

Despite the complexity of the notion of conceptualization, Genesereth and Nilsson choose to explain it by using a very simple mathematical representation: an extensional relational structure.

Definition 2.1 (Extensional relational structure): An extensional relational structure (or a conceptualization according to cite (Genesereth and Nilsson)), is a tuple  $(D, R)$  where:

- $D$  is a set called the universe of discourse
- $R$  is a set of relations on  $D$

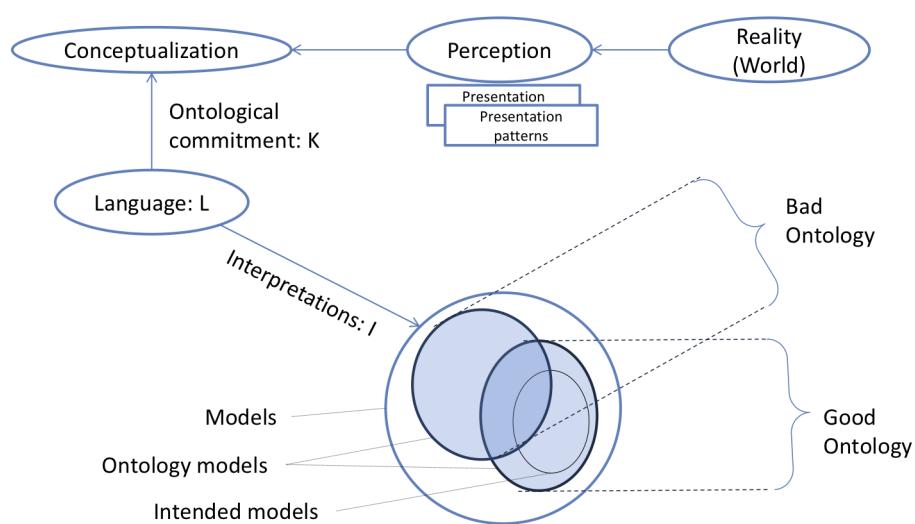


Figure 7.7: From real world to models and to ontologies (adapted from (add citation)).

## 230 7. ONTOLOGIES

Note that, in the above definition, the members of the set  $R$  are ordinary mathematical relations on  $D$ , i.e., sets of ordered tuples of elements of  $D$ . So each element of  $R$  is an extensional relation, reflecting a specific world state involving the elements of  $D$ .

**Definition 2.2 (World):** With respect to a specific system  $S$  we want to model, a world state for  $S$  is a maximal observable state of affairs, i.e., a unique assignment of values to all the observable variables that characterize the system. A world is a totally ordered set of world states, corresponding to the system's evolution in time. If we abstract from time for the sake of simplicity, a world state coincides with a world.

At this point, we are ready to define the notion of an intensional relation in more formal terms as follows:

**Definition 2.3 (Intensional relation, or conceptual relation)** Let  $S$  be an arbitrary system,  $D$  an arbitrary set of distinguished elements of  $S$ , and  $W$  the set of world states for  $S$  (also called worlds, or possible worlds). The tuple  $(D, W)$  is called a domain space for  $S$ , as it intuitively fixes the space of variability of the universe of discourse  $D$  with respect to the possible states of  $S$ . An intensional relation (or conceptual relation) of arity  $n$  on  $(D, W)$  is a total function:  $W \rightarrow D^n$  from the set  $W$  into the set of all  $n$ -ary (extensional) relations on  $D$ .

Once we have clarified what a conceptual relation is, we give a representation of a conceptualization in Definition 2.4.

**Definition 2.4 (Intensional relational structure, or conceptualization)** An intensional relational structure (or a conceptualization according to Guarino et al. (2008)) is a triple  $C = (D, W, R)$  with:

- $D$ : a universe of discourse
- $W$ : a set of possible worlds
- $R$ : a set of conceptual relations on the domain space  $(D, W)$

(add something about the language  $L$ )

The axioms for intensionally and explicitly specifying the conceptualization can be given in an informal or formal language  $L$ . Formal refers to the fact that the expressions must be machine-readable, hence natural language is excluded. Let us assume that our language  $L$  is (a variant of) a first-order logical language, with a vocabulary  $V$  consisting of a set of constant and predicate symbols (we shall not consider function symbols here). We introduce the notion of ontological commitment by extending the standard notion of a (extensional) first order structure to that of an intensional first order structure.

**Definition 3.1 (Extensional first-order structure):** Let  $L$  be a first-order logical language with vocabulary  $V$  and  $S = (D, R)$ , an extensional relational structure. An extensional first order structure (also called model for  $L$ ) is a tuple  $M = (S, I)$ , where  $I$  (called extensional interpretation function) is a total function  $I : V \rightarrow D^R$  that maps each vocabulary symbol of  $V$  to either an element of  $D$  or an extensional relation belonging to the set  $R$ .

Definition 3.2 (Intensional first-order structure) (also called: ontological commitment): Let  $L$  be a first-order logical language with vocabulary  $V$  and  $C = (D, W, R)$ , an intensional relational structure (i.e., a conceptualization). An intensional first order structure (also called ontological commitment) for  $L$  is a tuple  $K = (C, I)$ , where  $I$  (called intensional interpretation function) is a total function  $I : V \rightarrow D^R$  that maps each vocabulary symbol of  $V$  to either an element of  $D$  or an intensional relation belonging to the set  $R$ . It should be clear now that the definition of ontological commitment extends the usual (extensional) definition of ‘meaning’ for vocabulary symbols to the intensional case, replacing the notion of model with the notion of conceptualization. Let us introduce the notion of intended model with respect to a certain ontological commitment for this purpose.

Definition 3.3 (Intended models): Let  $C = (D, W, R)$  be a conceptualization,  $L$  a first-order logical language with vocabulary  $V$ , and ontological commitment  $K = (C, I)$ . A model  $M = (S, I)$ , with  $S = (D, R)$ , is called an intended model of  $L$  according to  $K$  iff. 1. For all constant symbols  $c \in V$  we have  $I(c) = I(c)$  2. There exists a world  $w \in W$  such that, for each predicate symbol  $v \in V$

### **Ontology models**

There exists an intensional relation  $R$  such that  $I(v) = w$  and  $I(v) = (w)$  The set  $IK(L)$  of all models of  $L$  that are compatible with  $K$  is called the set of intended models of  $L$  according to  $K$ .

The condition 1 above just requires that the mapping of constant symbols to elements of the universe of discourse is identical.

The condition 2 states that there must exist a world such that every predicate symbol is mapped into an intensional relation whose value, for that world, coincides with the extensional interpretation of such symbol. This means that our intended model will be so to speak a description of that world.

With the notion of intended models at hand, we can now clarify the role of an ontology, considered as a logical theory designed to account for the intended meaning of the vocabulary used by a logical language. In the following, we also provide an ontology for our running example.

Definition 3.4 (Ontology): Let  $C$  be a conceptualization, and  $L$  a logical language with vocabulary  $V$  and ontological commitment  $K$ . An ontology  $OK$  for  $C$  with vocabulary  $V$  and ontological commitment  $K$  is a logical theory consisting of a set of formulas of  $L$ , designed so that the set of its models approximates as well as possible the set of intended models of  $L$  according to  $K$ .

### **7.3.2 ONTOLOGY COMPONENTS**

(This section is adopted from Aubrecht et al. 2005 ”Ontology transformation using generalised formalism”. It will be revised and updated.)

## 232 7. ONTOLOGIES

Figure 8 presents that ontologies have two parts(Aubrecht et al., 2005). The structural part is divided into concepts and relations (Gruber et al., 1993). The procedural part consists of restrictions (constraints) and actions (rules). Thus, an ontology can be thought of as a 4-tuple: Ontology = (Concepts, Relations, Restrictions, Actions) Concepts = concept set

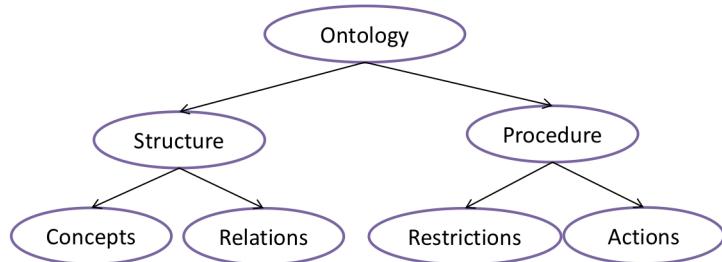


Figure 7.8: Ontology components.

## 7.4 ONTOLOGY ENGINEERING

There was a rapid proliferation of ontology developments beginning in the 1990's. Different groups pursued their own approaches. However, due to the lack of shared guidelines regarding development principles, processes, and methodologies, often these developments resulted in duplicate efforts and slow progress. In addition, the possibility of reusing and extending these ontologies for other applications was low. Accordingly, workshops on OE were held to explore and discuss the principles, rules and design decisions with the aim of identifying best practices.

Gómez-Pérez et al. define Ontological Engineering as ‘the set of activities that concern the ontology development process, the ontology life cycle, and the methodologies, tools and languages for building ontologies’ (Gómez-Pérez et al.2004). They elaborate on the activities to be performed in the Ontology Development process in Figure 7.9. The process consists of three parts: Management, Development, and Support. In this chapter, we focus on Development.

### 7.4.1 METHODOLOGIES

The Development portion of Figure 7.9 is divided into three phases. In the pre-development phase, a study of environment is carried out to identify the platform used as well as the application areas for the ontology. In the feasibility study, questions are asked regarding whether an ontology can be developed for that platform and application area. In the actual development phase, four activities are performed:

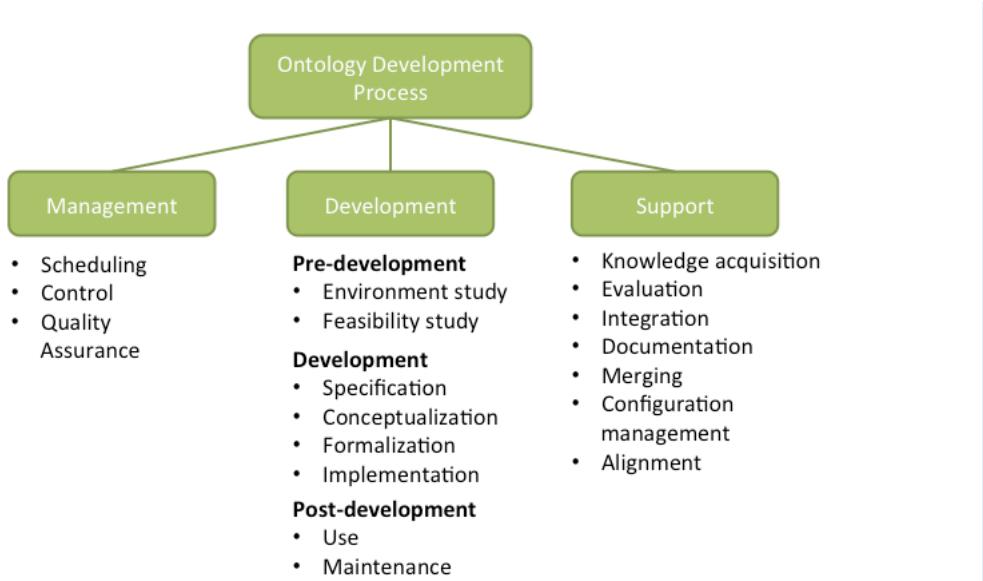


Figure 7.9: Ontology development processes

- Specification: goals, intended uses, and end-users are identified and documented.
- Conceptualization: domain knowledge becomes meaningful models at the knowledge level (Newell 1982).
- Formalization: the conceptual model is transformed into a formal and semi-computable model.
- Implementation: ontologies are built using an ontology language.

In the post-development phase, ontologies are updated through the maintenance activity, and used/re-used by other applications. Two examples of ontology development methods are presented below:

**The Cyc Method:** The Cyc knowledge base (KB) is one of the earliest ontologies. Its development started in the mid-1980's by manually adding more than a million assertions of common-sense knowledge about the world. For this, the following three-step processes were carried out (Lenat and Guha, 1990):

- Process I: Manual extraction of common sense knowledge. This knowledge was acquired manually in three steps:

## 234 7. ONTOLOGIES

- Encoding the knowledge required to understand books and newspapers: the knowledge that the authors of the books and articles expect that their readers already possessed was encoded.
- Examination of articles that are unbelievable: this was to study the rationale that makes some articles unbelievable.
- Identification of questions that anyone should be able to answer by having just read the text.
- Process II: Computer-aided extraction of common sense knowledge. Once enough common sense knowledge is gathered, tools that support natural language processing and machine learning might be used to search for new knowledge.
- Process III: Computer-managed extraction of common sense knowledge. Most work is performed by the system. Humans only recommend knowledge sources to the system.

In all three processes above, two activities were performed:

  - Activity 1: Development of a knowledge representation and top level ontology.
  - Activity 2: Representation of the knowledge of different domains.

There are several modules, which are integrated with the Cyc KB and CycL inference engine. For example, the WWW Information Retrieval module accesses the Cyc KB and extends it with the information available on the Web. These modules allow the Cyc KB to be applied in different contexts.

**Methontology method:** This method is based on both Software Engineering and Knowledge Engineering approaches. It includes techniques for the activities in the ontology development processes (Figure 7.9) and an ontology life cycle based on evolving prototypes. Among the activities in Development in Figure 7.9, the conceptualization activity requires special attention to avoid propagating errors because the next activities, formalization and implementation, are strongly dependent on it. The conceptualization activity converts informal views of a domain into semi-formal specifications using tables and graphical representations. These representations are useful in bridging the gap between humans' perception of a domain and the ontology languages used for implementation. A total of eleven tasks under the conceptualization activity are:

- Task 1: To build the glossary of terms. All the relevant terms in a domain such as concepts, instances, attributes, relations, natural language descriptions, synonyms, and acronyms are collected and organized in a table.
- Task 2: To build concept taxonomies. Based on the glossary of terms, sets of disjoint concepts, in other words concepts that cannot have common instances, are identified and their concept hierarchy is constructed.

- Task 3: To build ad hoc binary relation diagrams. Diagrams that show relationships between pairs of concepts in the concept taxonomy are created.
- Task 4: To build the concept dictionary. The concept dictionary shows concepts, their relationships, properties, and optionally instances, in a table.
- Task 5: To define ad hoc binary relations in detail. Each relation in the concept dictionary is elaborated with its source concept, target concept, source cardinality, mathematical properties, and its inverse relation.
- Task 6: To define instance attributes in detail. An instance attribute table is formed based on elaborating each attribute with its concept name, value type, range of values, etc.
- Task 7: To define class attributes in detail. A class attribute table is formed with each attribute name, value type, precision, cardinality, value, etc.
- Task 8: To define constants in detail. A constant table is formed based on detailed descriptions of constants.
- Task 9: To define formal axioms. A formal axiom table, where each axiom is specified with its name, description, expression, concepts, variables, etc., is constructed.
- Task 10: To define rules. After needed rules are identified, they are described in the rule table. Each rule will have its name, description, expression, concepts, etc.
- Task 11: To define instances.

At this point, an ontology conceptual model is developed. The relevant instances appearing in the concept dictionary are identified and described into an instance table. The name of the instance, related concept, and attribute values are added if they are known. These tables and graphical representations are further formalized and implemented using ontology editors. For example, the WebODE ontology editor can translate conceptual models into several ontology languages.

#### 7.4.2 TOOLS

Ontology development tools can be used to build an ontology from scratch. Basic features include editing and browsing of the ontology. Export/import for different formats, graphical editing, and documentation are often provided. Ontology merging tools identify conflicting concepts between two ontologies, so that users may resolve the issues semi-automatically. Ontology annotation tools allow users to insert instances of concepts and relations into existing ontologies.

Inference engines make querying of ontologies easier. They are usually dependent on the ontology language used. Ontology learning tools extract concepts and relations from a

## 236 7. ONTOLOGIES

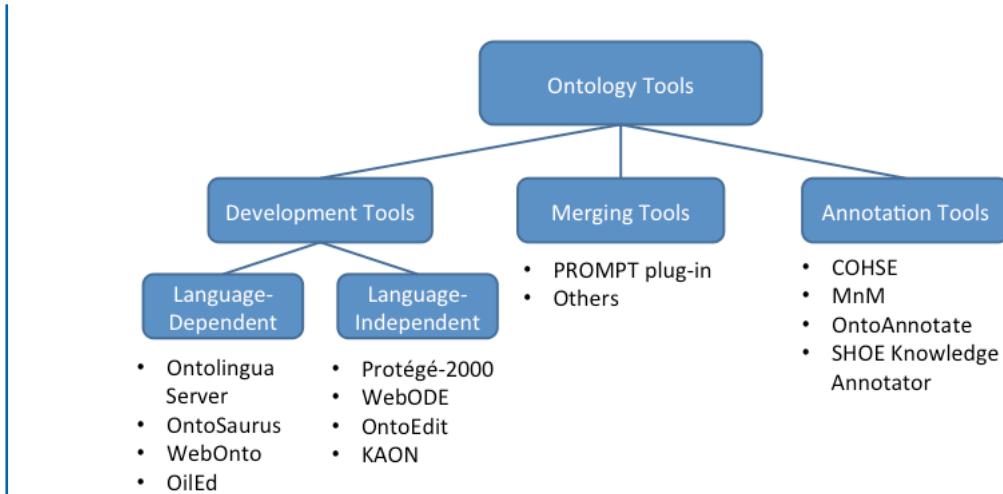


Figure 7.10: Ontology tools for building, merging, and annotation (citation)

textual corpus and can build a lightweight ontology semi-automatically. Natural language processing and machine learning techniques are used for this.

Two examples of ontology development tools are introduced:

**Protégé Editor and Framework:** The Protégé ontology editor and framework is widely-used. It supports modeling ontologies in two ways with its Protégé-Frames editor and Protégé-OWL editor:

- Frame-based ontology: Concepts are organized in classes, which have subsumption relationships. A set of slots in a class describe properties and relationships. Also a set of instances of a class is associated with the class.
- OWL-based ontology: OWL ontologies, suitable for the Semantic Web, include class descriptions, properties and instances. Logical reasoning is supported by inference engines, which can be installed as plug-ins.

In addition, there is a WebProtégé ontology editor, which is running on a server to allow collaborative ontology building. The developed ontologies can be exported to various formats such as OWL, RDF(S), and XML Schema. The editor can be extended with plug-ins or a supported API to develop knowledge-based applications.

**Sigma Ontology Development Environment:** This system integrates multiple tools to develop formal ontologies. Its primary component is ontology editing and browsing employing the Knowledge Interchange Format (KIF) (Genesereth 1991). Two types of browsers

exist. One shows a textual hierarchy where the other presents an automatic graph layout. An inference system that works with first order logic and natural language/logic translators also is included. Some of features are:

- Language generation: The formal statements written in SUO-KIF format can be paraphrased similar to a natural language. For example, a SUMO term *DiseaseOrSyndrome* is translated into *disease or syndrome*
- Natural language understanding: The system can translate a restricted English sentence (i.e., present tense, singular, ambiguous words are assigned the most popular sense, etc.) into KIF format based on terms from the SUMO upper ontology. Details are presented in (Murray, Pease & Sams, 2003).
- Inference, reasoning, proof, proof with equality: A previously unknown fact might be deduced using Sigma's inference engine.

Although the Sigma tool is designed to work with various ontologies, more features are available when a standard ontology, Standard Upper Merged Ontology (SUMO) (Niles & Pease, 2001), is used. A number of domain ontologies from e-commerce, governmental organizations, biological viruses, etc. have been created to extend SUMO.

#### 7.4.3 REASONING ONTOLOGY

Formal ontology languages have their own reasoning mechanisms as well as knowledge representations. There exists a tradeoff between the two. When a language is more expressive, the inference engine should create more complicated results with the corresponding mechanism. The inference engines have features below:

- Automatic classifier (for the DL-based languages): It computes the concept taxonomy from the ontology concept definitions.
- Inheritance (simple or multiple) management: Concept attributes and relations are managed using the concept taxonomy.
- Exceptions management: There might be conflicts in the property values of concepts. For example, a concept bird can have a property flies with a value true. An exception cases are ostrich and penguin. They are birds but cannot fly. An inference engine should deal with these situations.

### 7.5 ONTOLOGY APPLICATIONS

#### 7.5.1 DIGITAL LIBRARIES

Ontologies are used in digital libraries for many purposes: Domain modeling (representation), content annotation, providing semantic concepts and relationships, formal description, and representation of all aspects of digital libraries.

## 238 7. ONTOLOGIES

Ontologies are also used to enhance performance of services in digital libraries. Scholar ontology [600] is an ontology that maintains a semantic network of scholars interpretations, comments, and analyses of literature work. The ontology gives semantics over the existing research papers metadata and allows for better engagement of researchers in analyzing and asserting their claims on papers related to their work.

In [184], an ontology describes the different scenarios of using an e-learning digital library. A personalization system was built based on an ontology for describing and organizing user profiles, user preferences, navigation profiles, user actions, and the relationships between all these information instances. The ontology also was used to describe the scenario of new functionalities that a user likes to add into the system. The new functionality is implemented by integration of semantic web services and the ontology.

### 7.5.2 SEMANTIC WEB

The Semantic Web is about two things. It is about common formats for integration and combination of data drawn from diverse sources, in contrast to the original Web that mainly supports the interchange of documents. It is also about language for recording how the data relates to real world objects. That allows a person, or a machine, to start off in one database, and then move through an unending set of databases which are connected not by wires but by being about the same thing. [655]

Figure 7.11 shows the technology stack used in the Semantic Web. The ontology layer is built upon the RDF Model where entities are described as triples. The ontology also adds the relationship between these entities and allows for querying and reasoning.

Semantic web enables webpage's author to describe the semantic concepts of the webpage content by using ontologies. An entity in the webpage can be described by identifying the entity or concept in the ontology that corresponds to the semantic of the entity in the webpage. More semantics can be inferred using the relations between concepts in the ontology.

Figure 7.12 shows an example of an ontology describing software programs and their manuals. The software ontology describes the entities existing in the software domain: software, library, documents, images, topics, persons, and places. The ontology also describes the relations between these entities: hasManual, requires, isBasedOn, inPartOf, has Author, livesAt, and subject.

The information presented in Figure 7.12 can be easily inferred by any person reading a webpage, but can not be inferred by computers without the direct help of a person. Ontology plays the role of helping computers to infer information about webpages and how to relate this information to other sources of information.

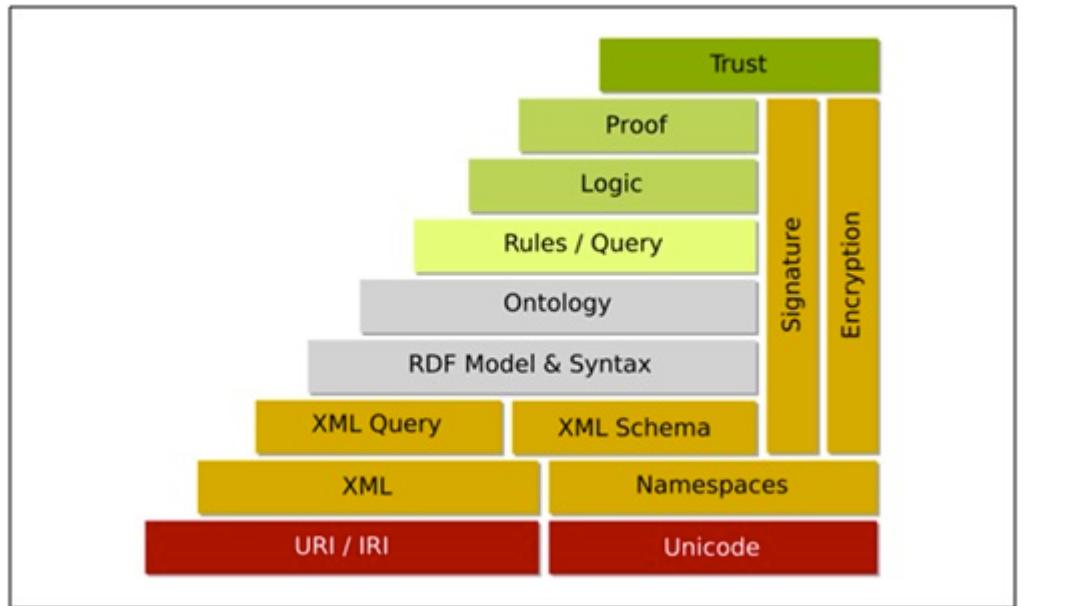


Figure 7.11: Technology stack used in Semantic Web []

### 7.5.3 FOCUSED CRAWLING

World Wide Web (WWW) is a huge source of different types of data and information. Many entities are interested in collecting data from the WWW. Search engine companies, business companies, as well as marketing and advertising organizations all seek information from the WWW, but for different needs and according to different perspectives. The huge evolving structure of the WWW made it difficult for the information seekers to use the traditional approaches of information retrieval where huge amount of resources are used. A different approaches have been developed for retrieving information with the minimum resources that can be used.

One of the most important services used in many applications on the WWW is crawling. Crawling is the process of retrieving web pages that are linked from a starting web page(s). The starting web pages are called seeds. Crawling is used in search engines for indexing web pages on the WWW for faster search results. There are several issues in the crawling process such as: checking web pages for update, permission for crawling, hidden web, avoiding loops, and parallel crawling.

Crawling can be used for building a customized collection, e.g., a collection of web pages that are related to a certain topic. This type of crawling is called focused crawling. Avoiding crawling the huge WWW, however, leads to less calculation while crawling iden-

## 240 7. ONTOLOGIES

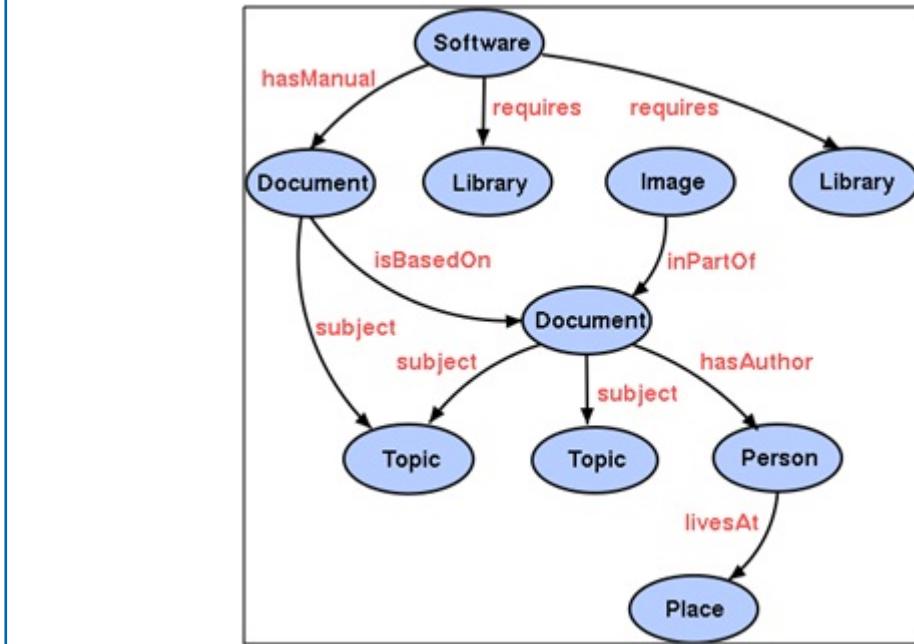


Figure 7.12: Example of Ontology: Software concepts and their relations []

tified web pages. A decision needs to be taken on every web page visited in the crawling phase. This decision is based on some calculation that leads to either retrieving this web page or skipping it, e.g., when not relevant to a topic.

The quality of the webpages collected using focused crawling can be measured, e.g., using two metrics: Precision and Recall. Precision is  $RC / TC$  and Recall is  $RC / AR$ , where  $RC$  is the number of collected webpages considered relevant to the topic specified,  $TC$  is the total number of webpages collected, and  $AR$  is the total number of webpages degree of relevant to the topic specified.

There are several approaches for calculating the degree of relevance of a webpage to a certain topic. The topic is specified by a number of keywords. The text of the webpage is analyzed using text analysis algorithms and a score is given to the webpage according to the occurrences of the topic keywords in the webpages text.

Another approach uses the URL text plus the webpage content. The URL text alone sometimes gives little indication regarding the topic of the webpage. Hence, one can supplement by adding in the anchor text or other context of the URL.

The representation of the topic of interest plays an important role in the performance of the focused crawling and the quality of the resulting webpages. A naive approach is using keywords or sample webpages. The performance of focused crawler using keywords suffers

from semantic problems which lowers the quality of the webpages collected. A word can have several meanings in different contexts and environments which could mislead a focused crawler based only on keyword matching techniques.

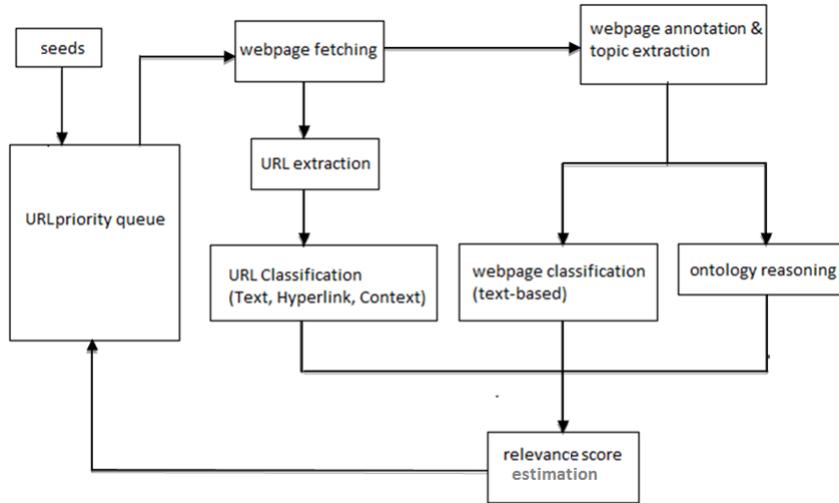


Figure 7.13: Architecture of Ontology-based Focused Crawler

An ontology can help solve the semantic problem faced by the focused crawler by better representing the topic of the domain. The ontology is used for determining whether the webpage is relevant or not. The text of the webpage is analyzed to get the semantics or the topic of the webpage. Then the topic produced is used to reason the ontology to check the relevance of the webpage .

## 7.6 ONTOLOGY EVALUATION

There are different kinds of approaches for evaluating ontologies. Ontologies are used in many fields and applications. Finding the best suitable ontology for a certain problem depends on the type of application or domain the ontology will be used in, the way the ontology will be used, and the purpose of using the ontology.

Ontology evaluation can be defined as the process of measuring the functionality of the ontology [482]. This consists of 3 major tasks: ontology verification, ontology validation, and ontology assessment (Figure 7.14).

Ontology validation is the process of checking that the ontology properly describes its entities and concepts. Ontology verification is the process of checking that the ontology

## 242 7. ONTOLOGIES

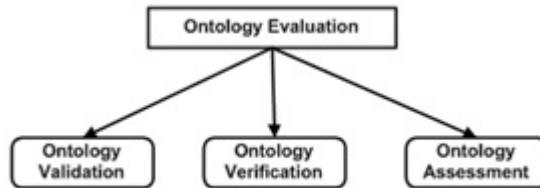


Figure 7.14: Components of ontology evaluation, [482]

Level	Approach to evaluation			
	Golden standard	Application-based	Data-driven	Assessment by humans
Lexical, vocabulary, concept, data	x	x	x	x
Hierarchy, taxonomy	x	x	x	x
Other semantic relations	x	x	x	x
Context, application		x		x
Syntactic	x <sup>1</sup>			x
Structure, architecture, design				x

Figure 7.15: Ontology evaluation approaches for different levels

describes the right entities and concepts. Ontology assessment is the process of measuring the usability of the ontology from the point of view of the user. The ontology evaluation tasks require the existence of a reference model for the domain of application such as a requirements specification, or a real-world problem definition which can be used in the validation and verification process. Some measures have been identified for ontology evaluation: completeness, consistency, conciseness, expandability, and sensitivity [482].

Another classification for ontology evaluation approaches is shown in Figure 7.15 [81]. The classification has levels which describe the different aspects of the ontology. [81] demonstrates that the best approach for evaluation depends on the purpose of the evaluation, the application that the ontology was built for, and the aspects of the ontology that are being evaluated.

## 7.7 CASE STUDY: CRISIS, TRAGEDY, AND RECOVERY (CTR) ONTOLOGY

The Crisis, Tragedy, and Recovery Network (CTRnet) project has been collecting news and online resources that are related to natural disasters (e.g., wildfires, floods, typhoons, and earthquakes) and man-made tragedies (e.g., campus shootings in the USA and internationally). Since the summer of 2010, the project group also has been working on collecting and providing CTR-related information from social media such as Twitter and Facebook.

Our goal for the development of the CTRnet digital library is to collect, organize, and serve resources that can cover the disaster and emergency management domain comprehensively. Without knowing which concepts are important and how they are related with one another, we may not see the big picture. This might lead the CTRnet DL to collect and provide resources that are unbalanced in covering the domain. For this reason, we are developing a CTR ontology with in-depth coverage. The CTR domain is broad. Therefore, having solid domain knowledge will help to collect and organize resources. It also will help visitors navigating through information in the CTRnet digital library.

### 7.7.1 APPROACH

*Semi-automatic ontology development:* To ensure high quality as well as to make the development effort scalable, it makes sense to create the ontology using a semi-automatic methodology that involves the least amount of human intervention as well as computational effort such as Natural Language Processing (NLP) (Balakrishna, 2010). The ontology building in this study has two parts. Part 1 is to merge multiple related ontologies, which have been built from existing disaster databases. Part 2 is to expand the ontology with related concepts.

*Part 1. Merging ontologies into a global CTR ontology:* Our initial effort was to find online information sources about disaster management, where we can identify disaster-related concepts and their hierarchical relationships. The disaster databases we selected include:

- EM-DAT: The International Disaster Database [190]
- The Disaster Database from University of Richmond [484]
- Canadian Disaster Database [95]
- DesInventar Disaster Inventory System [492]

We started with the disaster classification in the Richmond database, and then merged it with the EM-DAT database. The process went smoothly, with the Richmond database providing more leaves and the EM-DAT elements comprising more high-level concepts. Next

244 7. ONTOLOGIES

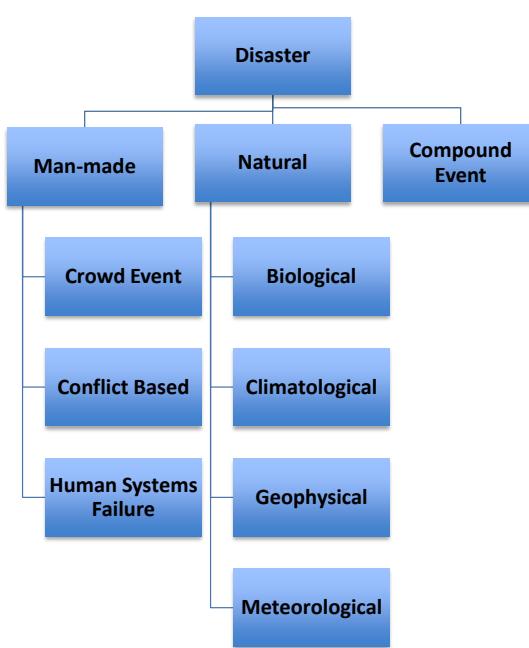


Figure 7.16: Highest level concepts from the current CTR ontology

we merged the Canadian disaster database into the ontology, assigning each of its concepts to existing concepts in the draft ontology or creating a new concept in the ontology.

The overall hierarchy of the draft ontology was stable through this process with most of the Canadian disaster database mapping to leaf concepts in the ontology. Finally, we merged the DesInventar disaster inventory system into the merged ontology that had been built from the other three databases. A large majority of DesInventar elements matched the partial union ontology. However, the concept of the ‘cause’ of a disaster was not included in any of the other databases. At first we decided to exclude this element from the merged ontology due to the lack of consensus across databases. However, DesInventar includes an extensive set of slots to be filled for every disaster, so we later adopted them for integration with the ontology.

The resulting CTR ontology has a total of 185 concepts. The number of man-made disaster concepts is 140. Figure 7.16 presents the highest-level concepts in the current CTR ontology.

This was not a trivial task due to the ambiguities of natural language as well as different scopes and hierarchical structures of the disaster databases. Although ontology

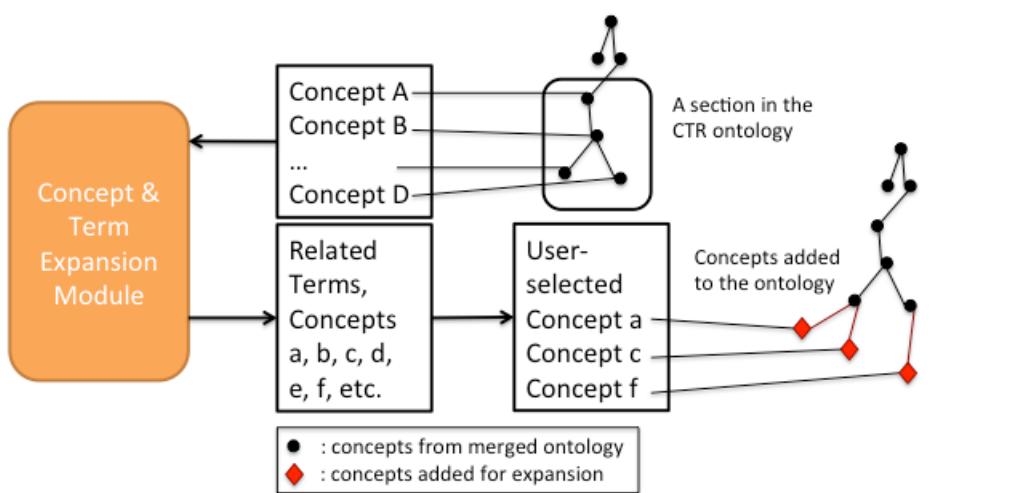


Figure 7.17: An ontology concept expansion process

merging algorithms have been developed (Stumme & Maedche, 2001), human involvement is still a necessity to ensure accuracy.

**Part 2. Enriching the CTR ontology:** Figure 7.17 illustrates a process whereby concepts in the merged CTR ontology might be expanded into related concepts. First, a group of concepts is selected from a region in the ontology. These concepts should have close relationships, for example, they are from the volcanic eruption part of the ontology. The Concept and Term Expansion Module (CTEM) accepts these concepts as an input and generates potentially related terms. Humans then identify appropriate concepts from the generated result, and add them to the existing ontology hierarchy. By iterating this process for different regions of the ontology, we might successfully broaden the coverage of the CTR ontology. As we mentioned above, we follow this semi-automatic ontology development approach to ensure high quality ontology expansion.

A conceptual diagram of an expanded ontology is shown in Figure 7.18. Black round dots represent concepts that are from the original ontology, while red diamonds represent concepts newly added through the expansion process. We may iterate the enriching process using the incrementally expanded ontology.

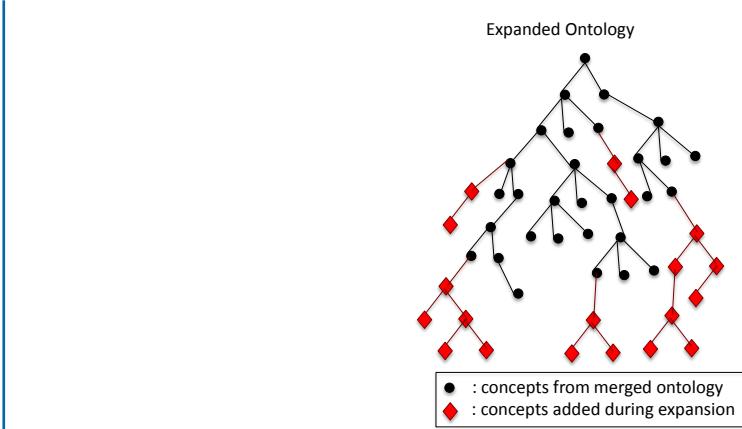


Figure 7.18: A conceptual diagram of an expanded ontology

## 7.8 EXERCISES AND PROJECTS

1. How can end-users work effectively with large ontologies that are integrated into a digital library?

## CHAPTER 8

# Chapter 8: Classification

by Venkat Srinivasan and Pranav Angara *Abstract:* When interacting with a large collection of documents, being able to browse them by topics can be particularly useful.

### 8.1 INTRODUCTION

Advances in electronic publishing, coupled with increased access to the Internet, have paved the way on a scale never seen before, for production and dissemination in electronic format of books, as well as book-sized documents (henceforth referred to as long documents) like legal reports and Electronic Theses and Dissertations (ETDs). Not only is this type of born-digital content being produced at a rapid rate, there are also numerous efforts to digitize existing paper collections. In the context of books, for example, the Million Books project involving Carnegie Mellon University and others, and Google's efforts on digitizing 10 million books, are two significant efforts to make accessible a large collection of e-books based on what currently is only available on paper. The Networked Digital Library of Theses and Dissertations (NDLTD) already includes about 2 million metadata records for ETDs.

One very exciting and critical use for these emerging collections of open access (or at least easily accessible) large sized documents, is in support of research and scholarship. Long documents like ETDs and e-books are richer in information content than short papers. User friendly modes of access to vast collections of such documents will be of invaluable aid to the scholarly community, as well as the general public.

However, even though such collections of long documents are growing at a rapid rate, techniques for managing and providing access to them have rarely progressed beyond decades old efforts based on fulltext searching and browsing. Large ETD collections, for example, are being made accessible through the NDLTD website <sup>1</sup> mostly via a keyword search and browse interface. There is very little use of advanced text analysis techniques or classification techniques, for example. Evaluation studies suggest that effectiveness and usability, as measured in various ways, is low [58, 259, 261].

In this research we are proposing to develop advanced text analysis based techniques, particularly based on text classification, to develop a Digital Library (DL) which makes vast collections of ETDs more accessible and usable for the scholarly community, and for the public in general. Consistent with the overall theme of the book, we also show how the 5S formalisms help in this regard.

<sup>1</sup><http://www.ndltd.org/find>

## **248 8. CHAPTER 8: CLASSIFICATION**

The chapter is organized as follows. In the rest of this section we discuss the usefulness of this effort, and some of the research questions we address. In Section 8.2 we briefly review some prior and related work in the field of text classification, and also provide relevant definitions and background. In Section 8.3 we discuss the relevant 5S formalisms, and how they apply to this particular problem situation. In Section 8.4, we discuss in detail the ETD classification and the DL building process.

### **8.1.1 INTELLECTUAL MERIT**

ETDs form an important testbed for several reasons. They are a key part of global scholarship, and are valuable resources, with comprehensive coverage of the topic of study. Each ETD summarizes specialty related literature, and has a large bibliography section. There is local quality control, and some are reviewed by external experts. Rarely is research described elsewhere in enough detail so reproducibility of science can be tested. Easier access to ETDs thus would be a valuable aid to scholarly activities.

It is in this context of ETDs that we propose to develop our research methodology, opening up further an important genre that now can be accessed only with fulltext search through production systems like Scirus<sup>2</sup>. More specifically, we are interested in developing methods for providing improved access to vast collections of ETDs to aid research and development activities of the scholarly community.

We are classifying ETDs into a topical taxonomy to facilitate browsing and searching. Taxonomies have been used for many years and in many disciplines to group (similar) information together into (similar) categories. Taxonomies mirror the mental model that humans use to organize information. Identifying constituent topics of an information rich and (likely) multi-topic document, and subsequently mapping the ETD to the corresponding node(s) in the taxonomy, will make navigating the huge ETD collection easier.

### **8.1.2 ETDS AND NDLTD**

The source of ETDs in this study is the NDLTD Union Catalog. The NDLTD initiative, which was started in the 90s, sought to expand electronic publication, and make accessible, ETDs from around the world. Our testbed is a collection of ETDs harvested using metadata from the NDLTD Union Catalog [480].

The Union Catalog consists of metadata records for ETDs from contributing member institutions (universities) around the world. As of March 2010, the Union Catalog consisted of metadata records for 820,000 ETDs in various languages, and is the single largest cumulative source of information on ETDs available on the Internet. The metadata format used is known as the Dublin Core [666], which is used for describing electronic resources, and consists of the following 15 fields for each resource: Title, Creator, Subject, Description, Publisher, Contributor, Date, Type, Format, Identifier, Source, Language, Relation,

<sup>2</sup><http://www.scirus.com/>

## 8.1. INTRODUCTION 249

identifier	<a href="#">oai:VIETD:etd-11172009-055013</a>
datestamp	2009-12-16
All	<a href="#">VIETD</a>
dc:title	A Comparison of Criteria used in Gifted Identification in the Commonwealth of Virginia
dc:creator	Palmer, Karen Smith
dc:subject	Educational Leadership and Policy Studies
dc:description	In the Commonwealth of Virginia, gifted education plans are submitted to the state every five years for state approval. The plans must indicate the use of a minimum of four criteria out of the eight criteria provided by the Commonwealth in the identification process. The concept of using multiple criteria stems from research. Research has shown that the criteria used in the identification of gifted students affect the proportion of identified students as well as the proportion of underrepresented (Davison & Cross, 2002). Research has also shown that the use of multiple criteria leads to a higher proportion of underrepresented students identified (Callahan, Tessmer, Adams, & Moore, 2002). The purpose of this study was to compare the gifted identification criteria used within the Commonwealth of Virginia's public school divisions and analyze the effects of the criteria on the percentages of underrepresented gifted within the divisions. In this study, the researcher analyzed the numbers of each minority in the total populations against the total gifted minority populations to identify those divisions that were prone to practice for traditionally underrepresented minorities. Aspects of the gifted identification process in which changes were then analyzed. The aspects of the gifted identification process in the proposed division and several divisions were evaluated in the identification process. Findings revealed that there were no divisions with reported minorities that were proportional in all traditionally underrepresented ethnicities. In addition, no one specific standardized measure was successfully used in identifying non-traditionally gifted minorities in all ethnic groups. The implication that can be drawn from this research is that despite all attempts to put research into practice by using multiple criteria in the identification of the gifted, there is no one criterion that ensures the proportional identification of underrepresented minorities.
dc:contributor	Dr. Carol Cash
dc:publisher	VI
dc:date	2009-12-08
dc:type	text
dc:format	application/pdf
dc:identifier	<a href="http://scholar.lib.vt.edu/theses/available/etd-11172009-055013/">http://scholar.lib.vt.edu/theses/available/etd-11172009-055013/</a>
dc:source	<a href="http://scholar.lib.vt.edu/theses/available/etd-11172009-055013/">http://scholar.lib.vt.edu/theses/available/etd-11172009-055013/</a>
dc:language	en
dc:rights	unrestricted
dc:rights	I hereby certify that, if appropriate, I have obtained and attached hereto a written permission statement from the owner(s) of each third party copyrighted matter to be included in my thesis, dissertation, or project report. I further certify that the version I submitted is the same as that approved by my advisory committee. I hereby grant to Virginia Tech or its agents the non-exclusive license to archive and make accessible my thesis, dissertation or project report in whole or in part in all forms of media, now or hereafter known, I retain all other ownership rights to the copyright of the thesis, dissertation or project report. I also retain the right to use in future works (such as articles or books) all or part of this thesis, dissertation, or project report.

Figure 8.1: A Sample ETD Record in the NDLTD Union Catalog

Coverage, and Rights. A sample metadata entry for an ETD in the Union Catalog is shown in Figure 8.1.

The Union Catalog also contains ETDs in languages other than English. For our study however we are focused on ETDs that are in English.

### 8.1.3 PROBLEM SUMMARY

A formal definition of the problem is as follows: Given a large collection of ETDs, and a topical taxonomy, identify the node in the taxonomy that each of the ETDs will be mapped into, depending on its topic(s).

In this chapter we present some of our work so far towards this goal. We specifically focus on the use of metadata information associated with an ETD to aid in identifying the category of the ETD. We also limit ourselves to identifying only the major topic of the ETD, leaving the task of identifying other topics as future work.

### 8.1.4 RESEARCH QUESTIONS

We will investigate the following research questions in the context of this work:

- *How much information does the metadata associated with ETDs provides towards category identification?*
- *Which of the metadata fields are most useful in category identification?*
- *Which automated text categorization algorithms perform best at this task?*

### 8.1.5 CONTRIBUTIONS OF THIS PROJECT

The following are the major contributions of our work on ETD categorization:

## 250 8. CHAPTER 8: CLASSIFICATION

- The literature accessible for aid with scholarly work so far has been limited to books, research papers, etc. Our work with ETD categorization is expected to make vast collections of ETDs more accessible to the scholarly community and the general public.
- By associating ETDs with topics that are organized systematically according to a widely used (and hence familiar) topical taxonomy like DMOZ, the ETD collections are easier to navigate and knowledge discovery is made easier, as compared to when all that is possible is searching by keywords.
- Tools from our work can be used for automated categorization of books and ETDs - a task currently being done by trained catalogers. Libraries and universities thereby are expected to save money spent toward these efforts. Further, in cases when no catalogers are available, works could still be categorized.

## 8.2 RELATED WORK

While text classification has received considerable attention in the research community [590], hierarchical text classification at best has seen only limited interest. In this section we discuss some representative works from the literature relating to hierarchical text categorization.

In order to aid the users' understanding of the material presented in this chapter, we first provide some relevant, but informal, definitions from the text classification field. Formal definitions of these terms are provided in Section 8.3.

### 8.2.1 DEFINITIONS

**Text Classification:** Refers to the process of identification of class(es) or topical categories of text documents from among  $k$  existing (and pre-defined) categories.

**Supervised Learning:** Supervised learning refers to a suite of machine learning techniques wherein the computer or the algorithm *learns* by means of examples provided to it. Learning typically is defined according to the task at hand, for example learning to identify hand written characters, or identifying e-mails as spam or non-spam.

**Classifier:** A classifier is a supervised machine learning algorithm which learns through examples to identify documents as belonging to one or more of  $k$  existing classes.

**Training Set:** A training set is a set of documents that are labeled by the respective topical categories to which they belong. These documents serve as examples to *train* the classifier to learn to identify the categories of documents based on word occurrence patterns.

**Feature Selection:** Words occurring in a document are referred to as *features* of the document. Feature selection is a process which eliminates non-informative words (words that are unlikely to be helpful in identifying the category of the documents) in a principled way using various statistical measures ( $\chi^2$ , odds ratio, Mutual information, etc.).

**Evaluation:** Refers to the process of evaluating the effectiveness of the classifier that has been induced or trained using the training data. The commonly used evaluation metrics are Precision, Recall, and the F1 measure (defined below).

**Precision:** Generally a classifier is trained on a certain fraction of the training data and evaluated on the data that has been held out. Precision is the fraction of documents from the held out data set whose category is correctly identified by the classifier after having been trained by the training data.

**Recall:** Recall is the ratio of true positive (documents of a class correctly identified), to the sum of the true positive and false negative (documents of a class wrongly discarded) documents identified by the classifier.

**F1 Measure:** Refers to the harmonic mean of precision and recall (as computed below). Sometimes it is reported to give one single evaluation measure instead of multiple ones.

$$F_1 = \frac{2 * Precision * Recall}{(Precision + Recall)}$$

### 8.2.2 HIERARCHICAL TEXT CLASSIFICATION

Fundamentally, hierarchical text classification methods have differed in the following dimensions:

- Type of taxonomy chosen
- Methods for feature selection
- Methods for training set selection
- Type of classifier used
- Evaluation metrics

Many of these methods use a top-down, or *pachinko machine*, method for doing the classification. This method involves inducing a classifier for every (internal) node in the tree in order to distinguish between the child nodes of this node. The classification proceeds in a top down fashion starting at the root node, where each classification decision decides which path along the tree will be followed. The classification proceeds in this manner, until the document processing reaches the leaf nodes.

The following sections describe some notable efforts in the area of hierarchical text categorization. Additional details for these methods are summarized in Table 1.

### 8.2.3 NAÏVE BAYE'S CLASSIFIER

One of the first attempts in hierarchical text classification was by Sahami et al. [345]. The taxonomy chosen is a “toy” taxonomy that was 2 levels deep, and consisted of 10 nodes in total. The document collection chosen for experiments was Reuters-22173 news stories.

Author	Classifier	Taxonomy	Training Set	Performance
Sahami et al. [345]	Naïve Baye's	10	1000	0.81 (accuracy)
Srinivasan et al. [557]	Neural Networks	120	300000	0.30 (accuracy)
Liu et al. [392]	SVM	246279	792601	0.24 (Micro- $F_1$ )
Frank et al. [218]	Centroid based	4214	868836	0.55 (accuracy)
Cai et al. [91]	SVM	1172	14690	0.34 (accuracy)
Dumais et al. [164]	SVM	163	10000	0.5 (Micro- $F_1$ )

Table 8.1: Hierarchical Text Classification Approaches

They train a Naïve Baye's classifier for each internal node of the taxonomy, after performing feature space reduction using Information Theoretic measures. Subsequently, a top down hierarchical classification approach is used to classify test documents. They reported substantial improvements over the conventional flat classification approach.

#### 8.2.4 NEURAL NETWORKS CLASSIFIER

Srinivasan et al. [557] developed a Neural Networks based classifier building on the Hierarchical Mixture of Experts (HME) model [323]. The HME model is a supervised feedforward network that can be used for classification or regression. This model uses a divide and conquer principle to reduce the categorization problem into smaller sub-problems (one classification decision, on a reduced feature set, at each node). Using this approach they train an array of neural network classifiers.

The taxonomy used is MeSH (Medical Subject Headings) and the document collection is the set of medical literature records obtained from the MEDLINE collection <sup>3</sup>. They however only use the title and abstract of the article for training and testing purposes. The training-testing split is 30%-70%.

$\chi^2$ , odds ratio, and mutual information are used for feature selection. The metric used for evaluating the performance of the classification algorithm is the  $F_1$  measure.

<sup>3</sup>[http://www.nlm.nih.gov/databases/databases\\_medline.html](http://www.nlm.nih.gov/databases/databases_medline.html)

### 8.2.5 SEARCH BASED STRATEGY

Xue et al. [687] propose a two stage search based classification algorithm. In the first stage candidate categories for a test document are identified. This is done by comparing the document with the entire training set (using a  $k$ -nearest neighbors approach), and then selecting the nodes corresponding to the nearest  $k$  training documents as the possible candidate categories for this new document. In the next stage, a statistical-language model based classifier is developed to perform the hierarchical top down classification.

The dataset chosen for analysis is a subset of DMOZ. Incidentally, they do not perform any feature selection to reduce the feature space. They measure the Micro- $F_1$  measure at various levels of the category tree, and report 0.8 as the best obtained Micro- $F_1$  measure (at level 1), and 0.2 as the worst (at level 9).

### 8.2.6 COMPARATIVE ANALYSIS

Ceci et al. [110] present a comparative analysis of various types of classifiers used in hierarchical text categorization. They choose a centroid based classifier, Naïve Bayes, and SVM for experimentation and comparison. They also study and contrast various methods for building training sets for a node in a taxonomy (Figure 8.2). The first method (as shown in (a)) involves using the training documents of a node and all of its children as the positive training set for that node, and those of its siblings (and all of their children) as the negative training set. The second method (as shown in (b)) involves using only the documents corresponding to that node as the positive training set, and the documents for its sibling nodes (not their children) as the negative training set.

Ceci et al. draw several interesting conclusions from their experimentation. Based on the results of their baseline analysis (flat classification) they conclude that SVMs are the best performing classifiers across all datasets they tested. They also found that certain types of classifiers (centroid based ones, in particular) do not perform well while doing hierarchical classification.

### 8.2.7 SCALABILITY ANALYSIS

While several methods attempt to do hierarchical text categorization, not many report the computational costs for the methods developed. What is fairly obvious from the work so far, though, is that the computational cost is proportional to the number of categories, and increases to a very high level as the size of the taxonomy increases.

Liu et al. [392] report that it took 2 months for their classification experiments on the entire Yahoo! directory to finish while running on a cluster consisting of 10 powerful machines. Yang et al.[689] conducted a theoretical analysis on the scalability of SVM classifiers when applied to hierarchical classification, making several assumptions about the taxonomy size, training set size, etc., that may or may not be applicable to real world data.

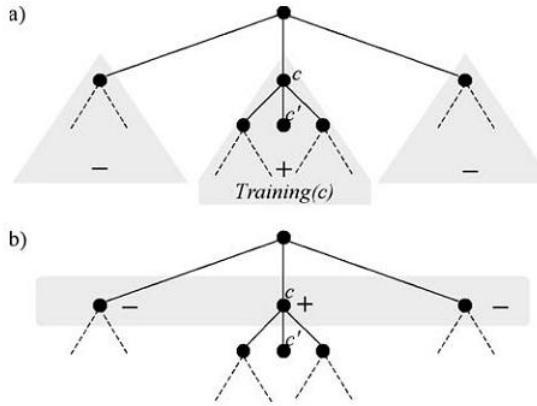


Figure 8.2: Positive and Negative Training Sets for a Node

### 8.3 5S FORMALISMS

Formal definitions of terms and concepts in the context of hierarchical multi-label text classification are presented below.

#### 8.3.1 STREAMS

Streams in the context of our work are no different from those defined elsewhere in the book. Streams consist of streams of text, video (static or dynamic), or audio material associated with an ETD.

ETDs also comprise structured streams as shown in Figure 8.3 (adapted from Gonçalves et.al.[254]), that model the structure inherent in ETDs.

#### 8.3.2 STRUCTURES

The taxonomy that is under use (DMOZ) comprises a structure. It is defined as a set  $(T, \leq_h)$  where  $T$  represents the set of nodes in the taxonomy and  $\leq_h$  is a partial ordering on the nodes of the taxonomy such that  $\forall p, q \in T, p \leq_h q$  implies that  $p$  is the ancestor of  $q$  in the taxonomy.

#### 8.3.3 SPACES

The document set  $D$  space is a collection of digital objects  $d_i$  (ETDs in our case). Each digital object by itself is a tuple such that  $d_i = (h, SM, ST, StructuredStream)$  where  $h \in H$  is a unique handle for an ETD,  $SM$  is the text/audio/video/image stream associated with an ETD,  $ST$  is a structural metadata specification, and Structured Streams are as presented above.

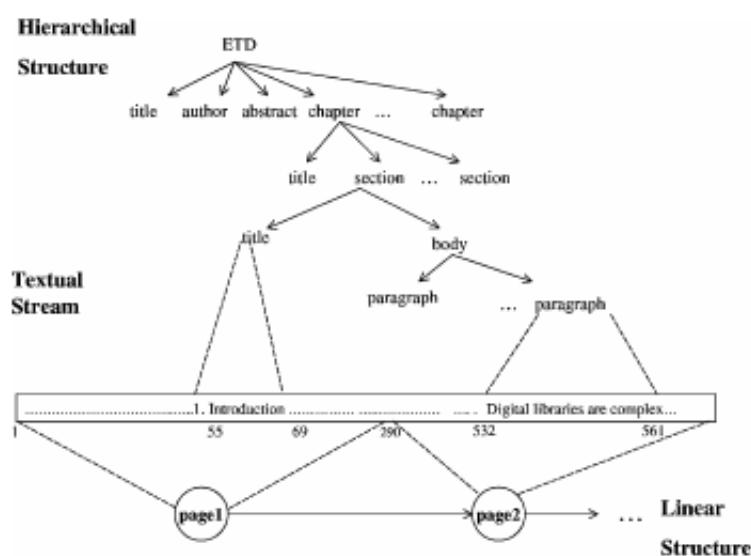


Figure 8.3: ETD Structured Stream

A collection  $C = do_1, do_2, \dots, do_k$  is a set of digital objects.

Let  $C$  be a collection with  $k$  handles in  $H$ . A metadata catalog  $DMC$  for  $C$  is a set of pairs  $(h, dm_1, \dots, dm_k)$ , where  $h \in H$  and the  $dm_i$  are descriptive metadata specifications.

While doing document classification, we represent documents as vectors in a vector space. A document is represented by a  $t$ -dimensional vector  $d_i = (w_{i1}, w_{i2}, \dots, w_{in})$ , where the term  $w_{in}$  represents relative weight (importance) of a word occurring in the document.

#### 8.3.4 SCENARIOS

Two important usage scenarios are relevant in the context of the Digital Library that we have built here.

In the first scenario, the users browse by topics (nodes) in the taxonomy to navigate to a node of interest. In the second scenario, the navigation could be by metadata fields (for example by department name, keywords, etc.).

Another possible usage scenario is browsing by both - nodes in the taxonomy as well as metadata fields. A user for example, may want to browse ETDs in Science (a node in the taxonomy) that contains a specific keyword of interest (a metadata field)

### 8.3.5 SOCIETIES

The DL caters to the needs of the scholarly community, particularly students. The tools developed herein also aid in automatic cataloging of ETDs and are hence of direct interest to librarians and catalogers.

### 8.3.6 FORMAL DEFINITION OF CLASSIFICATION

**Definition 8.1:** **Feature selection** is a scenario made up of transition events that begin from a state of having a *token feature sequence* and end with a state of having a *subset of token feature sequence*.

**Definition 8.2:** A **training set** is a streamed structure.

**Definition 8.3:** A **classifier** is a labeling function from a token to an entry in a classification scheme.

**Definition 8.4:** **Supervised learning** is a scenario made up of transition events from a state of having a *token feature sequence* and end with a state of having a *trained classifier*.

Supervised learning is a process that finds a labeling function to map tokens in a text stream to entries in a classification scheme.

**Definition 8.6:** **Text classification** is a service that takes in a stream, and produces a structured stream. This service usually consists of a set of scenarios such as *supervised learning* and *decoding*, each of which is composed of a sequence of scenarios, such as *feature selection*, and *training/decoding*. Prior to feature selection after tokenization, **stop word removal** and **stemming** may be performed.

### 8.3.7 HIERARCHICAL CLASSIFICATION

In light of the definitions presented in the preceding sections, the hierarchical classification problem can be formally defined as a function  $\mathcal{F} : D \rightarrow 2^T$  such that some quality metric  $m$  is maximized (or minimized). Additionally  $t \in \mathcal{F}(d) \Rightarrow \forall t' \leq_h t : t' \in \mathcal{F}(d)$ .

## 8.4 CASE STUDY: HIERARCHICAL CLASSIFICATION OF ETDs

ETD categorization in the setting described so far is a multi-step process. Our approach to ETD categorization is described in detail below.

### 8.4.1 BUILDING A TAXONOMY

The first step is to identify a suitable taxonomy for ETDs. While there exists at least one existing taxonomy for ETDs (the one developed by Proquest<sup>4</sup>), it was found to be

<sup>4</sup><http://www.proquest.com>

#### 8.4. CASE STUDY: HIERARCHICAL CLASSIFICATION OF ETDs 257

unsuitable for our purposes for various reasons. Firstly, the taxonomy is very general and not deep enough to cover very specialized categories within a domain. A taxonomy that is only 2 levels deep and has say Computer Engineering as the most specific category is unlikely to be of much help in browsing. Similarly a taxonomy that is say 10 levels deep and is very specific, is also unlikely to be of much help, since the users are unlikely to navigate to a node that is very deep in the tree (in this case, searching would be preferred).

The taxonomy that has been used for this work is the one provided by DMOZ<sup>5</sup>. DMOZ is often referred to as the yellow pages of the internet and has been extensively used for categorizing webpages and facilitating searching, and also browsing by topics. The DMOZ category tree by itself is very large, with in excess of 500,000 nodes. Therefore it is unusable by itself for ETD categorization.

For the purpose of this work though, we limit ourselves to categorizing only upto 2 levels deep in the DMOZ taxonomy. The top level of the taxonomy consists of only one “root” node, and the next level (children) consist of the following topics (after suitably pruning the taxonomy): Arts, Business, Computers, Health, Science, Society.

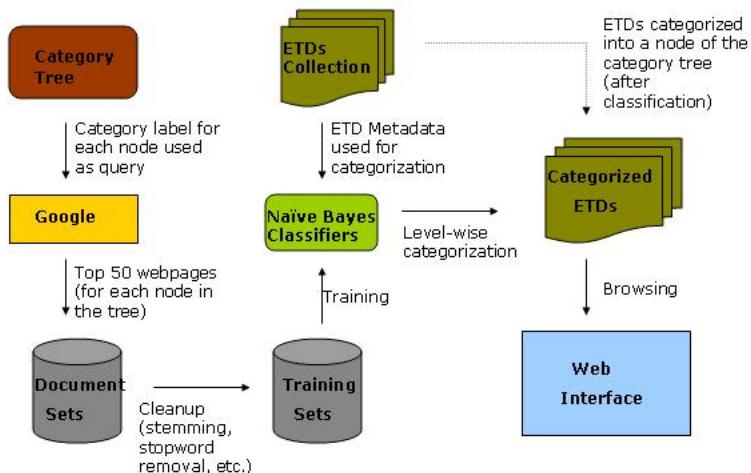


Figure 8.4: ETD Categorization Pipeline

<sup>5</sup><http://www.www.dmoz.org>

### 8.4.2 CRAWLING ETD METADATA

NDLTD's Union Catalog provides Dublin Core metadata for over 1 million ETDs. Much of this data was harvested and stored locally. Title, abstract, and keyword fields were combined to form a single string (more on this later in the section).

### 8.4.3 CATEGORIZING ETDs

The next step is to assign ETDs to their respective topic (or node) in the category tree. We explored various hierarchical categorization algorithms, and decided to use the top down supervised classification algorithm proposed by Koller et.al.[345]<sup>6</sup>

The detailed steps are in the following sections, and are shown in Figure 8.4.

#### **Building the training set**

Given the taxonomy, we first build a collection of training documents for each node. We build this document set by using the (enhanced) category label of the node as a query to a web search engine and retrieving the top 50 hits. We then crawl these webpages, and remove the HTML tags. Each document is subject to stopword removal and stemming, and once this is done, the stemmed words are used as features and a Naïve Bayes classifier is trained to distinguish between different child categories.

#### **Training the classifier**

Using the training set above, we build and train a Naïve Bayes classifier for the root node of the DMOZ category tree to distinguish between its children. Specifically, we train the classifier to distinguish between Arts, Business, Computers, Health, Science, and Society categories.

#### **Categorization**

Once the training has been done, the classifiers are used to map ETDs to their respective topic in the tree. As mentioned above, we make use of only the metadata information associated with each ETD, viz. title, subject, and description fields of Dublin Core, to categorize it suitably. The categorization is done in a level-wise manner. At every level in the category tree only 1 classification task is done. Since we are doing the categorization into a tree that is only two levels deep, we had to build only one Naïve Bayes classifier viz. for assigning ETDs into one of the 6 major areas as mentioned.

We selected ETDs from 8 different universities for topic identification. We have categorized them into the second level of the DMOZ category tree, and are working on doing categorization at the lower (more specific) levels in the category tree. Some results relating to this are presented in Figure 8.5.

<sup>6</sup>However since we are only classifying into 2 levels in the tree, our technique essentially reduces into a flat classification one

Name of the University	Total No. of ETDs	Category					
		Arts	Business	Computers	Health	Science	Society
MIT	29804	653	1847	6507	375	7141	555
Virginia Tech	11976	742	627	2665	1218	3317	340
Ohiolink	8020	1056	350	1267	1322	2887	345
Rice	6685	937	235	1181	145	2412	62
NCSU	5026	283	245	1419	512	2436	114
Texas A&M	4834	302	363	1363	566	2115	125
CalTech	4774	58	52	1392	29	3096	18
Georgia Tech	3582	32	133	1348	85	1233	23
<b>TOTAL</b>	<b>74701</b>	<b>4063</b>	<b>3852</b>	<b>17142</b>	<b>4252</b>	<b>24637</b>	<b>1582</b>

Figure 8.5: ETD Categorization for ETDs from 8 major US universities in Union Catalog

Training of the classifiers is done offline, and is quite efficient. Training on 300 documents (50 documents for each of the 6 categories) took less than 5 minutes, on a desktop computer with Dual Core Intel 2.80 GHz processors with 1 GB memory, and running Ubuntu Linux. Categorization is also very efficient; to categorize roughly 74,000 ETDs (Figure 8.5), took less than 30 minutes. Average precision and recall values were 0.9 and 0.88, respectively.

## 8.5 SUMMARY

In this chapter we have presented work on building a DL of ETDs categorized by topics to facilitate browsing and exploration. We also presented the 5S formalizations in this setting to aid in understanding of the DL system.

Hierarchical classification of documents remains an active area of research. We are currently extending our work on ETD classification by leveraging the full text of ETDs, in addition to metadata, for category identification. We are classifying the ETDs into a widely used and familiar taxonomy, Library of Congress Subject Classification (LCC). Ultimately we hope to make available a DL that contains all (freely available) ETDs from the Union Catalog organized by topics as specified by LCC.

## 8.6 EXERCISES AND PROJECTS

1. Assume you have 1 million ETDs and 3000 LCC categories, and enough training data to build 3000 classifiers for those categories. Assume you have a program that will split ETDs into chapters. Assume you have made runs to build classifiers for:

- (a) Each ETD using the metadata only
- (b) Each ETD using the full-text of the whole work
- (c) Each chapter of each ETD, using the full-text of the chapter

Please explain how you would use all this to aid those interested in discovering topically interesting ETDs, including methods and interface descriptions.

2. Familiarize yourself with LCC <sup>7</sup> and NDLTD Union Catalog's listing of ETDs by university <sup>8</sup>. Select 1 or more universities from the list (except Virginia Tech, NCSU, FSU, and LSU), and navigate to the main page with ETD listings of these universities. From there, get the list of departments that each have more than 50 ETDs listed in the ETD collection. For each such department, identify the *most specific* node(s) in the LCC that the ETDs from that department would be mapped into. For example, Virginia Tech Computer Science department's ETDs would be mapped into LCC category QA75.5-76.95. Your final results should contain a mapping of at least 5000 ETDs to LCC categories.

<sup>7</sup><http://www.loc.gov/catdir/cpsoc/lcco/>

<sup>8</sup><http://alcme.oclc.org/ndltd/servlet/OAIHandler?verb=ListSets>

## CHAPTER 9

# Content-based Image Retrieval

by Ricardo da Silva Torres, Nádia P. Kozievitch, Uma Murthy, and Alexandre X. Falcão

*Abstract:* In this chapter we exploit the 5S Framework to propose a formal description for Content-Based Image Retrieval, defining their fundamental concepts and relationships from a digital library (DL) perspective.

### 9.1 INTRODUCTION

Technological improvements in image acquisition and the decreasing cost of storage devices have supported the dissemination of large image collections, supported by efficient retrieval services. One of the most common approaches involves so-called *Content-Based Image Retrieval (CBIR) systems* [188, 487]. Basically, these systems try to retrieve images similar to a user-defined specification or pattern (e.g., shape sketch, image example). Their goal is to support image retrieval based on *content* properties (e.g., shape, color, or texture), usually encoded into *feature vectors*. One of the main advantages of CBIR is the possibility of an automatic retrieval process, avoiding the work of assigning keywords, which usually requires very laborious and time-consuming prior annotation of images.

Various Digital Libraries (DLs) support services based on image content [53, 697, 294, 221, 658, 660]. However, these systems are often designed and implemented without taking advantage of formal methods and frameworks. This chapter aims to extend the 5S (Streams, Structures, Spaces, Scenarios, and Societies) digital library formal framework [253] for describing services based on image content description. The main contribution of this chapter is the proposal of several constructs that extend the 5S framework to handle image content descriptions and related services. These constructs can aid understanding of content-based image retrieval concepts as they apply to DLs. They also can guide the design and implementation of new DL services based on image content.

We illustrate the proposed formalism by describing the interaction of two different services for image collections: image searching and image annotation. Image searching is supported by a content-based image search component (CBISC) [637], recently developed at State University of Campinas and Virginia Tech. The image annotation service will be illustrated through the description of the OntoSAIA (Ontology-based Semi-Automatic Image Annotation) tool [220].

This chapter describes the typical architecture of a CBIR system is discussed in Section 9.2. Related work and the basic concepts of the CBIR domain are presented in

## 262 9. CONTENT-BASED IMAGE RETRIEVAL

Section 9.3. The proposed extension to the 5S model is presented in Section 9.4. Next, the content-based image searching service is illustrated in a case study in Section 9.5. Some research challenges of the CBIR domain are introduced in Section 9.6.

## 9.2 CONTENT-BASED IMAGE RETRIEVAL

A typical *CBIR* solution requires the construction of **image descriptors**, which are characterized by: (i) an *extraction algorithm* to encode image features into *feature vectors*; and (ii) a *similarity measure* to compare two images based on the distance between their corresponding feature vectors. The similarity measure is a *matching function*, which gives the degree of similarity for a given pair of images represented by their feature vectors, often defined as a function of the distance (e.g., Euclidean), that is, the larger the distance value, the less similar the images.

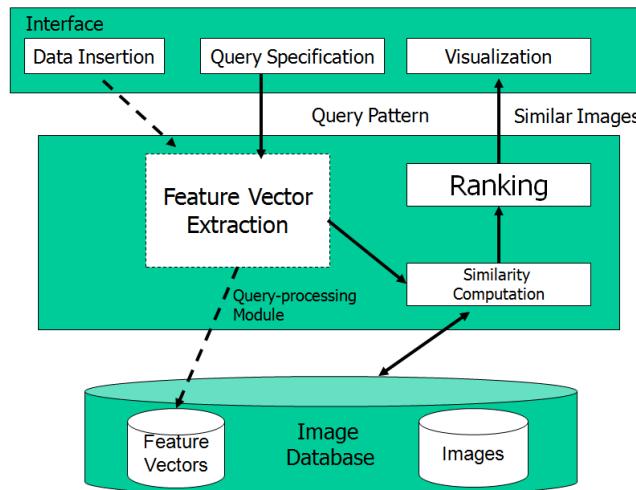


Figure 9.1: Typical CBIR system.

Figure 9.1 shows an overview of a content-based image retrieval system. The interface allows a user to specify a query by means of a query pattern (e.g., a query image) and to visualize the retrieved similar images. The query-processing module extracts a feature vector from a query pattern and applies a distance function (such as the Euclidean distance) to evaluate the similarity between the query image and the images. Next, it ranks the database images according to similarity and forwards the most similar images to the interface module. Note that database images are often indexed according to their feature vectors using structures to speed up retrieval and distance computation.

## 9.3 RELATED WORK

This section presents related work, discussing the main concepts of the CBIR domain.

### 9.3.1 IMAGE DESCRIPTORS

An image descriptor is a pair, *feature vector extraction function* and *distance function*, used for image indexation by similarity. The extracted feature vector subsumes the image properties and the distance function measures the dissimilarity between two images with respect to their properties. This section aims to present a brief overview of existing image descriptors.

#### Shape Descriptors

In pattern recognition and related areas, shape is an important characteristic to identify and distinguish objects [394, 693].

Shape descriptors are classified into *boundary-based* (or *contour-based*) and *region-based* methods [693]. This classification takes into account whether shape features are extracted from the contour only or from the whole shape region. These two classes, in turn, can be divided into *structural (local)* and *global* descriptors. This subdivision is based on whether the shape is represented as a whole or represented by segments/sections. Another possible classification categorizes shape description methods into *spatial* and *transform* domain techniques, depending on whether direct measurements of the shape are used or a transformation is applied [561]<sup>1</sup>.

Next, we present a brief overview of some shape descriptors. More details about existing shape representation techniques can be found in [394, 424, 693, 129].

**Moment Invariants:** For Moment Invariants, each object is represented by a 14-dimensional feature vector, including two sets of normalized Moment Invariants [296, 163], one from the object contour and another from its solid object silhouette. Again, the Euclidean distance is usually used to measure the similarity between different shapes as represented by their Moment Invariants.

**Curvature Scale Space (CSS)** [2, 437]: The CSS descriptor is used in the MPEG-7 standard [63] and represents a multiscale organization of the curvature zero-crossing points of a planar curve. In this sense, the dimension of its feature vectors varies for different contours, thus a special matching algorithm is necessary to compare two CSS descriptors (e.g., [139]).

**Beam Angle Statistics (BAS)** [25]: The BAS descriptor is based on the *beams* originated from a contour pixel. A beam is defined as the set of lines connecting a contour pixel to the rest of the pixels along the contour. At each contour pixel, the angle between a pair of lines is calculated, and the shape descriptor is defined by using the third-order statistics of all the beam angles in a set of neighborhoods. The similarity between two BAS

<sup>1</sup>Taxonomies of shape description techniques can be found in [561, 693, 129].

## 264 9. CONTENT-BASED IMAGE RETRIEVAL

moment functions is measured by an optimal correspondent subsequence (OCS) algorithm, as shown in [25].

**Tensor Scale Descriptor (TSD)** [433]: TSD is a shape descriptor based on the tensor scale concept [563]— a morphometric parameter yielding a unified representation of local structure thickness, orientation, and anisotropy. That is, at any image point, its tensor scale is represented by the largest ellipse (2D) centered at that point and within the same homogeneous region. TSD is obtained by extracting the tensor scale parameters for the original image and then computing the ellipse orientation histogram. TSDs are compared by using a correlation-based distance function.

**Contour Saliences (CS)** [138]: The CS computation uses the Image Foresting Transform [180] to compute the salience values of contour pixels and to locate salience points along the contour by exploiting the relation between a contour and its internal and external skeletons [385]. The contour salience descriptor consists of the salience values of salient pixels and their location along the contour, and on a heuristic matching algorithm as distance function.

**Segment Saliences (SS)** [138]: The segment salience descriptor is a variation of the contour salience descriptor which incorporates two improvements: the *salience values* of contour segments, in the place of salience values of isolated points, and another matching algorithm that replaces the heuristic matching by an optimum approach. The salience values along the contour are computed and the contour is divided into a predefined number  $s$  of segments of the same size. The internal and external influence areas of each segment are computed by summing up the influence areas of their corresponding pixels. In [138], SS is showed to present a better effectiveness than several other shape descriptors.

### Color Descriptors

Color property is one of the most widely used visual feature in content-based image retrieval (CBIR) systems. Research in this field can be grouped into three main subareas: (a) definition of adequate color space for a given target application, (b) proposal of appropriate extraction algorithms, and (c) study/evaluation of similarity measures.

Color information is represented as points in three-dimensional color spaces (such as RGB, HSV, YIQ,  $L^*u^*v^*$ ,  $L^*a^*b^*$  [151]). They allow discrimination between color stimuli and permit similarity judgment and identification [151]. Some of them are hardware-oriented (e.g., RGB, and CMY color space), as they were defined by taking into account properties of the devices used to reproduce colors. Others are user-inspired (e.g.,  $L^*u^*v^*$ ,  $L^*a^*b^*$ ) as they were defined to quantify color differences as perceived by humans.

Several color description techniques have been proposed [621, 611, 298, 501, 416]. They can be grouped into two classes based on whether or not they encode information related to the color spatial distribution.

Examples of descriptors that do not include spatial color distribution include Color Histogram and Color Moments. **Color Histogram** [621] is the most commonly used descriptor in image retrieval. The color histogram extraction algorithm can be divided into three steps: partition of the color space into cells, association of each cell to a histogram bin, and counting of the number of image pixels of each cell and storing this count in the corresponding histogram bin. This descriptor is invariant to translation and rotation. The similarity between two color histograms can be performed by computing the  $L_1$ ,  $L_2$ , or weighted Euclidean distances, as well as by computing their intersection [621].

Other example of descriptor that does not consider color spatial distribution are the so-called **Color Moments** [611]. Usually, the *mean* (first order), *variance* (second), and *skewness* (third) are used to form the feature vector. These moments are defined, respectively, as  $E_i = (1/N) \sum_{j=1}^N p_{ij}$ ,  $\sigma_i = \sqrt{(1/N) \sum_{j=1}^N (p_{ij} - E_i)^2}$ , and  $s_i = \sqrt[3]{(1/N) \sum_{j=1}^N (p_{ij} - E_i)^3}$ , where  $p_{ij}$  is the value of the  $i$ -th color component of the image pixel  $j$ , and  $N$  is the number of pixels in the image.

Examples of color descriptors that incorporate color spatial distribution include **Color Coherence Vector (CCV)** [501], **Border/Interior Pixel Classification (BIC)** [610], and **Color Correlogram** [298]. CCVs are created by computing, for each color, the total number of coherent ( $\alpha_i$ ) and incoherent pixels ( $\beta_i$ ). A pixel is considered coherent if it belongs to a largely uniformly-colored region. The CCV is defined as  $V_c = <(\alpha_1, \beta_1), (\alpha_2, \beta_2), \dots, (\alpha_N, \beta_N)>$ , where  $N$  is the number of colors. The color correlogram, in turn, encodes the spatial correlation of colors. It can be seen as a table  $\gamma$  indexed by color pairs. Given any pixel of color  $c_i$  in the image,  $\gamma_{c_i, c_j}^{(k)}$  gives the probability that a pixel at distance  $k$  away from the given pixel is of color  $c_j$ . The color correlogram is a table indexed by color pair, where the  $k$ -th entry for  $<i, j>$  specifies the probability of finding a pixel of color  $j$  at a distance  $k$  from a pixel of color  $i$  in the image. In the BIC approach, each image pixel is classified as a border or interior pixel, based on whether it is at the border of the image itself or if at least one of its 4-neighbors have a different color. In the following, two histograms are computed: one considering only border pixels and another for only interior pixels.

The MPEG-7 initiative [560], formally known as Multimedia Content Description Interface, focuses on the description of multimedia content, including content of various modalities like image, video, speech, graphics, and their combinations. One of the most important components of the MPEG-7 framework is the proposal of image descriptors. For the color property, MPEG-7 has defined a number of histogram descriptors, a dominant color descriptor, and a color layout descriptor [416].

### Texture Descriptors

There is no widely accepted definition of texture. However, this image property can be characterized by the existence of basic primitives, whose spatial distribution creates some

## 266 9. CONTENT-BASED IMAGE RETRIEVAL

visual patterns defined in terms of granularity, directionality, and repetitiveness. There exist different approaches to extract and represent textures. They can be classified into *space-based*, *frequency-based* models, and *texture signatures* [151]. Next, some of these approaches are described.

A co-occurrence matrix [277] is one of the most traditional techniques for encoding texture information. It describes spatial relationships among grey-levels in an image. A cell defined by the position  $(i, j)$  in this matrix registers the probability at which two pixels of gray levels  $i$  and  $j$  occur in two relative positions. A set of co-occurrence probabilities (such as, energy, entropy, contrast) has been proposed to characterize textured regions. Other example of space-based method includes the use of Auto-Regressive Models [277].

Frequency-based texture descriptors include, for instance, the Garbor wavelet coefficients [415]. An example of texture signatures can be found in the proposal of Tamura *et al.* [622]. This descriptor aims to characterize texture information in terms of contrast, coarseness, and directionality. The MPEG-7 initiative proposed three texture descriptors: texture browsing descriptor, homogeneous texture descriptor, and local edge histogram descriptor [416].

### 9.3.2 CBIR SYSTEMS

Several CBIR systems have been proposed recently. Even though a few of them became commercial products [188], many CBIR systems were proposed as research prototype, being developed in universities and research laboratories.

QBIC (*Query by Image Content*) [188], Photobook [511] – developed by the Massachusetts Institute of Technology (MIT), Chabot, Netra, and VisualSEEK [603] allow query by image content. Next, some of them are described.

Chabot [487] integrates image content retrieval based on color information with text-based queries. Its interface allows users to search and update the image database. This system does not include texture and shape descriptors.

The QBIC system was developed by IBM [188]. It uses color, shape, and texture to retrieve from image databases. Query specification follows the *query-by-example* paradigm. A user can sketch a shape, select colors, indicate color distributions, or pick pre-defined textures.

Ma *et al.* [405] describe a toolbox for browsing large database collections called *Netra*. This prototype uses color, texture, shape, and spatial location of image segmented regions to retrieve similar images from a database.

More recently, Cox *et al.* [516] present the PicHunter system. In this system, a Bayesian framework is used to model user needs during query formulation. With a different approach, [653] describes an image retrieval system based on regions of interest, that is, regions that contain relevant objects of a given image. Another region-based image retrieval system is the Blobworld [104]. In this system, pixels are clustered according

to their color and texture properties. These clusters are supposed to represent the image content.

A more complete description of existing CBIR systems can be found in [647].

### 9.3.3 INDEXING STRUCTURES

Not only does the effectiveness but also the efficiency (measured in terms of retrieving time) needs to be taken into account during the design of CBIR systems. Usually, fast searching strategies rely on the use of effective indexing schemes. However, as pointed out earlier, images usually are represented as points in high dimensional spaces. In this scenario, traditional indexing schemes (such as the approaches based on the *R-trees* [272]), which perform reasonably well for a small number of dimensions, have a poor performance. This phenomena is called the “curse of dimensionality”. One of the commonly approaches used for addressing this problem is applying dimension reduction techniques, such as Principal Component Analysis (PCA), and then using a traditional multidimensional indexing structure.

Another important research area includes the investigation of **Metric Access Methods** (MAMs). MAM is a class of access method (AM) that is used to manage large volumes of metric data allowing insertions, deletions and searches [640]. The definition of these indexing approaches relies on the use of a metric space. A metric space is a pair  $(O, d)$ , where  $O$  denotes the domain of a set of objects  $O = (O_1, O_2, \dots, O_n)$ , and  $d$  is a metric distance with the following properties: (i) symmetry ( $d(O_1, O_2) = d(O_2, O_1)$ ), (ii) positiveness ( $0 < d(O_1, O_2) < \infty$ ,  $O_1 \neq O_2$  and  $d(O_1, O_2) = 0$ ), and (iii) triangle inequality ( $d(O_1, O_3) \leq d(O_1, O_2) + d(O_2, O_3)$ ). Examples of MAMs include, among others, the M-tree [119] and the Slim-tree [640].

Further details about multidimensional indexing structures can be found in [235, 64].

### 9.3.4 EFFECTIVENESS MEASURES

Image descriptors vary with the application domain and expert requirements. Thus, in order to identify appropriate image descriptors (used in extraction and distance computation algorithms), experts must perform a set of experiments to evaluate them in terms of effectiveness for a given collection of images. Effectiveness evaluation is a very complex task, involving questions related to the definition of a collection of images, a set of query images, a set of relevant images for each query image, and adequate retrieval effectiveness measures.

The evaluation of image descriptors and CBIR systems usually adopts the *query-by-example (QBE)* [28] paradigm. This paradigm, in the image retrieval context, is based on providing an image as input, extracting its visual features (e.g., contour saliences), measuring the distance between the query image and the images stored in the image database and, finally, ranking the images in increasing order of their distance from the query image (similarity).

## 268 9. CONTENT-BASED IMAGE RETRIEVAL

Since each descriptor represents an image as a “point” in the corresponding metric space, its effectiveness will be higher as more separate the clusters of relevant images are in the metric space; and as more compact the clusters are in the metric space, higher will be the robustness of the image descriptor with respect to an increase in the number of classes. Therefore, a “good” effectiveness measure should capture the concept of *separability*, and perhaps the concept of *compactability* for sake of robustness. More formally, the compactability of a descriptor indicates its invariance to the object characteristics that belong to a same class, while the separability indicates its discriminatory ability among objects that belong to distinct classes [139]. While these concepts are commonly used to define validity measures in cluster analysis [166, 141], they seem to not have caught much attention in the literature of CBIR systems, where one of the most used effectiveness measures is *Precision × Recall* [37].

Precision vs. Recall ( $P \times R$ ) curve is the commonest evaluation measure used in CBIR domain. Precision is defined as the fraction of retrieved images which is relevant to a query. In contrast, recall measures the fraction of the relevant images which has been retrieved. A recall is a non-decreasing function of rank, while precision can be regarded as a function of recall rather than rank. In general, the curve closest to the top of the chart indicates the best performance.

The effectiveness in image retrieval was discussed with respect to the Precision × Recall measure in [138], where the multiscale separability [139] was proposed as a more appropriate effectiveness measure.

Examples of other effectiveness measures include the  $\theta \times$  recall curve [610], average precision [37], and average normalized modified retrieval rank (ANMRR) [416].

### 9.3.5 USER INTERACTION IN CBIR SYSTEMS

From the user’s perspective, CBIR systems offer more flexibility in specifying queries than those based on metadata. On the other hand, they present new challenges. The first is how to help users in the *query specification* process. Another problem is *information overload* – how to present the result to the user in a meaningful way. A third issue is that of providing users with tools to *interact* with the system in order to refine their query. This section presents a brief overview of existing approaches that address these problems.

#### Query Specification

Several querying mechanisms have been created to help users define their information need. Asladoglu *et al.* [28] presented a list of possible query strategies that can be employed in CBIR systems. This list includes, for example, *simple visual feature query*, *feature combination query*, *localized feature query*, *query by example*, *user-defined attribute query*, *object relationship query*, and *concept queries*. For instance, in the case of a feature combination

query, a user could ask the system to “*Retrieve images with blue color and stripped texture, where both properties have the same weight*”.

Another distinction is made based on whether the user is looking for a class of similar items to a given query pattern (“*category search*”) or is looking for a particular target item (“*target search*”) [696].

### Result Visualization

The most common result presentation technique is based on showing a 2D (two-dimensional) grid of thumbnail (miniature) image versions [188, 487]. The grid is organized according to the similarity of each returned image with the query pattern (e.g. from left to right, from top to bottom). It is a  $n \times m$  matrix, where position (1, 1) is occupied by a thumbnail of the query pattern, position (1, 2) by the one most similar to it, and so on. This helps browsing, allowing users to simply scan the grid image set as if they were reading a text [546]. This approach, however, displays retrieved images of different similarity degrees at the same physical distance from the image query: e.g., images (1, 2) and (2, 1) are displayed at the same physical distance from the query pattern, but the former is more similar to it than the latter. [46] and [46] try to improve this visual structure by studying zoom properties to enhance image browsing. Rodden *et al.* [546], in turn, investigate whether it benefits users to have sets of thumbnails arranged according to their similarity, so images that are alike are placed together. They describe experiments to examine whether similarity-based arrangements of the candidate images help in picture selection.

Other display approaches try to consider relative similarity not only between the query pattern and each retrieved image, but also among all retrieved images themselves [575, 607]. These initiatives have the drawback that visually similar images which are placed next to each other can sometimes appear to merge or overlap, making them less eye-catching than if they were separated [546].

Stan *et al.* [607] describe an exploration system for an image database, which deals with a tool for visualization of the database at different levels of details based on a multi-dimensional scaling technique. This visualization technique groups together perceptual similar images in a hierarchy of image clusters. Retrieved images can overlap. The overlap problem is also found in El Niño image database [575]. In this context, Tian *et al.* [631] propose a PCA (Principal Component Analysis)-based image browser which looks into an optimization strategy to adjust the position and size of images in order to minimize overlap (maximize visibility) while maintaining fidelity to the original positions which are indicative of mutual similarities.

Torres *et al.* present in [638] two visualization techniques based on Spiral and Concentric Rings to explore query results (see Figure 9.8). These visual structures are centered on keeping user focus on the query image and on the most similar retrieved images. These

## 270 9. CONTENT-BASED IMAGE RETRIEVAL

strategies improve traditional 2D grid presentation and avoid image overlaps, commonly found in CBIR systems.

### Relevance Feedback

Relevance feedback (RF) is a commonly accepted method to improve the effectiveness of retrieval systems interactively [406]. Basically, it is composed of three steps: (a) an initial search is made by the system for a user-supplied query pattern, returning a small number of images; (b) the user then indicates which of the retrieved images are useful (relevant); (c) finally, the system automatically reformulates the original query based upon user's relevance judgments. This process can continue to iterate until the user is satisfied. RF strategies help to alleviate the semantic gap problem, since it allows the CBIR system to learn user's image perceptions. RF strategies usually deal with small training samples (typically less than 20 per round of interaction), asymmetry in training sample (a few negative examples are usually fed back to the system), and real time requirement (RF algorithms should be fast enough to support real-time user interaction) [696]. Another important issue is concerned with the design and implementation of learning mechanisms. The commonest strategies use *weight-based learning approaches* [555], *genetic algorithms* [396], *Bayesian probabilistic methods* [516], and *Support Vector Machines* [634].

### 9.3.6 APPLICATIONS

The CBIR technology has been used in several applications such as fingerprint identification, biodiversity information systems, digital libraries, crime prevention, medicine, historical research, among others. Some of these applications are presented in this section.

#### Medical Applications

The use of CBIR can result in powerful services that can benefit biomedical information systems. Three large domains can instantly take advantage of CBIR techniques: teaching, research, and diagnostics [443]. From the teaching perspective, searching tools can be used to find important cases to present to students. Research also can be enhanced by using services combining image content information with different kinds of data. For example, scientists can use mining tools to discover unusual patterns among textual (e.g., treatments reports, and patient records) and image content information. Similarity queries based on image content descriptors can also help the diagnostic process. Clinicians usually use similar cases for case-based reasoning in their clinical decision-making process. In this sense, while textual data can be used to find images of interest, visual features can be used to retrieve relevant information for a clinical case (e.g., comments, related literature, HTML pages, etc.).

### Biodiversity Information Systems

Biologists gather many kinds of data for biodiversity studies, including spatial data, and images of living beings. Ideally, Biodiversity Information Systems (BIS) should help researchers to enhance or complete their knowledge and understanding about species and their habitats by combining textual, image content-based, and geographical queries. An example of such a query might start by providing an image as input (e.g., a photo of a fish) and then asking the system to “*Retrieve all database images containing fish whose fins are shaped like those of the fish in this photo*”. A combination of this query with textual and spatial predicates would consist of “*Show the drainages where the fish species with ‘large eyes’ coexists with fish whose fins are shaped like those of the fish in the photo*”. Examples of initiatives in this area include [636, 294].

### Digital Libraries

There are several digital libraries that support services based on image content [53, 697, 294, 221, 658, 660]. One example is the digital museum of butterflies [294], aimed at building a digital collection of Taiwanese butterflies. This digital library includes a module responsible for content-based image retrieval based on color, texture, and patterns. In a different image context, Zhu *et al.* [697] present a content-based image retrieval digital library that supports geographical image retrieval. The system manages air photos which can be retrieved through texture descriptors. Place names associated with retrieved images can be displayed by cross-referencing with a Geographical Name Information System (GNIS) gazetteer. In this same domain, Bergman *et al.* [53] describe an architecture for storage and retrieval of satellite images and video data from a collection of heterogeneous archives.

Several have worked to formalize content-based image retrieval systems [639, 31]. However, these formalisms typically describe these kinds of services under the database perspective (in general, based on the relational or object-relational models). To the best of our knowledge this chapter constitutes the first formal attempt to describe content-based image retrieval services by using digital library concepts. One benefit is that the 5S framework is generic enough to formalize these services without relying on implementation decisions.

Another important initiative for the digital library domain is related to the proposal of the Content-Based Image Search Component (CBISC) [636]. CBISC is a recently developed component that provides an easy-to-install content-based image search engine. It can be readily tailored for a particular collection by a domain expert, who carries out a clearly defined set of pilot experiments. It supports the use of different types of vector-based image descriptors (metric and non-metric; color, texture, and shape descriptors; with different data structures to represent feature vectors), which can be chosen based on the pilot experiments, and then easily combined to yield improved effectiveness. CBISC is an OAI-like search component which aims at supporting queries on image content. As in the OAI pro-

## 272 9. CONTENT-BASED IMAGE RETRIEVAL

tocol [476], queries are submitted via HTTP requests. Two special requests (“verbs”) are supported by this image search component: **ListDescriptors**, used to retrieve the list of image descriptors supported by CBISC; and **GetImages**, used to retrieve a set of images by taking into account their contents.

### 9.4 FORMALIZATION

Figure 9.2 presents the proposed concepts based on the 5S framework to handle image content descriptions and related digital library services. These concepts are precisely defined below.

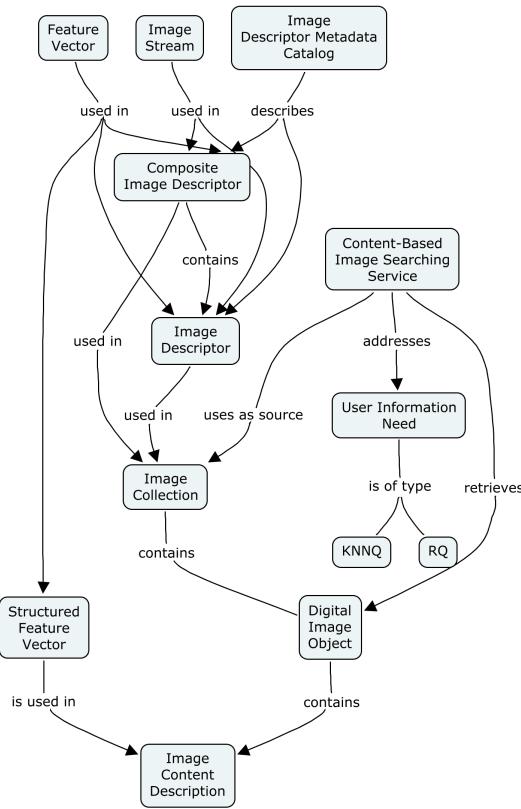


Figure 9.2: 5S extensions to support content-based image description and related services.

Some of these concepts were introduced in [637]. In this paper, we extend them by taking into account digital library aspects.

**Definition 9.1** An **image stream** (or simply **image**)  $\hat{I}$  is a pair  $(D_I, \vec{I})$ , where:

- $D_I$  is a finite set of *pixels* (points in  $\mathbb{N}^2$ , that is,  $D_I \subset \mathbb{N}^2$ ), and
- $\vec{I}: D_I \rightarrow D'$  is a function that assigns each pixel  $p$  in  $D_I$  to a vector  $\vec{I}(p)$  of values in some arbitrary space  $D'$  (for example,  $D' = \mathbb{R}^3$  when a color in the RGB system is assigned to a pixel).

**Definition 9.2** A **feature vector**  $\vec{fv}_{\hat{I}}$  of an image  $\hat{I}$  is a point in  $\mathbb{R}^n$  space:  $\vec{fv}_{\hat{I}} = (fv_1, fv_2, \dots, fv_n)$ , where  $n$  is the dimension of the vector.

Examples of possible feature vectors are a color histogram [621], a multiscale fractal curve [139], and a set of Fourier coefficients [512]. They essentially encode image properties, such as color, shape, and texture. Note that different types of feature vectors may require different similarity functions.

**Definition 9.3** Given a structure  $(G, L, \mathcal{F})$ ,  $G = (V, E)$  and a feature vector  $\vec{fv}_{\hat{I}}$ , a **StructuredFeatureVector** is a function  $V \rightarrow \mathbb{R}^n$  that associates each node  $v_k \in V$  with  $fv_i \in \vec{fv}_{\hat{I}}$ .

Figure 9.3 presents an example of the use of a **StructuredFeatureVector** function. In this case, an XML structure (structural metadata specification) is mapped to a feature vector obtained by applying the image descriptor *Contour Multiscale Fractal Dimension* [138] to the image stream defined by the file “fish0.pgm”.

```
<?xml version='1.0' encoding="UTF-8"?>
<feature_vector:Feature_Vector xmlns:feature_vector="http://feathers.dlib.vt.edu/~itorres/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://feathers.dlib.vt.edu/~itorres/feature_vector.xsd">
  <feature_vector:ImageName>fish0.pgm</feature_vector:ImageName>
  <feature_vector:DescriptorName>ContourMSFractalDimension</feature_vector:DescriptorName>
  <feature_vector:Type> I </feature_vector:Type>
  <feature_vector:Curve>
    <feature_vector:Nelements> 25 </feature_vector:Nelements>
    <feature_vector:CurveIDs>
      <feature_vector:Curve>
        <feature_vector:value> 0.95105259594482394192 </feature_vector:value>
        <feature_vector:value> 0.98551214588154611995 </feature_vector:value>
        <feature_vector:value> 1.00415492765507829986 </feature_vector:value>
        <feature_vector:value> 1.00931032237937512441 </feature_vector:value>
        <feature_vector:value> 1.00583781572741104426 </feature_vector:value>
        ...
        <feature_vector:value> 0.93810555611087775851 </feature_vector:value>
        <feature_vector:value> 0.87275204902189629230 </feature_vector:value>
        <feature_vector:value> 0.81066432563100665476 </feature_vector:value>
        <feature_vector:value> 0.7522463059381879515 </feature_vector:value>
      </feature_vector:Curve>
    </feature_vector:CurveIDs>
  </feature_vector:Curve>
</feature_vector:Feature_Vector>
```

Figure 9.3: Example of a structured feature vector.

**Definition 9.4** A **simple image content descriptor** (briefly, **image descriptor**)  $D$  is defined as a tuple  $(h_{desc}, \epsilon_D, \delta_D)$ , where:

274 9. CONTENT-BASED IMAGE RETRIEVAL

- $h_{desc} \in H$ , where  $H$  is a set of universally unique handles (labels);
- $\epsilon_D : \{\hat{I}\} \rightarrow \mathbb{R}^n$  is a function, which extracts a *feature vector*  $\vec{fv}_{\hat{I}}$  from an *image*  $\hat{I}$ .
- $\delta_D : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$  is a *similarity function* (e.g., based on a distance metric) that computes the similarity between two images as a function of the distance between their corresponding *feature vectors*.

Figure 9.4(b) illustrates the use of a simple descriptor  $D$  to compute the similarity between two images  $\hat{I}_A$  and  $\hat{I}_B$ . First, the extraction algorithm  $\epsilon_D$  is used to compute the feature vectors  $\vec{fv}_{\hat{I}_A}$  and  $\vec{fv}_{\hat{I}_B}$  associated with the images. Next, the similarity function  $\delta_D$  is used to determine the similarity value  $d$  between the images.

**Definition 9.5** A **composite image descriptor**  $\hat{D}$  is a tuple  $(h_{desc}, \mathcal{D}, \delta_{\mathcal{D}})$  (see Figure 9.4(b)), where:

- $h_{desc} \in H$ , where  $H$  is a set of universally unique handles (labels);
- $\mathcal{D} = \{D_1, D_2, \dots, D_k\}$  is a set of  $k$  pre-defined simple image descriptors;
- $\delta_{\mathcal{D}}$  is a similarity function which combines the similarity values obtained from each descriptor  $D_i \in \mathcal{D}$ ,  $i = 1, 2, \dots, k$ .

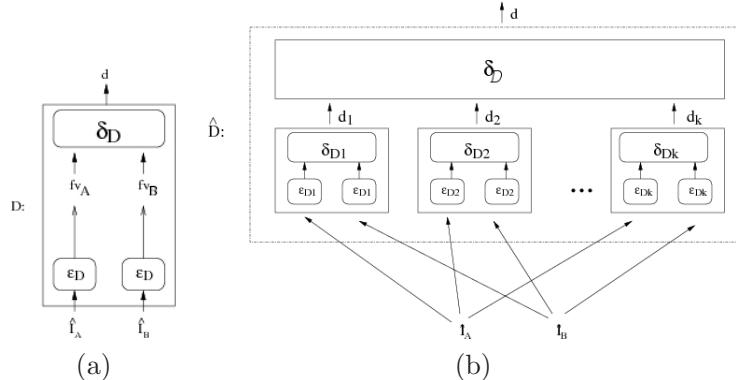


Figure 9.4: (a) The use of a simple descriptor  $D$  for computing the similarity between images. (b) Composite image descriptor.

**Definition 9.6** An **image content description**  $ICD$  is a tuple  $(FV, ST_{FVs}, Structured_{FVs})$ , where

- $FV = \{\vec{fv}_1, \vec{fv}_2, \dots, \vec{fv}_k\}$  is a set of feature vectors;
- $ST_{FVs} = \{stfv_1, stfv_2, \dots, stfv_m\}$  is a set of structural metadata specifications;
- $Structured_{FVs} = \{strfv_1, strfv_2, \dots, strfv_m\}$  is a set of StructuredFeatureVector functions defined from the *feature vectors* in the  $FV$  set (the first component) of the image content description and from the structures in the  $ST_{FVs}$  set (the second component).

**Definition 9.7** An **image digital object**  $ido$  is a digital object with the following extensions and constraints:

- $ido$  is a *digital object*  $= (h, SM, ST, StrStreams, ICD, StrICDStreams)$ , where
  1.  $h \in H$ , where  $H$  is a set of universally unique handles (labels);
  2.  $SM_{sd} = \{sm_{sd}[i, j]\} \in SM$ , where  $sm_{sd}[i, j] = \langle a_i, \dots, a_j \rangle$ ,  $0 \leq i \leq j \leq n$ .  $sm_{sd}[i, j]$  refers to substreams (regions) of an image stream.
  3.  $ST = \{st_1, st_2, \dots, st_m\}$  is a set of structural metadata specifications;
  4.  $StrStreams = \{stD_1, stD_2, \dots, stD_m\}$  is a set of StructuredStream functions defined from the image substreams in the  $SM$  set (the second component) of the digital object and from the structures in the  $ST$  set (the third component);
  5.  $ICD$  is an *image content description*;
  6.  $StrICDStreams = \{stimgD_1, stimgD_2, \dots, stimgD_m\}$  is a set of StructuredStream functions defined from the *image stream* in the  $SM$  set (the second component) of the image digital object and from the structures in the  $ST_{FVs} \in ICD(2)$  set.

Figure 9.5 illustrates the relations among the concepts used to define an image digital object.

The definition of  $StrICDStreams$  allows associating feature vectors to parts (objects, regions) of image streams.

**Definition 9.8** An **image collection**  $ImgC$  is a tuple  $(C, S_{imgdesc}, FV_{imgdesc})$ , where  $C$  is a collection (see Def. 17 in Chapter 1,  $S_{imgdesc}$  is a set of image descriptors, and  $FV_{desc}$  is a function  $FV_{desc} : \{C \times S_{imgdesc}\} \rightarrow ICD(1)$ , where  $ICD$  is  $ido(5)$  and  $ido \in C$ .

Function  $FV_{desc}$  defines how a feature vector was obtained, given an image digital object  $ido \in C$  and an image descriptor  $\hat{D} \in S_{imgdesc}$ . That is illustrated in Figure 9.6.

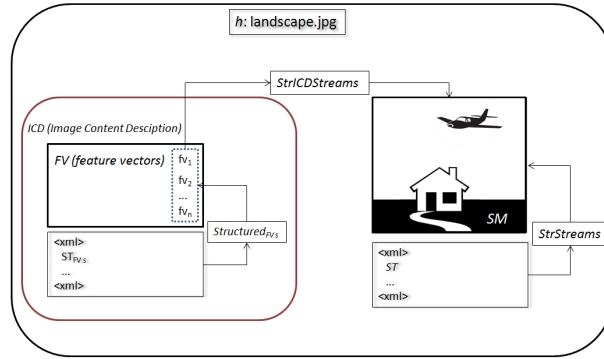


Figure 9.5: Image digital object elements.

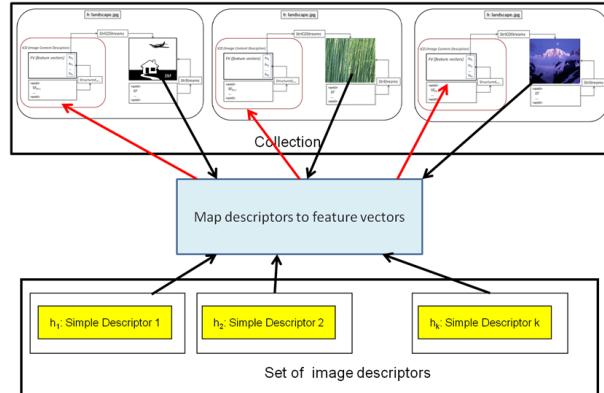


Figure 9.6: Use of a descriptor to extract feature vectors.

**Definition 9.9** Let  $S_{imgdesc}$  be a set of image descriptors with  $k$  handles in  $H$ . An **image descriptor metadata catalog**  $DM_{S_{imgdesc}}$  for  $S_{imgdesc}$  is a set of pairs  $\{(h, \{dmdesc_1, \dots, dmdesc_{k_h}\})\}$ , where  $h \in H$  and the  $dmdesc_i$  are descriptive metadata specifications for image descriptors.

Descriptive metadata specifications of descriptors could include, for example, data about the author (who implemented the extraction and similarity functions), implementation date, and related publication(s).

Recall that, in general, a metadata catalog is used to assign descriptive metadata specifications to image digital objects (see Def. 18 in Chapter 1).

**Definition 9.10** A conceptual representation for user information need is materialized into a query specification. A **query specification**  $Q$  is a tuple  $Q = \{(H_q, Contents_q, P_q)\}$ , where

$H_q = ((V_q, E_q), L_q, \mathcal{F}_q)$  is a structure (i.e., a directed graph with vertices  $V_q$  and edges  $E_q$ , along with labels  $L_q$  and labeling function  $\mathcal{F}_q$  on the graph; see Def. 2 in Chapter 1 for details),  $Contents_q$  includes 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 Def. 21 of Chapter 1 to define a searching service. The formal definition for conceptual representations for user information needs appears as Def. 2.1 in Chapter 2.

An example of a query specification is:  $q = (H_q, Contents_q, P_q) \in Q$ . For example:  $q$  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. 9.7(a). Thus,  $q = ((V_q, E_q), L_q, \mathcal{F}_q), Contents_q, P_q$ , where  $V_q = v_1, v_2, v_3, v_4, v_5$ ;  $E_q = e_1, e_2, e_3, e_4, e_5$ ;  $L_q = 'fire', 'earth', 'metal', 'water', 'wood', 'produce'$ ;  $\mathcal{F}_q : V_q \cup E_q \rightarrow L_q$ ,  $Contents_q$  is the stream of the five spatially related sub-images with their location information; and  $P_q : V_q \rightarrow Contents_q$  (see Fig. 9.7(b)).

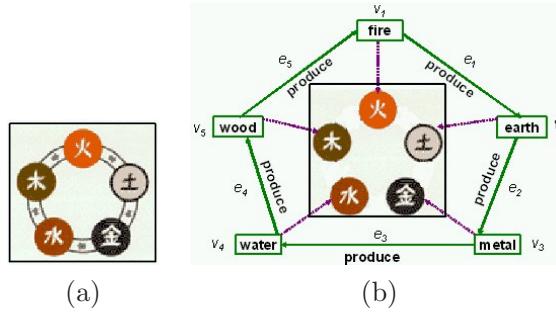


Figure 9.7: (a)  $q$  is an image of (b) 5 spatially related sub-images.

Usually, two kinds of queries are supported by CBIR systems [119]. In a *K-nearest neighbor query (KNNQ)*, the user specifies the number  $k$  of images to be retrieved that are closest to the query pattern. In a *range query (RQ)*, the user defines a search radius  $r$  and wants to retrieve all database images whose distance to the query pattern is less than  $r$ . In this case, both the specification of  $k$  in the KNNQ and the specification of  $r$  needs to be incorporated into  $Q$ .

**Definition 9.11** A query specification  $q \in Q$  is a **K-nearest neighbor query (KNNQ) information need** if there exists  $v \in V_q$ , a real number  $k \in Contents_q$ , and  $P_q(v) = k$ .

**Definition 9.12** A query specification  $q \in Q$  is a **range query (RQ) information need** if there exists  $v \in V_q$ , a real number  $r \in Contents_q$ , and  $P_q(v) = r$ .

## 278 9. CONTENT-BASED IMAGE RETRIEVAL

**Definition 9.13** Let  $V_{Spa}$  be a vector space (see Def. 13 in Appendix A) and  $Base$  be a set of basis vectors in  $V_{Spa}$ . 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.

A **visualization operation**  $OP_{viz}$  is a set of functions  $OP_{viz} = \{VisualMap_1, VisualMap_2, VisualMap_3\}$ , where  $VisualMap_1 : 2^C \rightarrow V_{Spa}$  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 type of visual mark; and  $VisualMap_3 : Base \rightarrow VisualMP$  associates a basis vector with a visual property of a visual mark.

Fig. 9.8 shows two examples of the use of  $OP_{viz}$  to visualize results in a shape-based image retrieval system [638]. Each of the returned images is mapped to a vector in a vector space  $V_{Spa}$  by function  $VisualMap_1$ .  $VisualMap_2$  maps returned images to thumbprints. In Figure 9.8(a), a function  $VisualMap'_3$  is used to present the most similar images. This function places the query image in the center, and fills a spiral line with the retrieved images at regular distances, according to their similarity to the query image [638]. In Figure 9.8(b) a function  $VisualMap''_3$  is used to present the most similar images in concentric rings. In this case, the rings are filled from the innermost ring to the outermost one, according to the image ranking [638].

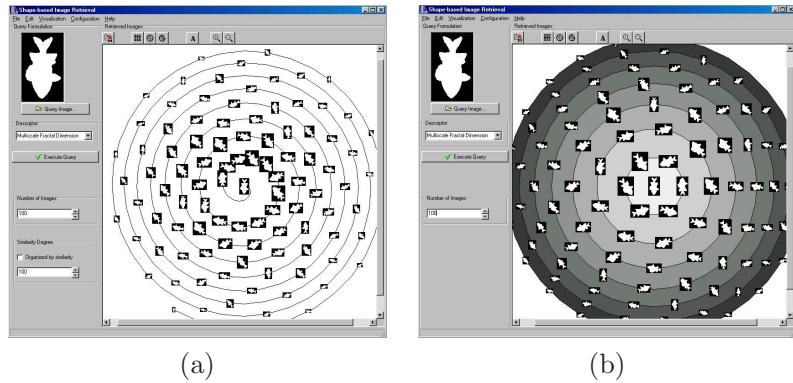


Figure 9.8: (a) Spiral approach. (b) Concentric rings approach.

**Definition 9.14 A content-based image searching service** is a set of searching scenarios  $\{sc_1, sc_2, \dots, sc_t\}$ , where each scenario  $sc_i$  is a sequence of events, and each event  $e_i$  is associated with the  $OP_s$  function defined as follows:

$OP_s : (Q \times C) \times Sim_s \rightarrow 2^{Contents}$ , where  $Sim_s = OP_q(q, ido) | q \in Q, ido \in C$ , and where  $OP_q : Q \times C \rightarrow \mathbb{R}$  is a matching function that associates a real number with  $q \in Q$  and a digital object  $ido \in C$ . The computation of  $OP_q$  relies

on the use of appropriate image descriptors (e.g., their extraction and distance computation algorithms) defined in the image collection  $ImgC$ .

The range of function  $OP_s$  is the *Contents* associated with collection  $ImgC$ . While a matching function was defined in Def. 21 in Chapter 1, the retrieved results were not defined there. We consider the retrieved results as (a subset of) the *Contents*.

## 9.5 CASE STUDY

In this section we exemplify how the 5S extensions for complex objects and content-based image retrieval can be explored to define the complex image object concepts in the CTRnet project. The services and digital objects of these tools are not unlike those of a digital library (DL) with extended functionality (such as annotations, and multimodal search).

The Crisis, Tragedy, and Recovery Network (CTRnet) [350, 688] objectives include to develop better approaches toward making technology useful for archiving information about such events, and to support analysis of rescue, relief, and recovery, from a digital library perspective. CTRnet has several modules, including crawling, filtering, a Facebook application, user visualization, metadata search, and Content-Based Image Retrieval. The CBIR module builds upon the EVA tool for evaluating image descriptors for content-based image retrieval. Eva integrates the most common stages of an image retrieval process and provides functionalities to facilitate the comparison of image descriptors in the context of content-based image retrieval.

CTRnet images can be evaluated by several descriptors, creating more components. Also, a clear description of ICO concepts can be used as a basis to integrate the CBIR module with other CTRnet modules. We defined that the complex image object would be composed of the image, the feature vectors, and similarity distances from each descriptor. Figure 9.9 shows the image ranking for Cathedral\_P\_A\_P.jpg, using two different descriptors: BIC [610] and SASI [5].

The ICO presented in Figure 9.9 can be defined by the structure  $ico = (h, SCDO = DO \cup SM, S)$ , where:

- $h$  is a unique handle that identifies ico;
- $DO = \{do_1, do_{21}, do_{22}, do_{31}, do_{32}\}$ , where  $do_1$  is an **image**,  $k = 2$  (BIC and SASI descriptors),  $do_{21}$  is a **feature vector digital object** using the BIC descriptor,  $do_{22}$  is a **feature vector digital object** using the SASI descriptor,  $do_{31}$  is a **StructuredFeatureVector** using the BIC descriptor, and  $do_{32}$  is a **StructuredFeatureVector** using the SASI descriptor;
- $SM = \{sm_1, sm_2, \dots, sm_n\}$  is a set of streams;
- $S$  is an XML structure that identifies how  $do_1, do_{21}, do_{22}, do_{31}$ , and  $do_{32}$  are composed.

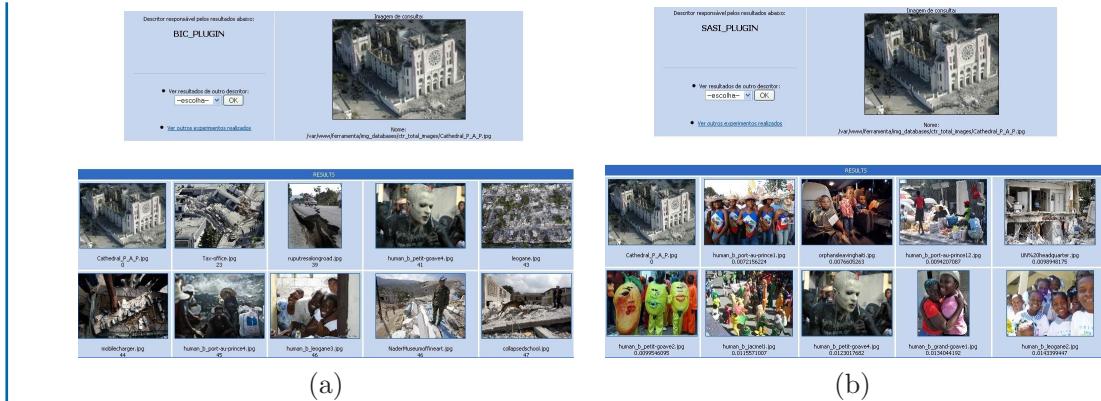


Figure 9.9: Image ranking in CTRnet using BIC (a) and SASI (b) descriptors.

The **complex image object collection** has 111 ICOs and two descriptors. Function  $FV_{desc1}$  (BIC) is based on the color descriptor. Function  $FV_{desc2}$  (SASI) is based on second order statistics of clique autocorrelation coefficients, which are the autocorrelation coefficients over a set of directional moving windows.

In summary, the CTRnet case study explored ICOs for aggregating information related to different descriptors, resulting from the CBIR process. These definitions can be used as a basis to aggregate information and define how these concepts relate to other CTRnet modules (such as metadata, and crawling).

## 9.6 RESEARCH CHALLENGES

The implementation of CBIR systems raises several research challenges, such as:

- Formalisms are needed not only to describe image content descriptions but also advanced services (e.g., multi-modal searches, content-based image annotation). This formalism can guide the design and implementation of new applications based on image content.
- Not many techniques are available to deal with the semantic gap presented in images and their textual descriptions. New tools for marking/annotating images (and their regions) need to be developed. Better semantically enriched descriptions can be created by taking advantage of ontologies [8, 220]. Another possible investigation area would be to incorporate classification strategies into the image retrieval process. The idea is to apply image retrieval and then classify the resulting images to change their order. In this case, the classifier works as an automatic approach for relevance feedback.

- Need for tools that automatically extract semantic features from images: extract high-level concepts contained in multimedia data.
- Development of new data fusion algorithms to support text-based and content-based retrieval combining information of different heterogeneous formats.
- Finding new connections, and mining patterns. Text mining techniques might be combined with visual-based descriptions.
- New user interfaces for annotating, browsing, and searching based on image content need to be investigated. Research in this area will require usability studies with practitioners.

## 9.7 SUMMARY

In this chapter we address the formal definitions and descriptions for Content-Based Image Retrieval. The proposed extensions for digital library functionality take advantage of formalization to understand clearly and unambiguously the characteristics, structure, and behaviour of the main concepts related to image content.

Later these definitions are explored in a case study, to exemplify how the CO and CBIR concepts can be explored to define the complex image object. Our contribution relies on (i) the formalization of content-based image retrieval; and (ii) the discussion of how to combine them to handle complex image objects in applications. The set of definitions also may impact future development efforts of a wide range of digital library experts since it can guide the design and implementation of new digital library services based on image content.

A straightforward benefit of this work is the use of these definitions to create applications, like those proposed in [251, 698, 362], or the formalization of more complex services that can be created by using the proposed constructs.

## 9.8 EXERCISES AND PROJECTS

1. How might CBIR be applied so teachers with a computer and connected camera can be reminded of the names of students in their class?
2. Consider the two colourful images (Image A and Image B) showed below, represented in the RGB color space. Suppose that the intensity values of each pixel in all bands (R, G, and B) are the same. Furthermore, each  $(R, G, B)$  triplet is represented by a single intensity value. For example the triplet  $(R, G, B) = (2, 2, 2)$  is represented by the intensity value 2.

Suppose also that the colour space was quantized in five colors with intensity values 0, 1, 2, 3, and 4.

0	0	1	2	4
0	0	3	2	4
3	3	1	0	1
3	1	4	2	1
3	4	4	2	2

Image A.

4	4	2	0	1
4	4	2	0	1
4	3	3	1	1
3	3	3	0	1
2	2	2	0	0

Image B.

- Compute the  $L_1$  between the *Color Histograms* (5 bins) of the two images.
  - By considering both the feature vector extraction function and the distance function defined of the descriptor *Color Coherence Vector – CCV* [501], compute the distance  $\delta_{CCV}(A, B)$  between the two images.
3. Consider the existence of two classes (*class 1* and *class 2*) composed by five images each. Consider the existence of three different descriptors (*descriptor 1*, *descriptor 2*, and *descriptor 3*), whose feature vector extraction functions extract feature vectors belonging to the  $\mathbb{R}^2$  space. Table 9.1 shows the coordinate of each image of each class, considering the three descriptors.

Classes	Descriptor 1
class 1	$\{(1.50, 2.50), (1.50, 2.00), (2.00, 2.00), (1.00, 2.00), (1.50, 1.50)\}$
class 2	$\{(1.00, 1.00), (1.00, 2.00), (1.00, 3.00), (1.00, 4.00), (1.00, 5.00)\}$
Classes	Descriptor 2
class 1	$\{(2.00, 1.00), (2.00, 2.00), (2.00, 3.00), (2.00, 4.00), (2.00, 5.00)\}$
class 2	$\{(1.40, 1.40), (1.60, 1.40), (1.60, 1.20), (1.40, 1.20), (1.50, 1.30)\}$
Classes	Descriptor 3
class 1	$\{(1.50, 2.50), (1.50, 2.00), (1.75, 2.25), (1.25, 2.00), (1.50, 1.50)\}$
class 2	$\{(1.50, 5.50), (1.25, 5.00), (1.50, 5.00), (1.15, 5.00), (1.50, 4.50)\}$

**Table 9.1:** Coordinates of each image of classes classes 1 and 2 for three different descriptors.

Figures 9.10(a), 9.10(b), and 9.10(c) show classes 1 and 2 in the  $\mathbb{R}^2$  space for descriptors 1, 2, and 3, respectively.

- Construct the average *Precision*  $\times$  *Recall* curves for the three descriptors.
  - Which descriptor is the best one in terms of effectiveness?
4. Extend the 5S framework to define image search services with relevance feedback.
5. Consider the SuperIDR tool available at <http://scholar.lib.vt.edu/theses/available/etd-04142011-175752/>.

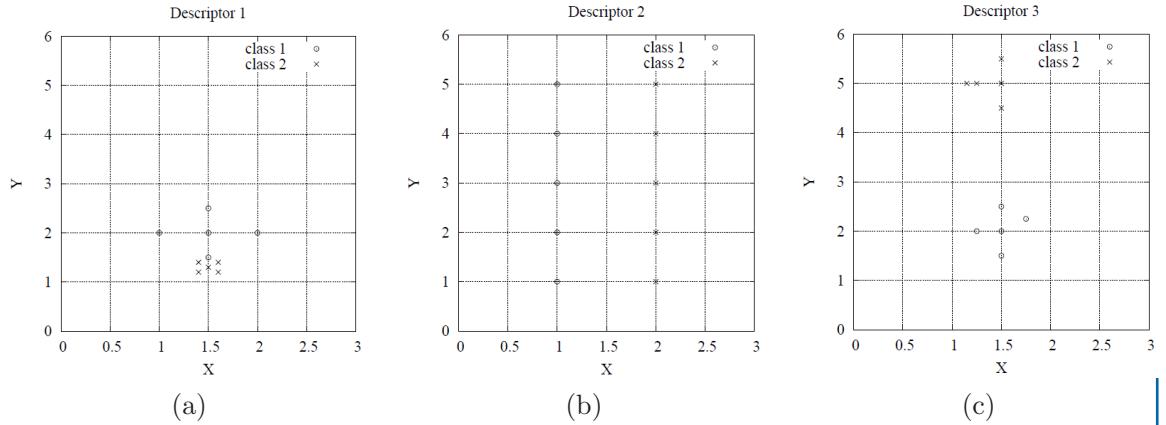


Figure 9.10: (a) Descriptor 1. (b) Descriptor 2. (c) Descriptor 3.

- Formalize the image searching services available in the SuperIDR tool, using the 5S framework.
- Implement a new descriptor to support image search services in the tool.
- Compare the performance (in terms of efficiency and effectiveness) of the implemented descriptor with the one available at the SuperIDR tool.

# Social Networks in Digital Libraries

by Monika Akbar

*Abstract:* Online communities and social networks in Web are growing rapidly, allowing users to connect, share, and learn in a collaborative environment. While the primary goal of digital libraries was to store and allow access to digital objects, now we need more than searching and browsing capabilities. We need to connect and communicate with other users in the same information space. Further, we should see, and possibly follow, their footsteps, to locate useful objects. In this chapter we describe how log files can be used to build behavioral networks within digital libraries. These networks can allow us to detect communities, interests, and trends among the users in a digital library. We also present a case study for an educational digital library. While this chapter emphasizes educational digital libraries, the methods discussed here have the potential to be useful in other digital libraries as well.

## 10.1 INTRODUCTION

The ever evolving world of Web technologies is forcing us to look closely at areas of digital libraries that relates users and digital artifacts in different ways. Just like real-world libraries, a digital library has some target audience, i.e., one or more societies connected with it, with people playing varying roles. As a whole the audience may have different needs and requirements than those of an individual user. Interactions and communications among the group can lead to an online community. Researchers have pointed out different aspects of establishing an online community in a digital library [71, 419, 301]. Additional research has focused on design issues [22, 271], studying and analyzing the overall architecture [149, 635], and identifying success factors [378, 391].

Success in a virtual community is largely dependent on the system's ability to provide suitable services, and on the performance of these services (e.g., fast, efficient search). Though the initial design of the WWW was targeted towards individual users, Web 2.0 supports online group-based activities and communications [491]. Online communities are becoming a common choice for groups of people with similar interest who are located in different places. In many domains, such as health [17] and education [29], communities are essential for research and development.

Educational DLs are becoming useful for teaching and learning. To be useful, an educational digital library must provide access to a range of educational resources (e.g., syllabi, book reviews, collections of teaching aids), as well as a wide range of services for the life cycle of information: collection, creation, dissemination, use, and reuse [254, 597]. While a number of such educational DLs exist, there is also the factor that instructors as a community are not sufficiently engaged in educational sites [19, 507]. An active online community of educators can not only generate additional information on available resources (e.g., quality information for materials through ratings or reviews) but also list common practices involving those resources.

While most digital library architectures support a number of collaborative tasks like commenting and rating, there has been less work done on successfully incorporating an online community within a digital library. Extending the traditional DL would allow it to be more adaptive and reflective to the community it serves. The digital nature of DL means that it can capture and present useful information to end users that cannot be done easily in a real life library. Such information includes object access rates, object usefulness based on user ratings/reviews, listings of similar objects, etc. Some of this information is generated automatically based on content similarity, some by those who use the digital object (e.g., most viewed object). Though the content of digital libraries is accessed and used by different audiences (e.g., educators, students, researchers), most users are in the dark about additional information that exists in the community (e.g., which object is useful, how to integrate the object). Proper cues and community support have the potential to motivate users to document their knowledge about the objects, thus capturing community knowledge [400].

In this chapter we explain how to use implicit user information to deduce a social network. Such networks can allow a DL to have tailored services based on user needs and trends.

## 10.2 RELATED WORK

### 10.2.1 ONLINE COMMUNITIES

Online communities depend on between-user interactions to become successful. Girgensohn, et al. [242] looked at social interactions in web sites. They identified three sociological design challenges for building a successful socio-technical site: encouraging user participation, fostering social interactions, and promoting visibility of people and their activities. They also mentioned two technical design issues that are important in a socio-technical site: usability and low maintenance cost. Two of their target sites, CHIplace and Portkey, used a number of services like featuring news in the front page, sending newsletters, providing extrinsic reward for various activities, and varying levels of registration to increase user participation.

## 286 10. SOCIAL NETWORKS IN DIGITAL LIBRARIES

Koh, et al. [343] studied user participation in virtual communities in detail. They explain that participation can be of two types: passive (i.e., viewing) and active (i.e., posting). Each of these activities depend on different stimuli. They listed four major factors for an engaging virtual community: active leadership, offline interaction (e.g., offline meetings), usefulness of the content, and sound infrastructure.

User participation in online communities has been studied in depth from other perspectives. Nov, et al. [473] examined the effects of different types of participation on the levels of membership in the community. Ludford, et al. [400] studied the effect of showing both similarity and distinctness information about a member and the groups where s/he belongs as a means for increasing online community participation. Similar studies based on social theories were done by Beenan, et al. [47]. Millen, et al. [427] investigated design decisions, member selection, and facilitating stimulating discussion topics for engaging the members of an online community. Preece, et al. [523] studied community members to find out reasons behind lower participation among a particular group of less active users known as *lurkers*.

### 10.2.2 SOCIAL AND BEHAVIORAL NETWORKS

Social network analysis has many aspects, one of which is structural analysis. Researchers have used the structural properties of the network to identify communities or groups of entities [192]. Behavioral networks introduce an object-centric approach for connecting people. While social networks mostly depended on the human-ties, behavioral networks harness user trends to connect users who are otherwise disconnected. Esslimani et al. [177] used behavioral networks for deducing social networks among users.

### 10.2.3 SOCIAL NAVIGATION

Following user trends and learning from those trends allow a DL to be adaptive to its users. Often, common navigation tendencies are used to guide new users. Further, diverse areas including document recommendation [113], social network analysis [347], and Usenet news [346] rely on user trends and preferences to cope with information overload.

User trends are often used in recommendation systems, which can be divided into 3 broad categories: content-based, collaborative, and hybrid. In collaborative recommendation system, similarity between users is calculated in order to find the ratings of users that most likely have similar preference. Reviews and ratings of similar users are used to find the item that is likely to have the most interest. Thus topics of recommendation are not limited to similar subjects, and, based on population trends, they can vary widely. One of the first examples of collaborative filtering is Tapestry [247] which allowed users to help each other perform filtering over emails or electronic documents. This system also supported content-based filtering. Online retailers like Amazon and Netflix are using collaborative filtering to provide best matched items for a particular user taste.

## 10.3 THE NEXT GENERATION OF EDUCATIONAL DIGITAL LIBRARIES

The information needs of DL audiences vary widely depending on the nature of the digital library. In this section we focus on the needs of educators for teaching and learning. Ensemble<sup>1</sup>, the computing education portal within the National Science Digital Library (NSDL), supports a wide range of computing education communities, provides resources for developing programs that blend computing with other STEM areas (e.g., *X-informatics* and *Computing+X*), and seeks to produce digital library innovations that can be propagated to other NSDL pathways. Ensemble is a distributed portal providing access to the broad range of existing educational resources while preserving the identity of the individual collections and their associated curation practices. Ensemble encourages contribution, use, reuse, review, and evaluation of educational materials at multiple levels of granularity.

Ensemble made public the beta version of its web site in March 2010 and the production version in 2011. As the project moved forward, researchers at Virginia Tech conducted two focus groups comprised of nine business faculty members who teach computing to business majors. These participants constitute the majority (90%) of the department of Business and Information Technology (BIT). This pool broadens our perspective as it has different information needs compared to Computer Science educators — a group which dominates the Ensemble project team.

We anticipated learning about the techniques and challenges educators face when using online resources. Analysis of the discussions indicates that the problems we face in serving these users are related to the quality and quantity of content as well as the ability to manage that content. We also found that current DLs can be improved if they support social interactions. Based on our findings, we proposed DL 2.0, which integrates user interactions with resources in a DL. Thus our findings have the potential to be useful across various education communities as well as other digital libraries.

### 10.3.1 DATA COLLECTION AND ANALYSIS

There were two main phases in our research as described in Table 10.1: data collection and analysis. The department of Business Information Technology at Virginia Tech has a unique pool of computing educators who teach IT and CS courses to Business majors. We invited 10 faculty from this department. Five were present at the first session, and four at the second. Each session was an hour long.

Our questions to participants were split across two broad topics: (i) How do they search for educational materials? and (ii) What is their feedback on the Ensemble portal? We posed a set of 10 questions based on these two broad topics, which are listed below, to all participants.

<sup>1</sup><http://www.computingportal.org/>

## 288 10. SOCIAL NETWORKS IN DIGITAL LIBRARIES

1. How do you search for resources to use in a course, lesson, or assignment related to an IS/IT-oriented course?
2. In which content areas would you normally seek resources to support learning and teaching?
3. Which formats might be most helpful to your teaching or your students' learning?
4. Which resources do you have the most difficulty finding and accessing?
5. How do you stay up-to-date in your field in terms of education?
6. Which web sites do you visit or which materials do you make regular use of? Why?
7. Do you use publisher sites often for your assessment needs?
8. Do you participate in any special interest groups (SIGs) or meetings to enrich your teaching or any social group? Do they have an online community site for it?

**Table 10.1:** Phases of Data Collection and Analysis

Data Collection	
System Review	Identified key areas of Ensemble for further research and development.
Protocol Development	Created a protocol and a set of questions for the focus groups.
Focus Groups	Virginia Tech (VT) conducted two focus groups. Each focus group was roughly one hour in duration.
Participants	Each of the 9 participants were Business faculty who teach computing to Business majors.
Data Analysis	
Transcription	Audio recordings were transcribed and combined with handwritten notes taken during the session to create a combined report of the two focus groups.
Coding	We identified repeated answers, patterns, and behaviors in the transcribed data and in the report. These were coded based on the themes they represented.
Themes	The codes were used to identify emerging themes which were then used to develop and connect high-level codes about the prevalent practices on locating and using electronic resources, and on creating active users in an educational DL.

### 10.3. THE NEXT GENERATION OF EDUCATIONAL DIGITAL LIBRARIES 289

9. How valuable do you consider the use of badges and rewards in building an online community?
10. What are your thoughts about the Ensemble web site?

We followed the grounded theory approach to analyze the data. Initial coding was done to identify recurring themes, or examples related to a theme, which resulted in 29 codes. Many of these codes relate to an underlying broader theme which helped us to identify different aspects of the code. For example, the code *Ease of navigation* (14 references) referred to various aspects of navigating through a site. While some participants argued that *organization* of content is a major issue for easy navigation, others were inclined toward better *search mechanisms*. We did not tie specific codes to specific questions. Participants provided more information as we progressed through the sessions, causing the same code to be linked with multiple questions. There were 246 references to these codes in the original transcripts. Figure 10.1 shows some of the top codes with their reference counts. For example, participants mentioned *format or type of the resources* 29 times. YouTube and educational video clips were mentioned as either motivating tools for students or informative resources. There also were mentions of syllabi, lecture notes, and PowerPoint slides that educators often seek on the Internet. *Quality* of available material is also a big concern (18 references). Many participants pointed out that they reuse or borrow existing course material as a starting point (*Recycling courseware*, 16 references).

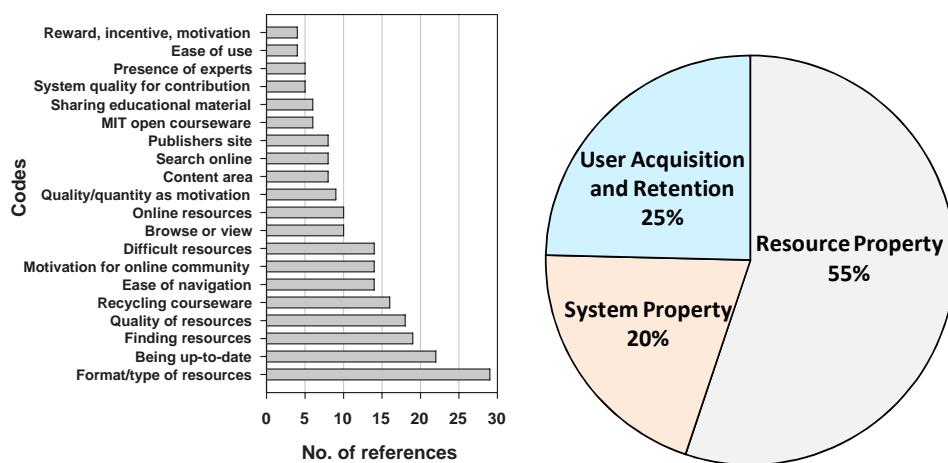
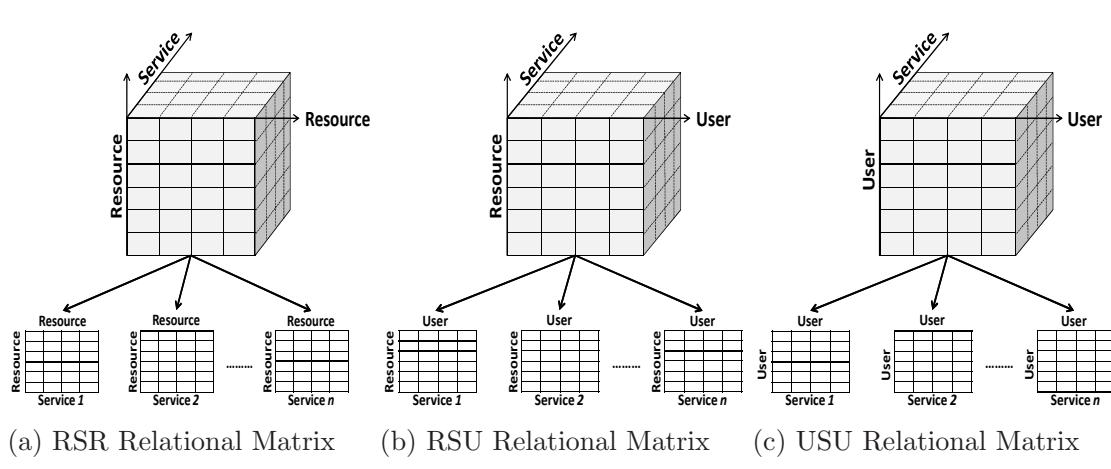


Figure 10.1: (Left) Sample codes with number of references. (Right) Distribution of references in three major themes described in Table 10.2

**290 10. SOCIAL NETWORKS IN DIGITAL LIBRARIES**

**Table 10.2:** Emerging Themes from the Focus Group Data

<b>Resource Property</b>
Format: Types/formats of educational materials.
Finding resource: Finding resource through Web search (e.g., Google), university sites (e.g., MIT OpenCourseWare), and personal connection.
Quality: Quality of available resources at various sites.
Recycling courseware: Reusing course material or borrowing course content.
<b>System Property</b>
Factors influencing site use.
Ease of navigation - Organization of content: Easier topical organization following any standard organization scheme.
Robust search: Visible search box/tab and granular searching options.
Interface: Takes less time to get used to and use the resource.
Association between content.
Factors influencing contribution.
Ease of contribution: Contribution should not take time.
Personalization
Content customization: Ability to customize textbook or assessments.
Add content to user list: Create personal collection from existing resources.
Differential access to resources: Access control to resources, especially for assessment materials.
User Acquisition and Retention
Motivation for using the site.
Existence of quality resource.
Existence of large quantity resource.
Existence of peer reviews.
Existence of experts in the community.
Critical mass: Large user base.
Saving time as a motivation for joining an educational DL.
Motivation for contribution
Peer recognition.
Quality of community and resources in the site.
Reward, incentive.
Academic recognition for contribution (e.g., promotion and tenure).
Building reputation (e.g., roles, badges) based on user activities.
Peer recognition.



Example of services (in bold text) for each relational matrix

- Linking **resources** (e.g., tags).
- Associating **resources** (e.g., exercises linked to a lecture slide).
- Peer reviews (e.g., ratings).
- A resource can have an **owner**.
- A resource can be **read/downloaded**.
- Users can **contribute** additional information (e.g., comments, ratings).
- Users can be **members** of a group or community.
- Users can **contact** other users.
- Users can be **connected via shared resources** (e.g., co-authors).

Figure 10.2: Relationship between Resource and User

After the initial coding, we grouped the codes based on their relevance to a set of broader themes. Three themes that emerged were Resource property, System property, and User acquisition and retention (Table 10.2). Resource property includes types of resources used by educators, difficult resources, methods on how to find resources online, etc. System property lists various aspects of a site that encourage participants to use the site. User acquisition and retention refers to factors that motivate users to actively use a site and participate. Some of the initial codes related to each of these themes are listed below them (see Table 10.2).

The codes in Table 10.2 reflect characteristics of an ideal DL, which are similar to those of Web 2.0. Web 2.0 provides a dynamic environment for users by supporting sets of activities that promote social interactions, encourage user contribution, or capture and

## 292 10. SOCIAL NETWORKS IN DIGITAL LIBRARIES

Table 10.3: Comparison between DL 1.0 and DL 2.0 based on 5S Definitions

5S Elements	DL 1.0	DL 2.0
<b>Stream</b>	Metadata of resources only.	Metadata with community-contributed information (e.g., comments, ratings, reviews) on resources.
<b>Structure</b>	Single listing of resources belonging to a particular collection/topic.	Cross-referenced resources across collections and attributes.
<b>Space</b>	Does not handle multiple spaces.	Supports multi-layered resource spaces. These layers can support various space-related entities (e.g., time series, feature spaces).
<b>Society</b>	Does not explicitly support group-oriented tasks.	Supports groups, communities, collaborations as well as individual user tasks.
<b>Scenario</b>	Services include browse, index, and search.	Services include personalization, recommendation, better organization, user-friendly navigation, faceted search, advanced ranking based on popularity, users' comments, ratings, tags (CRTs).

highlight collective knowledge. We propose Digital Library 2.0 for educational resources, that takes a user-centric approach by providing services to connect users and resources, and by hosting online communities.

### 10.3.2 DL 1.0 VS. DL 2.0

*Formal Definition:* DL 2.0 is a combination of three basic entities:  $\mathcal{R}$ ,  $\mathcal{S}$ , and  $\mathcal{U}$  (resource, service, and user, respectively). DL 2.0 architecture is dependent on three different arrangements of the basic entities:  $\{\mathcal{RSR}\}$ ,  $\{\mathcal{RSU}\}$ , and  $\{\mathcal{USU}\}$ . Service is the connecting entity, relating resources with other resources, resources with users, and users with users.

### 10.3. THE NEXT GENERATION OF EDUCATIONAL DIGITAL LIBRARIES 293

A service that connects two entities can implicitly create connections between or among other entities. Figure 10.2 shows these relations with examples. Figure 10.2(a) shows the Resource-Service-Resource (RSR) relational matrix. For each service in this relational matrix, there will be relationships between some of the resources. For example, a resource might contain annotations (which is another type of resource). Figure 10.2(b) shows the Resource-Service-User (RSU) relational matrix. A resource can be connected to users via a number of services such as authoring or viewing. Figure 10.2(c) presents the User-Service-User (USU) relational matrix. Connections and interactions between users would allow for a virtual social environment that is desired by a large number of participants.

More than half of the codes from our initial data analysis phase were related to some property of resources (see Figure 10.1, right). **Organization** and interconnection between resources are important to users. Participants identified a number of problems with various organization schemes used at different sites, with the most common being learning the many different organization schemes. One suggestion was to use existing standards to create the categorization scheme. This would allow all resources to be organized by a set of well-known topics. Use of non-standard terms was also confusing to many users. **Association** between content can be useful to users. Participants noted that they like to explore and use resources that are related to their course content. This highlights the fact that an individual resource page serves few information needs of educators who would prefer the resources to be linked properly. DLs need to have a robust organization scheme of content and proper association between various resources. Using deep navigation trees can be confusing. If the content is buried under five or six levels, a user often loses track of the context. Tags or lists with low depth can be useful. One suggestion was to show the context (e.g., tree, bread crumb). When applicable, information such as the link to the actual content should be *eye-catching* or visually appealing. It was suggested that for a DL that hosts groups and communities, the navigation scheme should be consistent across collections, communities, and other sections. **Search** is considered as an essential service. Several participants mentioned frequent use of advanced search features to locate relevant materials among a large number of resources. This feature is used even by those who are familiar with the site.

DL 2.0 is the next-generation approach to DL that blends the traditional digital library contents with user-contributed contents and provides online community support (e.g., relationship management among users and digital contents, such as user interactions, ratings, comments, bookmarks, queries, etc.). The core difference between traditional DL 1.0 and DL 2.0 lies in the fact that the latter is more dynamic, user-centric, encourages user contribution, fosters virtual community, and incorporates knowledge with resources. While core services of DL 1.0 were limited to searching, browsing, and indexing, DL 2.0 encompasses content management, dynamic services such as customization or personalization of content, and a collaborative environment. Table 10.3 provides a comparison of the 5S elements (recall Chapter 1) between DL 1.0 and DL 2.0.

## 10.4 FINDING COMMUNITIES IN DIGITAL LIBRARY

Building and sustaining an active community is difficult. This problem becomes even harder for an object-centric environment such as a digital library where members can submit, rate, review, or comment on an object. One way of engaging users in various activities (e.g., view, rate, comment) in the library could be to show what others have done. To find trends or communities in a DL, we can start by looking at any available social network in that DL.

Social networks can serve a number of purposes in a digital library. They can help in harnessing and spreading community knowledge. They can be used to identify common practices of users. They even can help users with a similar interest to interact with each other.

We can divide social networks into two broad categories, described below.

**Active Social Networks:** These networks are actively created by users. There are websites that specifically allow users to link with other users by sending requests. Social networking sites such as LinkedIn, CiteULike, or Delicious follow this principle. Along with building the network in these sites, users provide information on their background and preferences.

**Passive Networks:** There exists another set of sites where the focus is more on providing information than on creating social networks. Digital libraries fall in this category. These libraries act as a rich source of information and provide services to facilitate information retrieval. User activity in these DLs can be roughly divided into two categories: Passive and Active. Passive user activity includes viewing a page or downloading an object. Active user activity is content contributed by the user that can appear in many forms: new resources, ratings, reviews, comments, etc. A goal discussed in this chapter is to find and utilize passive user behaviors.

User logs provide useful information. Like other websites, DLs are now logging user data. These data can allow us to deduce the underlying network of user behaviors. Such a network, which may connect one user with another based on their navigation history, can be interpreted as a behavioral social network.

Passive social networks take a more object-centric approach rather than the relationship-centric approach of active social networks. A wide range of objects can be used in a DL to create passive social networks. For example, in an educational DL, there are often groups, collections of metadata, and tools for teaching. It is possible to use these objects and user trends to form different networks. Described below are examples of how a network can be deduced in such a DL.

1. Deduced social networks (DSN): We can connect the users based on their activities in the site. For example, we can create a network of users based on their viewing of the same pages. In this network, the users will be the nodes, and an edge between

## 10.4. FINDING COMMUNITIES IN DIGITAL LIBRARY 295

two users will indicate that they have viewed a same page. The following subsection contains more details on formalizing and constructing DSNs.

2. Navigation networks of objects: It is not necessary that we only rely on user networks to find useful information. A connection between objects can be similarly interesting. For example, finding a group of objects that users view within a session can reveal useful information on their usage.
3. Navigation/Usage network of tools: Networks of downloaded tools will allow us to find another set of patterns, which may identify the usage trends of these tools.

Passive networks, such as DSNs, can help us understand how users behave in a DL. When used appropriately this information has the potential to improve DL services.

### 10.4.1 SOCIAL GRAPH CONSTRUCTION USING LOGGING AND METRICS

A digital library for a specific user community may contain disparate networks. Discovering these networks will allow us to get a better understanding of user interactions [80]. That knowledge then can be used to provide personalized user services.

A large number of personalized services depend on web metrics. Usage log analysis shows key pieces of information like average time spent on a certain page, bounce rate, and exit percentage for a webpage. Identifying similar users depending on their interests and providing them with possibly helpful contents to explore is another advantage of having a robust logging system. Having metrics also can facilitate social navigation (e.g., show contents that are viewed together in a user session).

Generic social networks tie users with other users of the same site. Besides user-user interaction, it also is possible to derive connections between users by way of different objects, such as the pages they view [177]. While most attributes of traditional social networks are static (e.g., user is a member of a community), user logs and metrics can provide a range of dynamic information (e.g., browsing history). It is possible to create a range of networks with this dynamic information. Analysis of these networks and their contextual information can reveal interesting user behavior, different user roles, and communities with similar interests. Use of log data also will help developers making changes in the DLs to enhance user experiences.

**Definition 10.1** An Educational DL is a tuple (*Collectedcontent*, *User*, *Service*, *Generatedcontent*), where:

- *Collectedcontent* is the metadata coming from different collections and user submissions - listed under various topics,
- *User* is the set of people who uses *Collectedcontent*. For example, educators, students, policy makers, etc.

## 296 10. SOCIAL NETWORKS IN DIGITAL LIBRARIES

- *Generatedcontent* is the content generated by the *User* on the *Collectedcontent*. Examples include: comments, ratings, tagging (CRT), and
- *Service* refers to the functions in an Educational DL that generates a set of content-both *Collected* and *Generated* based on a *User* query.

**Definition 10.2** A Deduced Social Network (DSN) is a Graph with tuple (*Entity*, *Object*, *k*), where:

- *Entity* is the node of the network,
- *Object* is an attribute linked to *Entity*, where multiple instances of *Object* can be listed under one *Entity*; and
- *k* is a function that returns the minimum number of *Object* that must be common between two *Entity* to create a connection (i.e., edge) between them.

Figure 10.3 shows examples of deduced social graphs. The figure shows two (top) social graphs constructed based on two different types of log information: topic of interest and URLs visited. Their overlaps can reveal certain groups with special interests. The figure at the bottom shows such an overlap of two social graphs constructed based on disparate criteria.

## 10.5 ANALYSIS OF PASSIVE SOCIAL NETWORKS

Network analysis can help us understand hidden trends and user preferences. Depending on the network characteristics, a number of approaches can be used to analyze the network. For example, for dense graphs, graph partitioning can help us to find smaller sub-graphs that might reveal interesting information. Similarly, it is possible to analyze pairs of passive networks. We describe below some approaches that might be useful for analyzing the passive networks discussed in the previous section.

### 10.5.1 GRAPH PARTITIONING

When it comes to understanding user trends, constructing graphs alone is not sufficient. We need to study the properties of the graphs in order to get a meaningful insight into the users' behavior. There are a number of ways in which graphs can be analyzed, graph partitioning being one of them. The graph partitioning approach breaks down the graph into disjoint subsets such that the number of connections within the subsets are high but the number of connections between the subsets are low. Graph partitioning has been studied

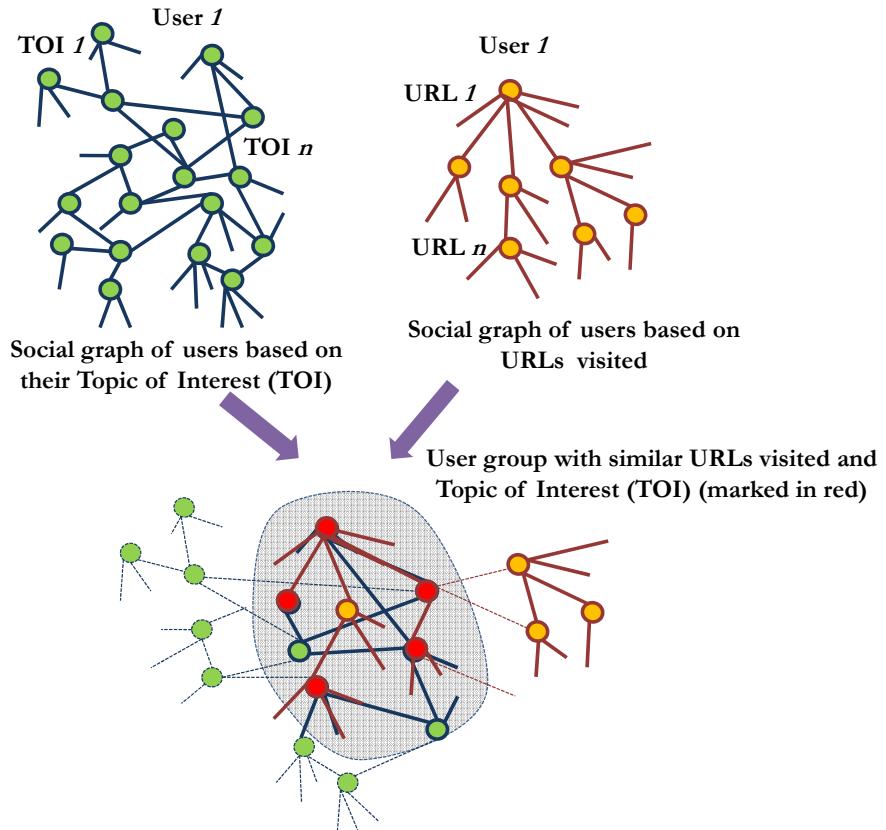


Figure 10.3: Social graphs can be created using different log information.

in various areas including web science [304], epidemiology [273], sensor networks [554], and parallel computing [289, 357].

Networks have been extensively studied in a variety of fields including physics, biology, and sociology (to find communities) [582, 496, 122]. Newman and Girvan used betweenness as a measure for removing edges and finding communities in graphs [243]. Clauset et al. [122] used hierarchical agglomeration algorithms for detecting communities in large networks. Among various clustering techniques, spectral clustering and modularity clustering are gaining momentum in community detection.

**Spectral clustering** Spectral clustering refers to a clustering approach that depends on the eigen vectors of the similarity matrix to partition the data points into disjoint clusters such that the similarity of points within the cluster is high while that across them is low [651]. Unlike most clustering technique, spectral clustering transforms the objects

## 298 10. SOCIAL NETWORKS IN DIGITAL LIBRARIES

into a set of points in space. These points then can be clustered using standard clustering algorithms. Fortunato [191] described a number of techniques for community detection including spectral clustering.

While spectral clustering has been used effectively in areas such as image segmentation, it can encounter difficulties in detecting different community structures. For example, a dense community may consist of nodes of similar degrees, while another community may have few central nodes with high degree but many nodes with lower degree. With the various social networking groups in Web 2.0, the web alone is hosting networks of different structure, dynamics, and behaviors. Users are connecting with other users, joining groups, or following one another. Shah and Zaman [593] used network centrality to propose the leader-follower algorithm for communities that are formed around a few central users. Their approach, when compared to spectral clustering, performed better in locating the communities.

**Modularity clustering** Modularity, introduced by Girvan and Newman, is a quality measure for clustering that has been successfully adopted in many areas [243, 468]. Modularity clustering is dependent on edge betweenness — a measure that assigns weights on an edge as the number of shortest paths between pairs of vertices containing this edge. If a network contains multiple communities then the number of edges connecting the communities will be less than the number of edges within the community and all shortest paths between those communities will contain one of those edges that connect the communities. Thus, the edges that connect the communities will have relatively higher edge betweenness values.

### 10.5.2 TOPIC MODELING

Finding clusters alone is not sufficient to understand the pattern. Topic modeling allows us to get an overview of the subjects addressed in the clusters. Probabilistic models such as Latent Dirichlet allocation (LDA) [61] have been used extensively to detect topics for a document corpus. LDA is a generative probabilistic model that uses a fixed vocabulary and assigns a Dirichlet distributed vector to each document while detecting the topics in a document corpus. One of the shortcomings of LDA analysis is its requirement of a fixed number of topics. Blei et al. [60] addressed this issue with Bayesian non-parametric methods and introduced hierarchical Dirichlet processes (HDP) that can handle arbitrary numbers of topics and can generate new topics for previously unseen documents.

LDA is also expanded to analyze linked documents. Chang and Blei introduced the relational topic modeling [111] approach, where along with LDA, the links between the documents are modeled as binary variables that suggest whether a pair of documents is linked or not. According to this approach, the link information is connected to the content of the documents. While most topic modeling research focuses on document corpora, Ramage et al. [533] used LDA to analyze topics appearing in micro-blogs.

### 10.5.3 ANALYZE A PAIR OF PASSIVE NETWORKS

As we showed, it is possible to create a number of different passive networks in a DL using the log data. It also is possible to create networks of objects within a DL based on their attributes (e.g., topics, creator). Finding relations between these networks can be useful in deducing both user needs and trends. Co-clustering or bi-clustering techniques simultaneously cluster the data-points in two different areas. Co-clustering is now frequently used in areas such as text mining [157], bioinformatics [117], and image segmentation [650].

## 10.6 CASE STUDY: THE ALGOVIZ PORTAL

Online communities are using social navigation to help guide users through large collections. The distributed nature of communities and content in a portal demands a system that will allow users to see how other users have walked through the information space.

The AlgoViz portal collects user history in several different tables in the database. A sample of one of these, the Accesslog table, is given in Table 10.4, showing data on the session, user IDs, hostname, timestamp of when the page was visited, etc. Algoviz content is open for public viewing, hence it is possible for users not to register and receive an user ID. These users are referred to as Anonymous users and so have a default user ID of 0. We used hostnames (i.e., IP addresses), instead of user IDs, to identify any trends in our user-base. For any given hostname, we are able to view which pages were viewed in that session, and the time when the page loaded. We used this data to deduce a behavioral social network. Additionally, the accesslog table uses access-id (aid) as the primary key. It also stores session information. Each page viewed in a session generates a new aid (i.e., row) in the table. We selected two months (i.e., September and October) from 2010 for processing.

Table 10.4: Log data for AlgoViz

Session ID	Page Title	Internal Path/Page URL	Hostname	User ID	Timestamp
ievav83	Lifting the hood of the computer...	node/1413	93.158.151.25	0	1276272047
t5fuuba	biblio/export/tagged/118/popup	research.cs.vt.edu/algoviz/biblio	207.241.228.172	0	1276260935
ivuks8s	Has an AV helped you learn a topic in computer science?	research.cs.vt.edu/algoviz/poll/	95.79.100.90	0	1276260943

### 300 10. SOCIAL NETWORKS IN DIGITAL LIBRARIES

On an average, a month has 100,000 rows in the table. Much of the data are generated from spammers, crawlers, bots, etc. We followed a three-step process to clear the log data of such outliers, as is described below:

1. Filter data based on page titles: Many pages in AlgoViz are generic and less informative for understanding user behaviors. At the first stage of data cleaning, we prune the rows based on the titles of the pages. Examples of pages titles that are less informative and less important include: ‘Welcome’, ‘Access denied’, ‘Page not found’, etc.
2. Filter data based on the path: Sometimes the title of the page alone is not sufficient to understand content. For example, the profile page of a user contains the user name as the title. The accesslog table logs the internal path of the page. We used these paths to prune rows that include generic internal paths such as ‘user/register’, ‘user/login’, ‘node’, etc.
3. Filter data based on the session information: At this phase, we investigated user behaviors based on the session information. It was at this step where we can detect general session-wise behavior such as average pageviews per session, average length of session if at least two pages were viewed, etc. With such information, we are able to identify the outliers, possibly bots, in the log data, and filtered them out. The session-based pruning was done in three stages as described below.
  - (a) Pages per session — count of the number of unique pages generated in a session and prune if value is greater than a certain threshold  $x$ .
  - (b) Seconds per page — on average, in a session, how much time was spent on a page and prune if value is greater than a certain threshold  $y$ .
  - (c) Number of sessions with one pageview: Usually a session would contain multiple pageviews. We observed a tendency of generating a large number of sessions with only one pageviews for certain hostnames (i.e., IP address). We pruned all the rows containing the suspicious hostname.

#### 10.6.1 DEDUCED SOCIAL NETWORKS

The filtered data was then used to connect pairs of users based on their common pageviews. These connections created a network which we call Deduced Social Network (DSN). Nodes represent a user and the edges indicate they have viewed at least  $k$  common pages. We used a *connection threshold* parameter to vary the network strength. A *connection threshold* of size  $k$  for an edge indicates that two users have viewed at least  $k$  common pages.

Figure 10.4 shows a DSN based on the log data of AlgoViz for the months of September and October 2010 — with a connection threshold of 10. Two users were connected only if they viewed at least 10 similar pages. We used a force-directed graph layout [170] to draw

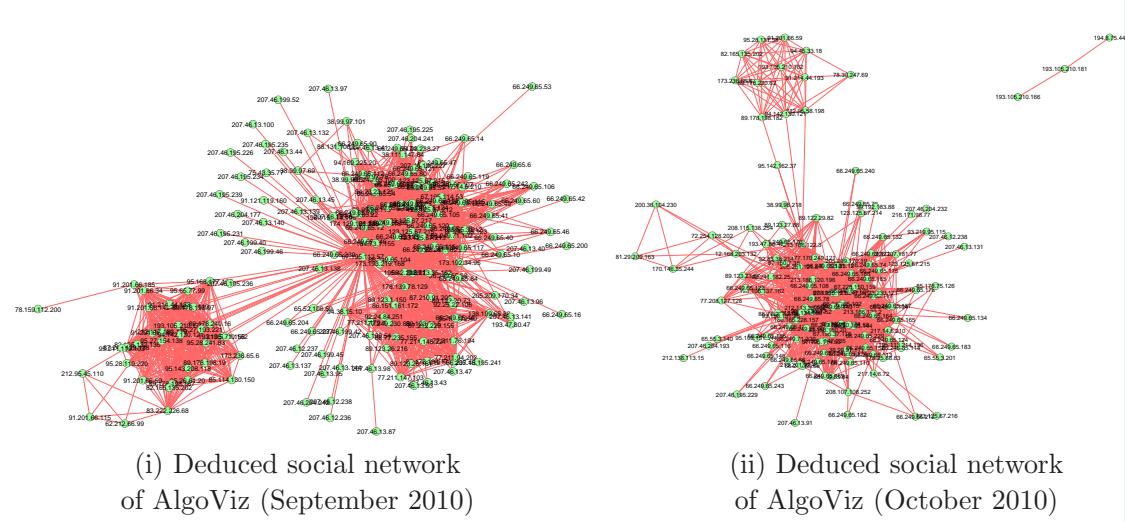


Figure 10.4: Using log to find user communities with similar interest (connection threshold  $k=10$ )

these networks. The network shown in Figure 10.4 (i) has 195 nodes and 2255 edges, while the network in Figure 10.4 (ii) contains 130 nodes with 1180 edges.

**Network Characteristics** Among various measures, node degree distribution and betweenness centrality are used frequently to understand the characteristics of large networks. Provided below is a brief description of these values and their usefulness.

**Distribution of node degree:** The DSNs are undirected graphs where the node degree of a node  $n$  is the number of edges connected to it. Degree distribution is an important measure for understanding whether the network is random or scale-free.

**Betweenness centrality:** The betweenness centrality value assigns weights to a node  $n$  based on how many times this node appears in the shortest path between all the other nodes. We use a normalized value so that the betweenness centrality of each node is a number between 0 and 1. If betweenness centrality is high it indicates that the node is present in many shortest paths. Removing a node with high betweenness centrality can create strongly connected components in a network, thus revealing sub-groups.

Figures 10.5 and 10.6 provide these values for the respective DNSs.

### 10.6.2 COMMUNITY DETECTION ON DSN

As we see from Figure 10.4, the graphs are very dense, making it difficult to analyze user trends. Graph-partitioning methods are used to find sub-graphs or communities within such dense graphs. We used Modularity clustering that has been successfully used to find

302 10. SOCIAL NETWORKS IN DIGITAL LIBRARIES

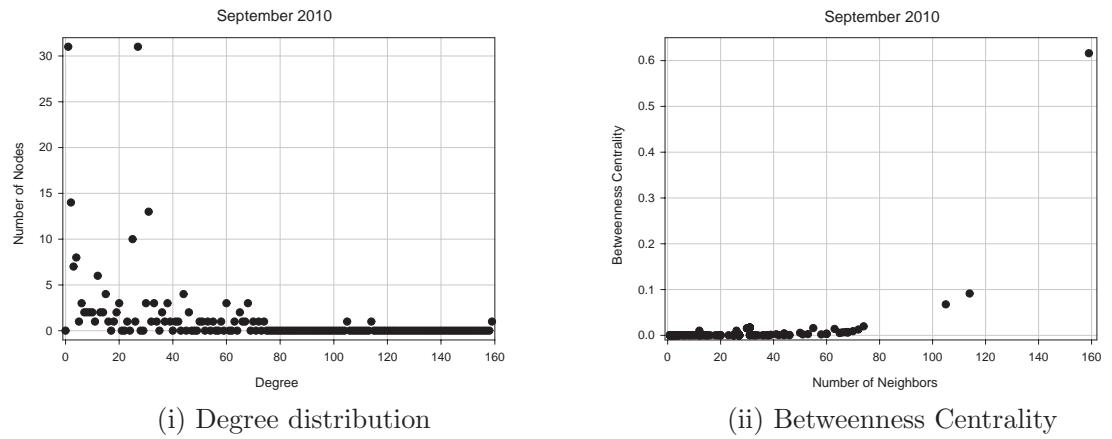


Figure 10.5: Network Statistics of the DSN for September 2010 (connection threshold  $k=10$ )

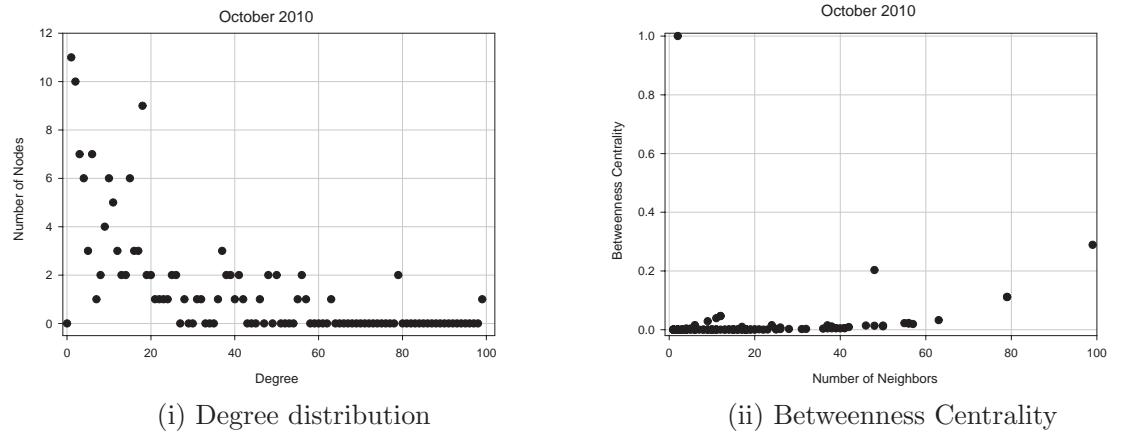


Figure 10.6: Network Statistics of the DSN for October 2010 (connection threshold  $k=10$ )

communities within large networks in other domains. The result of the clustering is given in Figure 10.7. Also, the user (i.e., node) distribution in the clusters is given below the cluster result. The figure shows that most of the clusters consist of more than 20 users. Also, there are clusters with low numbers of users, indicating possible outliers (e.g., cluster 4 in September 2010 has 2 members). Thus, the clustering result has the potential to be used for further data cleaning.

## 10.6. CASE STUDY: THE ALGOVIZ PORTAL 303

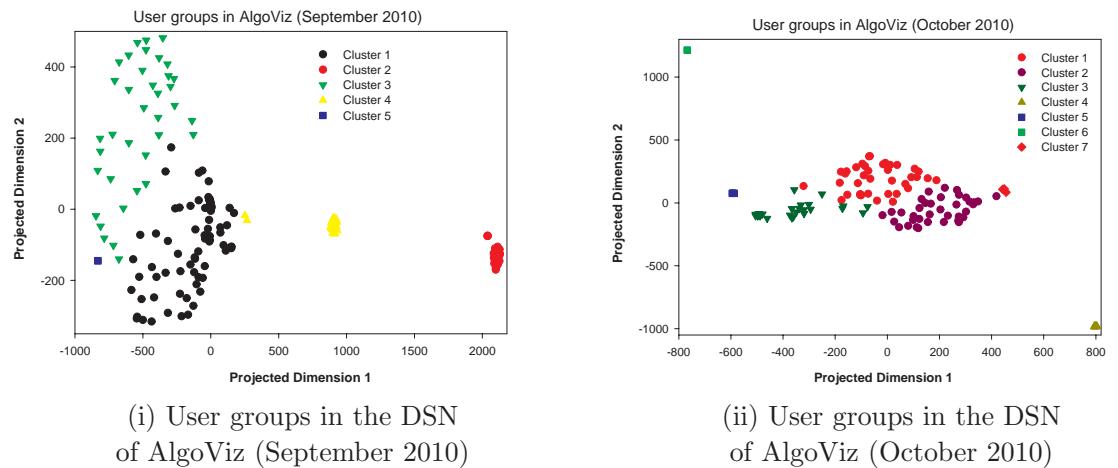


Figure 10.7: Clusters found in the DSNs of Figure 10.4

### 10.6.3 COMMUNITY INTERESTS

While we are currently working on topic modeling, we used tag clouds to get an overview of the page titles in the clusters. Following figure shows the title of the pages for each clusters in AlgoViz for November 2010.

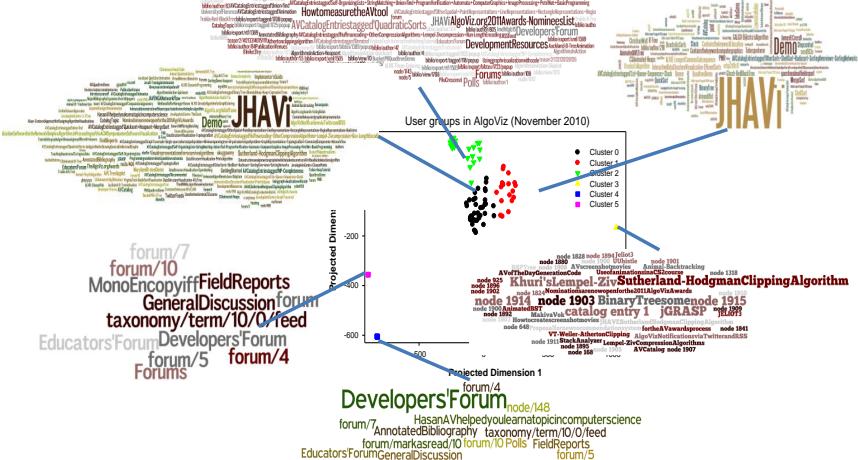


Figure 10.8: Page titles of clusters found in the November 2010 DSN

## 10.7 EXERCISES AND PROJECTS

1. How does an authority control system relate to developing and maintaining a digital library with an integrated social network?
2. Suppose that Scholar provides us with user data. What information can you find from the data that can be beneficial to the user as well as the developers of Scholar?

## CHAPTER 11

# Education

by Eric Fouh and Yinlin Chen

*Abstract:* Education is one of the most common applications of digital libraries (DLs). The development of e-learning has increased the importance and the presence of digital libraries in most educational processes. Educational DLs allow members of the educational community to create, evaluate, share, and preserve educational resources. Successful education DLs contain high quality educational resources and are easy to browse, search, and access. In order to have broad impact on the communities in education and to serve for a long period of time, digital librarians need to structure and organize the resources in a way that facilitates the dissemination and the reusability of the resources. In addition, digital librarians who collaborate with patrons have to repeatedly review, verify, and extend the resources they have collected and harvested. Thus, data quality is particularly important in educational DLs. In this chapter, we use the 5S framework perspective first to describe educational DLs, and then to review current education DL applications. Next, we discuss key concepts regarding data quality in educational DLs. Lastly, two educational DLs, *AlgoViz* [19] and *Ensemble* [176], serve as case studies.

## 11.1 INTRODUCTION

Libraries have always been the cornerstone of education and can be viewed as a sign of great scholastic and technological achievements. With educators and students relying more on computers in most educational activities, the need for software and digital educational materials that will support those activities becomes a major challenge in e-learning. Digital educational materials need to be organized in educational digital libraries (EDLs) in order to improve the accessibility and reliability of such resources.

EDLs aim to address several needs of the educational community: how to find educational resources, how to use or integrate the resources gathered, how to share educational materials, and how to assess the quality of the resources. The 5S framework provides us with a formal model to describe and build effective digital libraries, that has been used in several DL applications [202, 214]. Accordingly, this chapter focuses on presenting a 5S perspective on educational DLs.

## 11.2 RELATED WORK

The United States National Science Digital Library (NSDL) is a national network of digital environments dedicated to advancing science, technology, engineering, and mathematics (STEM) teaching and learning, in both formal and informal settings. Ensemble is an NSF (United States National Science Foundation) NSDL Pathways project working to establish a national, distributed digital library for computing education. Based on their experience, and lessons learned from working with different NSDL projects, the Ensemble working group proposed 8 principles for distributed portals (PDPs) [205]. They advocate for

- Articulation of EDLs across communities using ontologies
- Browsing services should be tailored to collections
- A close Integration across interfaces and virtual environments
- Metadata interoperability and integration
- Social graph construction using logging and metrics
- Superimposed information and annotation integrated across distributed systems
- Streamlined user access with IDs
- Web 2.0 multiple social network system interconnection

Some DLs are built using open source softwares, such as PLoS ONE<sup>1</sup>, which is a communication tool for peer-reviewed science and medicine. PLoS ONE is built upon the innovative technologies of Topaz, Fedora, and Mulgara, providing an open publishing platform that combines an online scientific journal with community features such as tags, annotations, discussions, and ratings.

Forced Migration Online [543] is a comprehensive web site that provides access to a diverse range of relevant information resources on forced migration. It is a technically and intellectually administered resource, combining specialist subject knowledge with high standards of information management.

The ARROW project's [525] aim is to identify, test, and develop software or solutions to support best practice institutional digital repositories. While initially a research project, ARROW has now co-developed a working institutional repository solution.

The three above-mentioned systems use the Fedora Commons software. On the other hand, more than 1000 DLs are built using DSpace [42]; almost all of them are academic and research center projects. DSpace is free software developed by HP and MIT to primarily help educators build open digital repositories. Its design was driven by the following goals: provide an easy way to browse, submit, and retrieve documents using a digital asset store; be

<sup>1</sup><http://www.plosone.org/home.action>

### 11.3. FORMALIZATION: EDUCATIONAL DL 307

economically viable; provide authentication mechanisms; create a community; etc. Since the middle of 2009, Fedora Commons and the DSpace Foundation have been working together to create the DuraSpace organization [341].

Summer et al. presented a model composed of three concepts to build EDLs [618]. They claimed that their constructs provided a structural way to build educational digital libraries that will fully capture the interaction between humans and technology along with contextual information. They present EDLs as *cognitive tools* meaning that it should have interfaces, tools, and services that will help the users to make sense of information from different sources and of different types (text, image, etc.). Their second construct refers to EDLs as *component repositories*. They argue that EDLs should be designed with the goal of improving the quality of education by enhancing the reuse of resources, sharing best practices, etc. Their last construct depicts EDLs as *knowledge networks* indicating that EDLs' services and tools should foster interactions between all users in order to support knowledge building. The separation between the *content* of the EDL and the *context* in which the resource is going to be used is also an important principle. Further, the growth in the number, size, and diversity of digital collections makes metadata quality an increasingly important issue. In order to build systems that will address all those concerns we need to have a formal model to analyze and describe all of the elements involved in DL design and implementation. To the best of our knowledge the 5S framework seems to be the most appropriate tool.

## 11.3 FORMALIZATION: EDUCATIONAL DL

Education can be viewed as a form of formal training happening in an institutional setting. Education engages all stakeholders into several types of learning activities.

**Definition 11.1** **learning** is a tuple  $(K, f)$  where:

- $K$  is a set (body) of knowledge concepts, and
- $f$  is a function,  $f = \{\forall x \in K, \text{ associates } f(x), \text{ where } f \text{ is anti-reflexive}\}$

$f$  can be seen as a transition event on the state set  $K$  (of knowledge).

Learning is about modifying the state of knowledge of individuals, where that change is triggered during learning activities.

**Definition 11.2** **A learning activity** is a tuple  $(U, S, L)$  where:  $U$  is a set of users,  $S$  is a set of scenarios reflecting how the users interact with each other, and  $L$  is a set that represents the learning states of the users before and after the interactions.

### 308 11. EDUCATION

**Definition 11.3** An Educational DL is a tuple (*Collectedcontent*, *User*, *Service*, *Generatedcontent*), where:

- *Collectedcontent* is the metadata coming from different collections and listed under various topics,
- *User* is the set of people who uses *Collectedcontent*. For example, educators, students, policy makers, etc.
- *Service* refers to the functions in an Educational DL that generates a set of *Collectedcontent* based on a *User* query, and
- *Generatedcontent* is the content generated by the *User* on the *CollectedContent*. Examples include: comments, ratings, tagging (CRT).

Davis et al. advocated that innovative design and delivery of services should leverage technology in order to create services that will increase self service, automation, globalization, choice, information, and fidelity [143]. In the context of EDL, self service means the user needs less time and minimal external help or support when interacting with the system. It should come as result of all the (underlying) services being automated. It also is desirable to have all the system resources available anywhere (globalization). All these changes should lead to more choices for the user because of the availability and the (high) amount of accessible information. An EDL should acknowledge the contributions of the most active members of the community. Some of the most important of those contributions are new resources.

**Definition 11.4** A learning object (lo) is a digital object which can be used, re-used, or referenced during technology supported learning [670].

It is a tuple:  $lo = (h, SM, ST, StructuredStreams, c)$  where

1.  $h \in H$ , where  $H$  is a set of universally unique handles (labels);
2.  $SM = \{sm_1, sm_2, \dots, sm_n\}$  is a set of streams;
3.  $ST = \{st_1, st_2, \dots, st_m\}$  is a set of structural metadata specifications;
4.  $StructuredStreams = \{stsm_1, stsm_2, \dots, stsm_p\}$  is a set of StructuredStream functions defined from the streams in the  $SM$  set (the second component) of the digital object and from the structures in the  $ST$  set (the third component).
5.  $c$  is a tuple  $c = (c_i, c_j)$  with  $c_i$  is the context within which learning takes place. Examples include: K-12, higher education, etc. And  $c_j$  is the subject context of the learning. Examples include: Computer Science, Mathematics, etc.

### 11.3. FORMALIZATION: EDUCATIONAL DL 309

Digital objects can be combined together to form “*compound (or complex) learning objects (clo)*”.

**Definition 11.5** A compound learning object is defined as a tuple  $clo = (h, SCL O, S)$  where

1.  $h \in H$ , where  $H$  is a set of universally unique handles (labels);
2.  $SCL O = \{LO \cup SM\}$ , where  $LO = \{lo_1, lo_2, \dots, lo_n\}$ , and  $lo_i$  is a learning object or another complex learning object; and  $SM = \{sm_a, sm_b, \dots, sm_z\}$  is a set of streams;
3.  $S$  is a structure that composes the compound learning object  $clo$  into its parts in  $SCL O$ .

Accordingly, Wiley divided a  $clo$  into two categories: *closed clo* and *open clo* [670]. A *closed compound learning objects (cclo)* is a set of *los* aggregated in a logical way during the design phase by its author(s). Further, an *open compound learning objects (oclo)* is a set of *los* aggregated on the fly by a software system during the retrieval phase. The logic on how to combine the *los* is stored in the digital library, in the metadata or elsewhere, as part of provenance. Compound learning objects are packaged into “*learning modules*”.

#### 11.3.1 5S PERSPECTIVE ON EDUCATIONAL DL

**Society:** Educational DL stakeholders include Instructors, Researchers, and Students. Each stakeholder has one or many *role(s)* in the DL such as:

- Researchers: users in this role are engaged in discovering, organizing, integrating, synthesizing, applying, and sharing knowledge, including to support learning.
- Authors: users in this role are engaged in creating digital educational resources; this group comprises instructors, researchers, and sometimes students. Most educational DLs divide resources into *standalone* “modules” to increase the reuse of educational resources. Each module typically talks about a specific concept. This is an application of the “High cohesion low coupling pattern” of software engineering.
- Reviewers: assess the quality of educational resources; they are domain experts and may be instructors or researchers. In most situations reviewers also are authors and have the ability to directly modify (e.g., edit) a resource to improve its quality.
- Learners: members of this category are the main ones undergoing a change in the state of their knowledge. They are the final consumers of the digital resources made available in EDLs. Sometimes those resources are produced as a result of the efforts of one or more of: researchers, authors, and reviewers. Sometimes they identify these

### 310 11. EDUCATION

resources on their own. Sometimes these resources are identified by teachers, who may recommend them for informal learning, or who may list them in a syllabus and cover them in a course or larger educational program.

- Instructors: lead a tutorial, course, or class as part of a formal educational activity aimed to help learners who are interested in education.
- Teachers: are instructors or others involved in less formal educational activities. They use or reuse educational resources for teaching purposes. They can use the entire resource or integrate a portion of it into an existing material. Sometimes students or others, like mentors, can perform this task as well.

**User data model in Educational DL:** A user model allows *personalization* of the user experience through “user profiling”, which makes use of, and may modify, a user model..

**Definition 11.6** A **user model** is a tuple  $(D, R, A)$ , where  $D$  is a set of information related to user description such as name, address, and experience.  $R$  a list of all the roles associated with the user.  $A$  is a record of user’s activities, that includes the type of activity (creation, edition, download, etc.) along with references to the resources involved. See Figure 11.1.

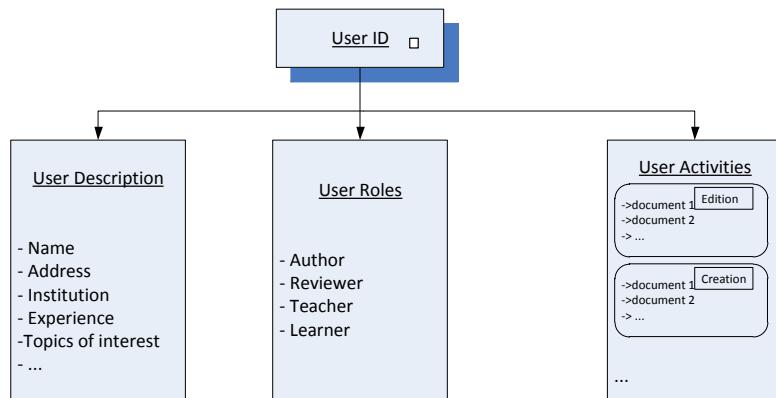


Figure 11.1: User Model

Personalization aims to automatically recommend resources (at an appropriate time and in an appropriate setting) to the user by analyzing metadata information and/or user

### 11.3. FORMALIZATION: EDUCATIONAL DL 311

behavior. *Metadata analysis* can lead to one example of static recommendation. Here the system tries to find resources with metadata information matching what is in a user model, for example suggesting all the documents about the user's topics of interest. A *behavior model* builds on a study of user activity and normally uses machine-learning methods to find useful patterns in a log of the user's behavior [426]. In this approach, the system records items visited by the user and applies some heuristics (e.g., time spent on the resources, number of hits, etc.) to mark them as interesting or not interesting. Then machine-learning techniques are used to find items similar to the "interesting" resources. The machine learning techniques include [426]:

- content-based filtering: computing resource-to-resource similarity; in the case of text documents the "cosine similarity" is frequently used to measure the similarity;
- collaborative filtering: looking at item reviews (e.g., tags, comments, and ratings) by the other members of the online community;
- heuristic-based recommendation: analyzing semantic relationships between resources;
- composite filtering: combining some of the above approaches and assigning a weight to each.

**Structures:** The IEEE Learning Technology Standards Committee published the *Learning Object Metadata (LOM)* standard to "specify the syntax and semantics of Learning Object Metadata, as well as the attributes required to fully describe a Learning Object" [393]. Some of the goals followed by LOM include [393]:

- To enable learners or instructors to search, evaluate, acquire, and utilize LOs.
- To ease the sharing, exchange and interoperability of LOs across multiple learning systems.
- To enable the development of learning objects in units that can be combined and decomposed in meaningful ways.
- To enable computer agents to automatically and dynamically compose personalized lessons for an individual learner.
- To complement the direct work on standards that are focused on enabling multiple Learning Objects to work together within a open distributed learning environment.
- To enable, where desired, the documentation and recognition of the completion of existing or new learning & performance objectives associated with Learning Objects.
- To enable a strong and growing economy for Learning Objects that supports and sustains all forms of distribution: non-profit, not-for-profit, and for profit.

### 312 11. EDUCATION

- To enable education, training, and learning organizations, including government, public, and private, to express educational content and performance standards in a standardized format that is independent of the content itself.
- To provide researchers with standards that support the collection and sharing of comparable data concerning the applicability and effectiveness of Learning Objects.
- To define a standard that is simple yet extensible to multiple domains and jurisdictions so as to be most easily and broadly adopted and applied.
- To support necessary security and authentication for the distribution and use of Learning Objects.

LOM schema are hierarchically organized and describe LO in nine categories: General, Life Cycle, Meta-Metadata, Technical, Educational, Rights, Relation, Annotation, and Classification. LOM-based metadata schema are being used in MERLOT [410], a web portal providing instructors with a large collection of digital learning materials along with evaluations and instructions on how to use them.

NSF NSDL defined its own metadata schema to describe educational resources. It follows the Dublin Core (DC) standard and it is called NSDL\_DC. Dublin Core has two implementations: “Simple” and “Qualified” (with component refinements and encoding schemes). NSDL\_DC has been built up as a variant of Qualified Dublin Core. The idea behind the design of NSDL\_DC was to have a standardized and common approach to metadata among all the NSDL pathway projects. In addition, the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH), used by NSDL to collect metadata, expects Dublin Core as a required encoding schema, though other metadata formats can be used in addition. The minimal required elements in the NSDL\_DC schema include: Title, Identifier (URL/URI), Description, Subject (and/or keywords), Education Level, and Type.

Recently, JSON (JavaScript Object Notation) has emerged as an alternative to XML for structured data in a repository [135]. Like XML, JSON aims to structure data stored in a digital library. JSON is being used to describe data in Freebase [66], a collaborative database project with a faceted web user interface. Unlike resources metadata that can be reused and transferred between different systems using standardized protocol, user models still lack reusability and interoperability in the majority of cases. Most DL systems represent user information and data in proprietary formats; as a consequence, they can only provide limited structured (user model) answers to other systems.

In [102] Carmagnola et al. listed systems that provided structured user model information: JSON and UserML (an XML-based language) are among the languages used. In addition, Google, Yahoo!, and MySpace, grouped in the OpenSocial<sup>2</sup> foundation, have constructed APIs to help developers to build social applications that access and share users' data from different websites.

<sup>2</sup><http://docs.opensocial.org/display/OS/Home>

### 11.3. FORMALIZATION: EDUCATIONAL DL 313

**Spaces:** The Vector Space model is generally used to represent user model data and is suitable to compute resource-resource similarity. It usually represents a user model as a “feature vector” with features including user interests, activities (browsing, downloading, authoring, and more), experience, etc.

**Streams:** Educational DLs manage streams of characters, pixels, audio, and video. These can describe not only content, but also logs of user actions, RSS-type releases of announcements or other information, and flows of assessment data.

**Scenarios:** Users’ roles define how users interact with the DL. Educators typically use a digital library to find educational materials that can be used during learning activities. They can just browse or search the collection – or based on their rights and permissions they can create, review, edit, or download items. Commenting, rating, and tagging digital resources also are common scenarios that added value to the digital objects. On the other hand, the system can recommend user resources that are relevant in regards to a user’s activities within the DL.

#### 11.3.2 FEDERATED SEARCH AND HARVESTING OF METADATA

Federated search and harvesting are two important mechanisms in DLs, and particularly in EDLs. They provide the users with a large amount of educational resources from different sources. Federated search allows the user to search multiple databases from a single entry point. Here any query is run against the system’s database; when desired (or as a default in many cases), that also leads to processing of the query against the data from other systems. One challenge of federated search is to select a set of systems likely to provide relevant answers. Metadata harvesting consists of gathering copies of metadata from other sites and storing them locally. To ensure interoperability between systems, the harvester and the data providers have to use the same protocol, and metadata must conform to a specified schema. The NSDL uses the OAI-PMH protocol to harvest metadata and its own metadata schema to validate harvested data. All NSDL data providers have to comply with that schema.

#### 11.3.3 CLASSIFICATION

One criterion that has been used to classify EDLs is the nature of the content stored by the system. That is whether the EDL stores only metadata or LOs or both [423]. EDLs are thus divided between:

- *portals*: systems that only store resources’ metadata and have a link to the actual location of the resource. AlgoViz is an example of a portal system.
- *warehouse*: systems storing all (actual) content locally. The TRAKLA [411] learning environment is an example of a warehouse system.

### 314 11. EDUCATION

- *hybrid*: systems are the combination of the two above systems. Ensemble [176] is an example of a hybrid system.

Another way to classify EDLs is to group them by the main educational activities they support, in addition to core EDL services. These activities include authoring materials, grading assignments, etc.

#### Authoring tools (ATs)

With the democratization enabled by the personal computer, the need for more digital education resources has increased in all fields. Before the Internet boom, most electronic resources were delivered via CD-ROM (inside a hard copy book). The goals of early authoring tools included to provide instructors (with limited programming skills) a platform to easily and rapidly create their class materials. Some of the most used authoring tools included HyperCard, Macromedia Authorware, Dreamweaver, Microsoft Word, and Front-Page. Brusilovsky et al. developed the InterBook system [83] using Microsoft Word as an authoring tool to create *adaptive* electronic textbooks; the document are then converted into HTML files. Authoring tools also have been widely used to make Intelligent Tutoring Systems, e.g., InterBook [83] and EDUCA [90]. In most cases, instructors use ATs to create learning materials, and to specify the relationship between these materials, while a learner or instructor consumes the information provided by the system. Some systems, on the other hand, involve the learner in the learning material creation process. For example in EDUCA [90], all learners have a user profile (feature vector) containing data like uploaded resources, web sites visited, grades, etc. When the student visits an external web page while using the system, information from the page is added to the learner profile, and the system engages in text mining so that suitable information can enhance the content of the web page in the learning materials.

**Example of Authoring Tool:** Connexions [39] is a web-based system for authoring and sharing educational materials. Resources are organized within Connexions as small modules that can be easily exported (as PDF files, text files, or web pages) and integrated into existing class materials. They also can be combined to make new class materials. Connexions brings together a community of educational resource authors (faculty members, industry professionals, and students) in order for them to create new materials in a collaborative way. This online community also peer-reviews the resources, thus enhancing the overall quality of the digital content in Connexions. Resources in Connexions are structured using an XML schema and are stored in a central repository. Metadata information (module title, authors, keywords or tags, and the relations with other modules) related to each module is stored in a database that enhances the search and retrieval of resources. Connexions implements the standard metadata tags from the Dublin Core Initiative.

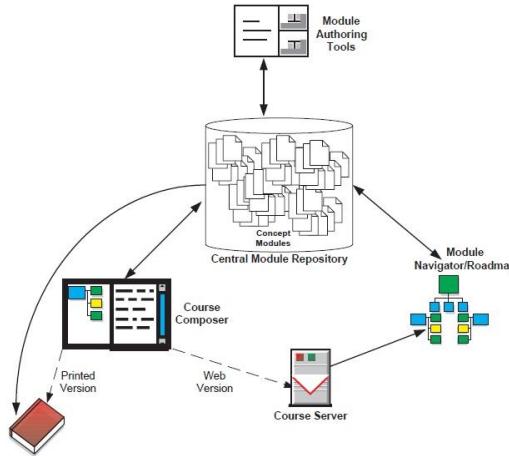


Figure 11.2: Connexions architecture [39]

### Assessment tools

Assessment activities aim to verify that the learners meet the learning objectives of a lesson or other educational activity. With the development of computer-aided learning, several automated assessment systems have been built. Most assessment systems rely on a *question bank*. A question bank is a database of questions and answers in which questions can be aggregated to form tests and can be exported and shared between different learning environments.

**Example of Automated Assessment Systems in Computer Science Education:** TRAKLA2 [411] is a learning environment comprising a large collection of Algorithm Visualization simulation exercises with automatic feedback (grading) built at the Helsinki University of Technology in Finland. It is a web-based system widely used to teach Algorithm and Data Structures courses. Exercises are grouped using a taxonomy of algorithm-related topics (sorting algorithms, Graph algorithm, hashing, etc.). The system allows the instructor to include some course notes. Security and the confidentiality is enforced by an authentication module and there are two roles available for the users: a “student role” and an “instructor role”. A database stores user information, including submissions, grades, etc., in addition to information about courses and exercises (e.g., deadlines, maximum points, grade limits). Every instance of a parameterized exercise is launched with a model solution, and TRAKLA provides some statistics about student performance. Exercise-related metadata is stored into a database and can be exported using XML and JSON schemas. TRAKLA exercises can be retrieved using the oEmbed<sup>3</sup> specification format as defined in [329] and be embedded as standalone applications in any hypertext media as a Java

<sup>3</sup><http://www.oembed.com/>

### 316 11. EDUCATION

Applet. oEmbed is an API, allowing resources from a website to be embedded on third party sites. It is one of the techniques that can be used to create Open Compound Digital Learning Objects.

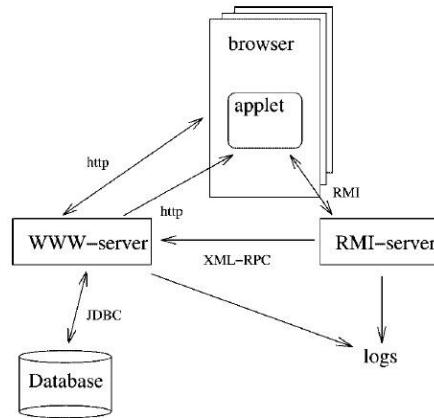


Figure 11.3: TRAKLA2 architecture [411]

## 11.4 CASE STUDIES

### 11.4.1 ALGOVIZ

Algoviz is a web portal for algorithm visualization (AV). It is driven by *Drupal*<sup>4</sup>, a Content Management System. Algoviz comprises a catalog of more than 500 AVs, and a large bibliography of related research literature. It aims to provide instructors with information about AV availability and usage.

**Society in AlgoViz** comprises: AV developers, instructors looking for AVs to use as class materiala,s and any other AV user. Algoviz allows all the members of this online community to add qualitative information on top of that which is delivered by a simple AV catalog. This information adds more value to the content and is the result of direct interaction and collaboration between community members. AlgoViz implements several modules and tools to support such interactions, including a forum. In addition, users can comment, rate, or tag resources and also share their experiences on how to make the best use of content, through what is called a “field report”.

**Structures in AlgoViz** are limited since it is a portal-type of EDL; thus it does not store LOs locally. A relational database management system stores information about all the

<sup>4</sup><http://drupal.org/>

## 11.4. CASE STUDIES 317

resources, the users, and all the events occurring within the portal. AlgoViz uses taxonomies to help organize AVs, with each taxonomy term coming from a controlled vocabulary. One of the many advantages of using a taxonomy is it makes easier for the indexer to locate the key feature of a resource. This is materialized through “faceted search and browsing” in AlgoViz see Figure 11.4. Metadata are formated following two standards; the Dublin Core (DC) Metadata Standard and the NSDL Metadata Standard. This allows Algoviz’s resources to be harvested and validated by any system following DC standards and by the NSDL or any NSDL partner (Ensemble).

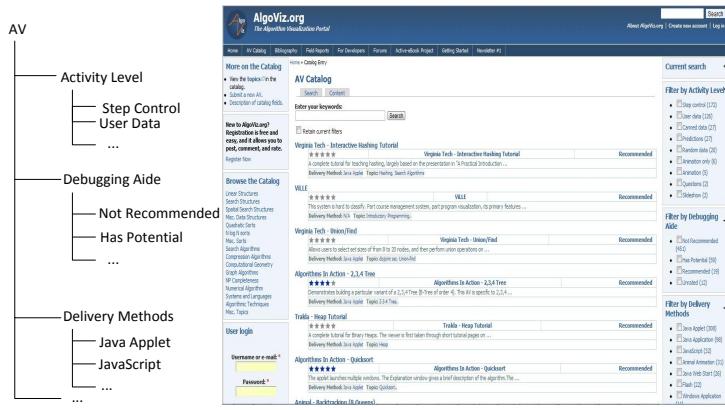


Figure 11.4: Algoviz’s taxonomy organization translated in faceted browsing

**Spaces** relate to the 2D portrayal used in most AVs. Further, resources in Algoviz are indexed using Solr<sup>5</sup>, an “open source enterprise search platform from the Apache Lucene project”. It uses a combination of a Boolean Model and a Vector Space Model to represent documents. The Solr index consist of documents, each document having several fields containing resources’ metadata information.

**Scenarios** support the AlgoViz community. Users come to AlgoViz not only to find AVs, but also to provide feedback on AVs they have tested and used. AV developers can create entries in the catalog for the artifacts they have developed. In AlgoViz, users can interact directly with each other through a forum, and find related research literature.

**Streams** can be used to describe the forum communications. Further, catalog entries in Algoviz provide not only textual information about AVs but also screenshots of the AVs and in some cases a short video describing the artifact.

<sup>5</sup><http://lucene.apache.org/solr/>

## 318 11. EDUCATION

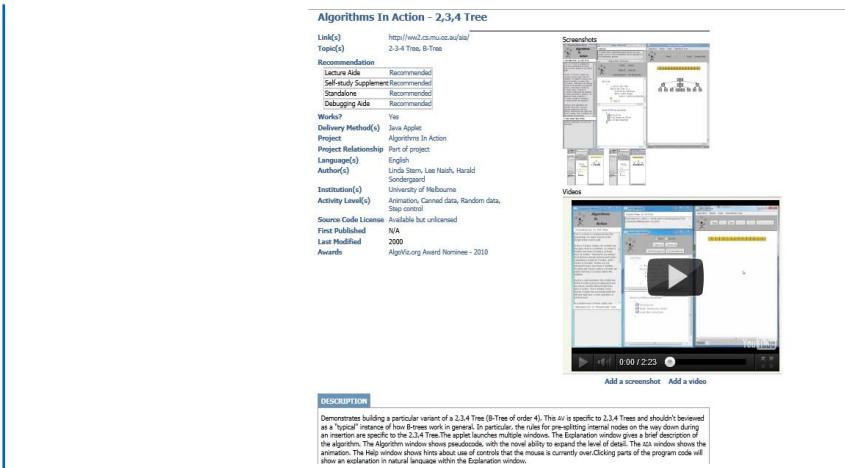


Figure 11.5: A catalog entry in AlgoViz

### 11.4.2 ENSEMBLE

Ensemble is an NSDL Pathways project working to establish a national, distributed digital library for computing education. The project is building a distributed portal providing access to a broad range of existing educational resources for computing while preserving the collections and their associated curation processes. The developers want to encourage contribution, use, reuse, review, and evaluation of educational materials at multiple levels of granularity and seek to support the full range of computing education communities including computer science, computer engineering, software engineering, information science, information systems and information technology as well as other areas often called computing + X, or X informatics.

**Society:** Ensemble defines different user roles to manage the operation of the Ensemble site. For example, administrators manage the site-wide configuration and service upgrade, and group managers handle the things related to a group among the Ensemble communities.

**Structures:** Ensemble acts as data harvester and also data provider. All the metadata in Ensemble has two formats, one is Dublin Core metadata and the other one is NSDL Metadata. All the metadata harvested from other DLs are cataloged and represent in a user readable record page in Drupal, that also is accessible to the general user. Each record page contains text, links, labels, and graphs in the hierarchies structure.

**Spaces:** All the contents in Ensemble are indexed using Solr. Ensemble uses the functionalities provided by Solr to provide faceted search and supports searching with browsing capabilities.

**Scenarios:** Users can find educational materials in Ensemble through browsing and searching services. They can create groups in Ensemble and invite others with similar interests to join those communities. They also can contribute their educational content and add to the Ensemble user contributions collection.

**Streams:** Computing educational contents in Ensemble include textual information (HTML, PDF, Word, and PowerPoint), screenshots of educational tools, and a short introductory video describing educational tools.

## 11.5 SUMMARY

Like many other digital library implementations, educational digital libraries are becoming more complex systems managing an increasing amount of digital resources. They should benefit from a formal model to rely on for design and implementation. Most of the models and concepts proposed in the field of EDLs are based on lessons learned and best practices. The 5S framework is a robust and general framework, backed by a strong theoretical work, and has been used to design and build several DL applications. In this chapter we presented how EDLs extend minimal DLs in order to provide services and tools that will satisfy the educational community information needs. We studied several examples of systems and showed how the 5S framework can be used to fully describe them.

An Educational DL was defined by the UNESCO Institute for Information Technologies in Education working group as an “*environment bringing together collections, services, and people to support the full cycle of creation, dissemination, discussion, collaboration, use, new authoring, and preservation of data, information, and knowledge [305]*”. Figure 11.6 presents a concept map based on this definition, allowing us to group related concepts under the umbrella of each element of the 5S framework.

## 11.6 EXERCISES AND PROJECTS

1. Besides the content, what most differentiates an educational digital library from other digital libraries?

320 11. EDUCATION

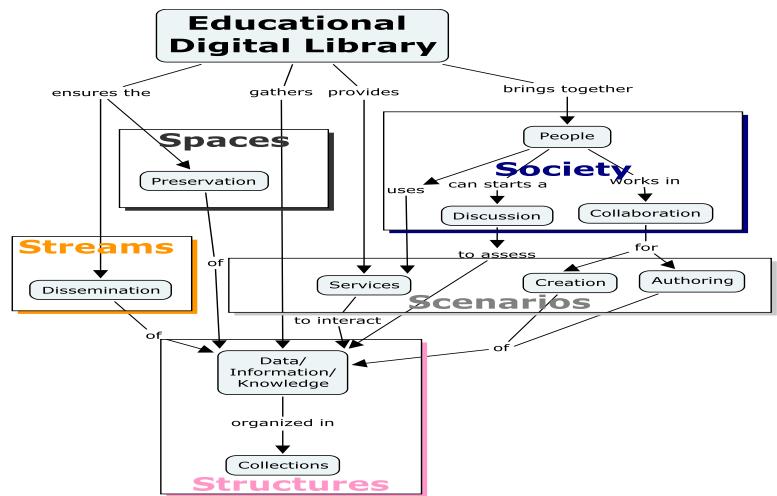


Figure 11.6: EDL Concept Map

## CHAPTER 12

# Bioinformatics, Scientific, and Simulation Digital Libraries

by Jonathan Leidig, Spencer Lee, and Sung Hee Park

*Abstract:* Digital libraries have historically been used in publication, cultural heritage, and document contexts. Recent trends in scientific fields (e.g., toward e-science) have provided impetus towards the generation of scientific digital libraries. Content producers such as high energy physics instruments and high performance computing simulation systems generate continuous streams of content at a large scale. Research groups in scientific fields require management of summarized datasets, experimentation environment, and findings. Applications from bioinformatics, scientific algorithms, and modeling and simulation contexts provide a representative sample of these research groups. Formal definitions of content, users, user requirements, generic services, and science-specific services are produced for each context. The definitions are used to evaluate the coverage and performance of services; provide interoperability between content, systems, and services; and serve as a basis for a service registry. Three digital library implementations serve as case studies from each context.

## 12.1 INTRODUCTION

Scientists and research organizations produce enormous quantities of digital content through experimentation. Sensors and instruments produce continuous streams of data, often formatted into collections composed of thousands of files. Large-scale scientific applications executed on high performance computing resources currently produce several petabytes of output for single experiments (e.g., climate modeling and bioinformatics). These research organizations are typically comprised of computer scientists, mathematicians, physicists, statisticians, and multiple domain experts. Scientific research groups rarely have expertise or collaborators in the information sciences. Managing large quantities of scientific content is increasingly being identified as a problem by content producers. Unfortunately, current practices for many of the largest data-producing applications existing in academia and national laboratories are inept in automating the management of content and metadata. In many cases, the only datasets archived are input configuration files and the only metadata tracked is related to application profiling in order to improve the scheduling

### **322 12. BIOINFORMATICS, SCIENTIFIC, AND SIMULATION DIGITAL LIBRARIES**

and performance of computing resources (e.g., expected memory and computation resources required by an important application).

A new class of digital libraries are needed to support scientific research. Due to the size of scientific datasets, collaborator access, and property ownership, actual data must be stored in a distributed fashion. A scientific DL may be used to federate multiple data storage systems using metadata in locating and retrieving individual datasets. A parallel body of work exists in linked data approaches. Federation across multiple domains, production systems, and research groups requires ontologies or domain models in order to describe content using appropriate context-specific metadata. A scientific digital library also requires connections to external components in an infrastructure. Research environments commonly include user interfaces, high-performance computing resources, instruments or simulation systems, analyses, visualizations, and data mining components. Content collections may be formed at each stage of the experimental workflows. A simulation workflow might have collections of simulation models, underlying datasets, input configurations, simulation results, result summaries, analysis results, visualizations, human-drawn conclusions, and publications.

Services for managing scientific content require high degrees of automation. The first step in this process begins with assistance in developing metadata description sets (schemas) and describing infrastructural workflows. Ideal DL systems will then automatically capture content as it is produced in each stage of the experimentation workflow. Metadata values from each object may then be extracted and indexed. Traditional similarity measures are ineffective in ranking and searching scientific content due to a lack of full text and the presence of highly numeric content. Curation services are needed to archive or remove content from system storage. Communication brokers are required to transfer content to and from other infrastructure components. Additional services are needed for specific services required by individual user roles, as described in section 12.3.

Previous digital library efforts have been insufficient in automating typical experimentation workflows and supporting scientific content. Novel formalisms and service implementations demonstrate the opportunity to tailor traditional DLs to efficiently support science. This chapter will use concrete examples motivated by computational epidemiology systems, fingerprint analysis research, and modeling and simulation of extremely large networks. These examples are representative of bioinformatics, science, and simulation applications.

## **12.2 RELATED WORK**

(Below: Adapted from JCDL Doctoral Consortium)

Efforts to enhance scientific data management practices have been a major focus for eScience and cyberinfrastructure groups over the last several years. Several examples of scientific data management include earthquake simulation repositories [324], embedded

sensor network DLs [75], community earth systems [165], D4Science II [379], mathematical-based retrieval [695], chemistry systems [387], national research data plans [339], and science portals [440].

Numerous workflow management systems exist but are not tailored specifically for simulation workflows. Standard workflow systems, not tailored for simulation systems, include Kepler [399], Taverna [488], Triana [409], and Pegasus [150]. Computational epidemiology workflows consist of model design and software implementation by model developers as well as study design, input configuration design, simulation execution, result summarization, analysis execution, analyses gathering, publication, and policy decision making by public health researchers. The workflow may be defined in an ontology and used to define a minimal set of SimDL services.

(Below: Adapted from ECDL 2010)

Examples of efforts to manage scientific and simulation content can be seen in the following systems; NASA NVO<sup>1</sup>, science portals [440], earthquake simulation knowledge bases [324], embedded sensor network DLs [75], community earth systems [165], D4Science II [379], and workflow management systems such as Kepler [399]. In other infrastructures, SimDL might interface with standard workflow systems such as Kepler, Taverna [488], Triana [409], and Pegasus [150] instead of static, ontology described workflows.

(Below: Adapted from JCDL Doc Consort/TCDL Bulletin)

Currently, there is a significant lack of deployable digital library options for simulation research institutions. Existing digital libraries lack a means of communicating specifically with cyberinfrastructure components, provisioning numeric-based services, automatically constructing content metadata records, supporting simulation-based tasks, and allowing federation across simulation systems, models, and model versions.

## 12.3 FORMALISM

Formalisms defined in the 5S framework expose the differences in science-supporting DLs and traditional full-text or digital humanities-supporting DLs. Common services appearing in these classes of digital libraries have been defined previously and the formal definitions are reused here where possible. This class of simulation-supporting digital libraries is formalized through a compilation of user, content, services definitions.

(Below: Adapted from ECDL 2010 paper)

Previous efforts to define digital libraries have progressed towards a digital library reference model. Two such efforts include the 5S framework [254] and the DELOS Reference Model [105]. The initially proposed 5S framework consisted of formal descriptions of core DL functionality. The framework has since been extended through the addition of subsequent definitions tailored to describe aspects of digital libraries within a particular scope (e.g., content based image retrieval) [454]. Common services required by many DLs involve

<sup>1</sup>see [www.us-vo.org](http://www.us-vo.org)

### **324 12. BIOINFORMATICS, SCIENTIFIC, AND SIMULATION DIGITAL LIBRARIES**

indexing, searching, and browsing content as well as query and annotation processes [254]. Definitions of new services or aspects to DLs may build on top of existing definitions, and producing formal descriptions of digital libraries is facilitated through reuse of the existing set of definitions. Assumptions 1-3 below underlie the reasoning for expending efforts in formally describing the foundation for SimDL.

**Assumption 1:** *Simulation-supporting DLs can provide interoperability through UI generation, infrastructure functionality, and managing experiments.*

**Assumption 2:** *Formal descriptions of SimDL content, services, and users exist and can fully characterize this class of DLs.*

**Assumption 3:** *Formal descriptions of DL components and functionality may be leveraged to produce and deploy DL instances with the stated capabilities.*

Extending the 5S framework concentrates new definitions to describing functionality required by experiment supporting DL systems and not included in previous DL descriptions. The following set of 5S extending definitions describe the digital content supported by the informally described functionality and services of a scientific experiment supporting DL instance.

#### **12.3.1 WORKFLOWS, CONTENT, AND ONTOLOGIES DEFINITIONS**

##### **Workflow of models**

(Below: Adapted from TPDL:SimDL article) The typical simulation experiment workflow will require simulation model design and software implementation of the model as well as services to support tasks including input configuration generation, model execution, result harvesting, analysis execution, analysis harvesting, human-intensive review, and publication or documentation. This workflow can be described in an ontology and modified for tasks with alternative workflows. As an example, arbitrary data mining of the existing corpus of results will simply include result harvesting, analysis, and human-intensive review.

##### **Workflow of content** (Below: Adapted from ECDL 2010 paper)

These definitions build upon the set of previous 5S definitions (i.e., handles, streams, structures, digital objects, complex objects, and annotations).

*Definition 1:* A schema is a digital object of tuple  $sch = (h, sm, S)$  where

1.  $h \in H$ , where  $H$  is a set of universally unique handles (labels);
2.  $sm$  is a stream;
3.  $S$  is a structure that composes the schema into a specific format (e.g., XSD structure of elements and attributes of restricted values <sup>2</sup>).

<sup>2</sup>W3C - XSD

*Definition 2:* An *input configuration* specification matching an XSD schema is a tuple  $icfg = (h, sm, ELE, ATT)$  where

1.  $ELE$  is a set of XSD elements;
2.  $ATT$  is a set of XSD attribute values for an element  $ELE_i$ .

*Definition 3:* A *sub-configuration*, a subset of an input configuration, is a tuple  $sub - icfg = (h, sm, icfg, ELE, ATT)$  where

1.  $icfg$  conforms to an XSD schema and  $icfg \supseteq sub - icfg$ ;
2.  $ELE$  is a set of XSD elements where  $ELE \subseteq icfg$ 's  $ELE$ ;
3.  $ATT$  is a set of XSD attribute values for an element in  $ELE$ .

*Definition 4:* *Analysis* of a set of experiments is a complex object consisting of textual or numeric documents and images (e.g., plots and graphs) and is defined as a tuple  $ana = (h, SCDO = DO \cup SM, S, icfg)$  where

1.  $DO = do_1, do_2, \dots, do_n$ , where  $do_i$  is a digital object;
2.  $SM = sm_1, sm_2, \dots, sm_n$  is a set of streams;
3.  $S$  is a structure that composes the complex object  $cdo$  into its parts in  $SCDO$ ;
4.  $icfg$  is the input configuration and one-to-one mapping to a raw dataset.

*Definition 5:* An *experiment* is a complex object consisting of the full range of information constituting an experiment within a domain and is defined as a tuple  $exp = (h, SM, sch, icfg, ana, D, A)$  where

1.  $SM = sm_1, sm_2, \dots, sm_n$  is a set of streams;
2.  $D = d_1, d_2, \dots, d_n$  a set of additional documents, e.g., summary or publication;
3.  $A = an_1, an_2, \dots, an_n$  a set of annotations describing the overall experiment and individual digital documents.

Support for provenance investigations follows directly from these definitions of the structured workflow of scientific studies and simulation experiments. Allowing a digital library's users access to an entire provenance stream allows claims to be supported, organizes the existing body of previous work, and allows data mining of collections related to a simulation system. See Fig. 12.1 for an Open Provenance Model representation overview of the preceding definitions [441].

Federation of multiple heterogeneous collections is provided by leveraging the general one-to-one mapping between sets of items produced by each experimentation stage. The use

### 326 12. BIOINFORMATICS, SCIENTIFIC, AND SIMULATION DIGITAL LIBRARIES

of model schemas by SimDL allows digital library services to make use of contextual information, e.g., allow for model parameters to be queried in search functions. In computational epidemiology, connecting non-standardized models requires human-intensive collaboration between model builders aided by schemas and ontologies, falling outside of SimDL's initial scope of automated services. Users could conceivably test hypotheses across semantically linked systems, models, and datasets from different groups.

**Domain Ontologies** (Below: Adapted from JCDL:2011 Poster)

Three categories of ontologies are used in SimDL. Simulation experimentation consists of a workflow of tasks that can be modeled in an ontology. This category of ontology may be used by a coordinating component of an infrastructure to identify and compose a sequence of the minimal set of services required for a workflow. An example workflow might consist of producing an input configuration, validating inputs with a model schema,

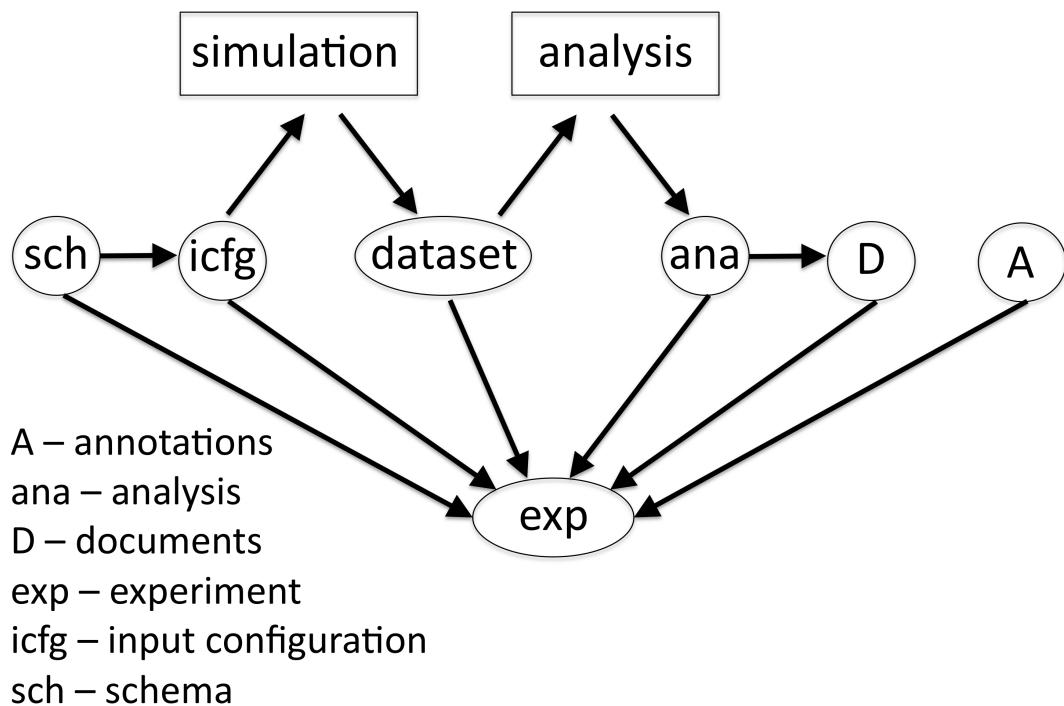


Figure 12.1: Context-specific simulation-based digital object provenance as represented by the Open Provenance Model.

submitting simulation requests, executing a simulation model, gathering results, submitting standard analysis requests, executing analyses, gathering analyses, and managing each stage of content.

The logic for DL services can be tailored for specific simulation contexts through a category of model describing ontologies. Another category of ontologies, a metaontology, is used to semantically link related model ontologies through term harmonization and organization. While DLs have been constructed for specific scientific domains, DL practices currently lack a methodology for customizing services at the granularity of individual simulation models through entire simulation domains. SimDL manages domain ontologies in standardized formats (e.g., OWL and RDF) and uses the ontologies to customize internal DL services for a domain.

Developing an ontology for a scientific domain is a major, complex undertaking involving multiple modeling and simulation groups. The human-intensive developmental process is eased by a SimDL ontology pre-processing generation service. The ontology generation service parses structured simulation content, e.g., input and output files, and presents suggested ontology terms for human-revision to a domain expert modeler that is familiar with the simulation model. The domain expert then may revise the ontology and provide a URL namespace describing each term. The ontology producing service currently parses XML, RDF, text, and XML Schema files for terms and may produce an ontology in OWL, RDF, or XML Schema format. Aggregating domain ontologies from multiple simulation models is conducted through a SimDL service to allow a human-intensive graphical pairing of semantically related terms.

Domain ontologies may be used to tailor services for specific models within a model-independent framework. As presented in [377], domain ontologies may be used to expose simulation parameters through a generic interface, structure the organization of content, and provide domain-specific metadata. SimDL uses the pre-processing ontology service to assist experts in generating domain-specific metadata description sets. The metadata schemas are then used to guide the automatic production of searchable metadata records for simulation content. As input and output files are produced, objects are parsed with document-specific search functions to produce metadata records as defined by the document-specific model ontology. DL search services are constructed in SimDL to use model-specific and aggregated ontology mappings to provide search between models with the guarantee that generated records will be properly formatted.

Inclusion of domain terms with a known semantic meaning, as defined in the modeling groups URL namespace, into ontology-using DL services, allows for customization while preserving the generality of service software. As an example, constructing a user interface (UI) to search, browse, and acquire inputs to a new simulation model is extremely rapid after developing a UI service to parse a model-specific input-file ontology and present the content in a generic UI. Although the generic-looking interface will not be aesthetically

### **328 12. BIOINFORMATICS, SCIENTIFIC, AND SIMULATION DIGITAL LIBRARIES**

appealing and efficient in maximizing tab space in a browser, generality is provided by encapsulating domain information in an ontology and constructing generic DL services to make use of domain information. We are currently conducting user studies to evaluate the usability of the ontology generation tools and designing quantification measurements on the effectiveness of ontology-using, simulation-specific services.

(Below: Adapted from TPDL SimDL paper)

Model ontologies are used by SimDL to generate a metadata schema for the digital objects related to the simulation model described by the ontology. Support for domain specific services, such as semantic searching between simulation models, may be tailored through ontologies that describe individual simulation models and ontologies aggregated to the domain level. We have developed a semi-automatic process for generating epidemiology-model ontologies based on the structured inputs and outputs of the simulation software. We have generated multiple model-ontologies and used the ontologies for organizing storage, presenting a model-specific interface to users, and providing model-specific content discovery [377]. The ontologies, at various levels of domain and model granularity, are used to customize services, e.g., model-specific search. Ontologies are defined as digital objects of tuple  $ont = (h, sm, S, C_i \subset C, T)$ , where

1.  $h \in H$ , where  $H$  is a set of universally unique handles (labels);
2.  $sm$  is a stream;
3.  $S$  is a structure that composes the schema into a specific format (e.g., OWL);
4.  $C_i$  represents a set of digital objects described by the ontology;
5.  $T$  is a set of ontological terms and term relationships.

**Scientific Images, Plots, and Graphs** (Below: Adapted from TPDL: Fingerprint article)

FBI's Integrated Automated Fingerprint Identification System (IAFIS) is large fingerprint management system, supporting search capabilities against both latent and ten prints, storing electronic images, and electronically exchanging fingerprints. However, it does not support a series of services for experiment digital libraries such as experiment setting, distorting, plotting, and visualizing. The Universal Latent Workstation (ULW) is the first latent workstation supporting interoperability and sharing latent identification services with local and state authorities, and with the FBI IAFIS, all with a single encoding.

[509] proposed an experiment management tool, *Eva*, for evaluating descriptors in content-base image retrieval, providing image descriptors, and image management, to run comparative experiments. This tool has stimulated the development of our holistic DL experiment framework. Previous work also supported scientific communities in a web-based integration framework [657].

Fingerprint analysis has been challenged by various distortions such as merged prints, pressured impressions, humidity on fingertips, partial prints, or simultaneous prints. Distortions are likely to affect minutiae extraction quality, ridge tracing quality, matching scores, and image quality. The Analysis, Comparison, Evaluation and Verification (ACE-V) and Scientific Working Group on Friction Ridge Analysis, Study and Technology (SWGFAST) groups (see swgfast.org) have worked on fingerprint analysis. [489] proposed a multiscale directional operator and morphological tools for reconnecting broken ridges in fingerprint images. [297] proposed singular point detection.

From the object perspective in very large digital libraries, [353] proposed a solution to integrate four different very-large fingerprint digital libraries. A proposed compound object (CO) scheme uses the 5S framework, modeling different types of objects found in those DLs, to allow uniform use in an integrated DL. Our work is focused on designing a DL framework, from a services perspective, to deliver analytical results of an experiment that integrates related services designed by different researchers.

### 12.3.2 USER ROLE AND TASK DEFINITIONS

**User modeling** (Below: Adapted from ECDL 2010 paper)

Definitions of content, users, and tasks for user roles informally describe the necessary services in a DL.

The following communities maintain inter-community relationships through activities as described through a set of community-specific digital library scenarios.

1. *Tool builder*: formally define a proposed DL; locate, reuse, and assemble existing components; generate novel components; deploy a DL instance; integrate with a simulation or digitized experimentation infrastructure
2. *System and DL administrators*: set data management policies; clean datasets or metadata; curate content; manage accounts; evaluate existing components
3. *Related systems*: submission and retrieval with experimentation applications and high-performance computing architectures; analysis of submissions and retrieval with analyst oriented software; validation of inputs
4. *Study designer*: generate new model schemas and transform to updated versions; enter or load input configurations; query, browse, and load sub-configurations; save configurations and sub-configurations; validate configurations; submit configurations for execution; monitor experiment progress
5. *Analyst*: submit new analysis requests; view automated or requested analyses
6. *Annotators*: mark annotations on streams of content or individual documents
7. *Explorer*: query and browse collections or experiment streams of documents

### **330 12. BIOINFORMATICS, SCIENTIFIC, AND SIMULATION DIGITAL LIBRARIES**

**User Collaboration and Cooperation** Simulation digital libraries support typical user communities (e.g., tool builders, system administrators, annotators, explorers, and external systems) as well as previously undefined communities (e.g., study designers, analysts, and simulation and analysis brokers).

The definitions for experiment-related digital objects, services, and users provides tool builders of proposed digital libraries a mechanism for collecting requirements, describing the desired DL, identifying a set of reusable components from a well described existing component pool, and guiding development of customized functions.

Collaborations between computational epidemiology institutes based in the US and Switzerland, each with years of experience in simulation studies, have identified the need for SimDL. SimDL aims to provide a generic, extensible component-based digital library. The reliance on schemas allows SimDL instances to support simulation applications for an unrestricted set of domains and contexts that have structured input requirements and a waiting simulation launching broker. Non-simulation scientific content also may be handled by SimDL instances sans the simulation submission component (e.g., wet labs or instruments producing digital data points along with environmental input conditions).

SimDL was designed out of a need to automate the management of information produced by multiple simulation applications at each institute. Tool builders desired fully automated deployment of a basic DL instance and low amounts of continued maintenance. The management of scientific data requires policy decisions for data preservation, curation, and distribution mechanisms within the digital library. Shared access to a digital library hosted by one institution allows privileged experts at multiple locations to directly and indirectly collaborate, communicate, cooperate, and coordinate research efforts through the DL. The automation of conducting simulation studies is accomplished by the integration of SimDL into an existing simulation infrastructure. Until scientific digital libraries become mainstream in the simulation community, DLs will likely continue to be integrated into a pre-established experimentation process and existing software system. Customization of the simulation-launching component may be required to transmit inputs to simulation software (e.g., transmitting XML input files to a waiting computation request broker). The generic user interface, shown in Fig. 12.2 and implemented with the Google Web Toolkit<sup>1</sup>, is constructed by displaying a user selected model. The parameters to the model are then displayed along with tabs to load existing inputs or switch roles (e.g., analyst).

Through SimDL, various types of users can interact in one place. Multiple experts are required to conduct complex simulation-based research. Similarly, multiple user roles exist for simulation-based digital libraries which support collaboration, participation, and privacy as defined in [307]. In SimDL, collaboration is assisted by capturing and exchanging information produced from complex tasks performed by users at different stages of the experimentation process. The collaboration requires multiple tools to answer research ques-

<sup>1</sup><http://code.google.com/webtoolkit>

tions by translating a query or simulation request into successive units of work. Although the submission of simulation results to an analysis system may be automated, participation from users is required to launch new simulations, perform customized analysis, draw conclusions, and annotate streams of content. Users with similar and differing roles work on the same content, further existing work between and within streams of content, and switch between consuming digital objects from previous experimental stages to providing content for users in successive stages. Practical consideration may restrict the number of users with a simulation launching role to reduce the stress on computational and data storage resources. Users under most roles interact with a digital library's content and are presented with the ability to interact with multiple schemas and versions of schemas across all available simulation applications, domains, and contexts within a single, generic interface. By selecting tabs in the UI, users with a particular role may transition seamlessly to

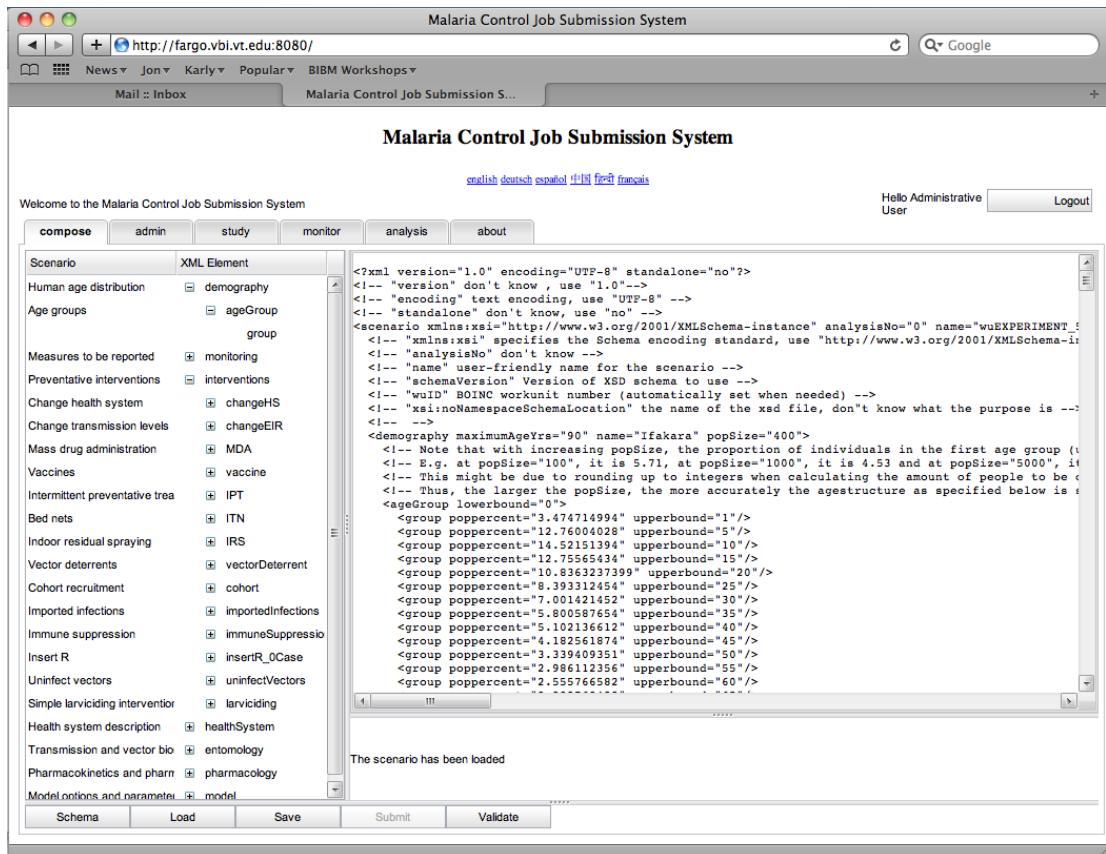


Figure 12.2: SimDL's automated, generic web UI snapshot displaying a model's schema.

### **332 12. BIOINFORMATICS, SCIENTIFIC, AND SIMULATION DIGITAL LIBRARIES**

another role (e.g., study designers switching to analysts when viewing the results from a launched simulation).

Tool builders are provided with reusable, formally defined components. The use of a schema automates the UI generation, database mapping, simulation launching, and collection management processes. Human involvement consists of developing a schema for a model and starting the generation process. For study designers, the digital library hides the complexity of launching and analyzing simulations. Experts in a field (e.g., mosquito vectors or population demographics) may upload high-quality sets of input parameters to be reused by users with less knowledge of a portion of an epidemiological model. The designer may reuse and modify existing sub-configurations, submit simulation requests, make use of data management services, and interact with the simulation infrastructure. Analysts are able to retrieve datasets and summaries of datasets along with the input conditions from which the results are derived. Automated tasks for analysts include the generation of plots and summary statistics. Explorers are able to query and retrieve content across the entire provenance trail for findings without having to interact with each simulation system component. Users with this role may discover existing content or identify a lack of previous simulations for a queried segment of the simulation system's multi-dimensional input space.

#### **12.3.3 SERVICE DEFINITIONS**

(Below: Adapted from ECDL: 2010 article)

Digital libraries have been non-uniform in the provision and implementation of services. The DL community benefits from formal definitions of existing and proposed digital libraries along with the services provided. Formal definitions of content promotes services for discovery, reuse, and provenance investigations of scientific data. Simulation related content includes a chain of staged data produced by successive steps in the experimental process. A simulation model's schema provides the domain context required to automate the generation of a model-specific DL interface with data management support. A minimal digital library for simulations must track provenance, manage content, organize input and output files, support commonly used HPC systems, and integrate simulation models into a workflow.

(Below: Adapted from TPDL: SimDL article)

We aim to formally define a suite of DL services to be implemented in SimDL and other simulation management systems. Formal definitions provide a means of explicitly specifying the requirements for services. Formal definitions allow for registered, existing services to be identified and reused in multiple DL implementations. Requirement identification and development may use formal definitions to guide efforts. Definitions may be used to prove sufficiency and completeness of services without specifying implementation decisions. With formal definitions of services, we envision a future registration system for simulation-specific services described by languages such as UDDI, OWL-S, and the 5S

framework. A service registry would be useful in rapidly prototyping DL service layers that support HPC infrastructures and scientific applications as described in [270, 299]. The following set of services are essential for a minimal simulation supporting digital library. Existing service definitions and notation are reused where possible but are mainly limited to text-based versions of simulation-specific services, e.g., generic indexing and search.

#### **Discovery and Dissemination**(Below: Adapted from TPDL: SimDL article)

Browsing scientific content is similar to browsing textual documents with the added assumption that documents will be displayed in a faceted design. SimDL has implemented faceted browsing for a clustered document space based on the simulation model source and the stage of the workflow producing each type of content as described by existing ontologies. Browsing requires as input an initial simulation facet,  $\text{anchor}_f$ , and potential links,  $\text{Hyptxt}_j$ . Browsing produces as output a set of digital objects,  $\{do_i : i \in I\}$ . The pre-condition for browsing is that the anchor must exist in the network of hyperlinks,  $\text{anchor} \in \text{Hyptxt}_j$ . The post-condition restricts the resulting digital objects to existing documents in the collection valid for  $\text{Hyptxt}_j$ ,  $\exists C \in \text{Coll} : \{do_i : i \in I\} \subseteq C$ .

Existing search functions are typically suited for full-text documents and textual metadata. Simulation content requires search over numerical metadata and summarized content. Search implementations relying on text-based metrics, such as term frequencies, thesauri, dictionaries, stemming, and term co-occurrence, break down in scientific digital libraries. However, queries may be segmented into  $k$  Boolean clauses for numerical metadata terms. In this type of querying system implemented for SimDL, a similarity score between a document and query is provided by a weighted summation over the  $k$  clauses as presented in [377]. The formalism for scientific search is identical to full-text search as a query ( $q$ ), collection ( $C_i$ ), and index ( $I_{C_i}$ ) are required to produce a set of weighted results,  $\{(do_i, w_{qk}) : k \in K\}$ . A detailed formal definition of the technical aspects used to implement a search service would be required to differentiate between full-text and numerical-based search mechanisms.

Discovery mechanisms are aided by model-specific ontologies for browsing and model-specific search. Content from successive simulation workflow stages are derived from preceding stages. Results from browsing and searching at one stage are linked through unique identifiers to each item's provenance stream. SimDL requires data fusion between ontology-specific collections with an attached richly defined ontology in addition to typical fusion between institutional repositories. Input configuration retrieval is specifically used to examine the search space of existing studies and to reuse portions of an input configuration when designing new simulations. Search over output results yields datasets supporting data mining and customized analysis.

The experimentation workflow and model ontologies allow for guided browsing of each collection from a specific model version. Customized analysis, publications, and related documents may involve comparisons between datasets and models. Collections in the later

### **334 12. BIOINFORMATICS, SCIENTIFIC, AND SIMULATION DIGITAL LIBRARIES**

stages of the experimental workflow retain provenance links back to underlying datasets and have a one-to-many mapping between collection and ontologies. Current versions of SimDL services and the generic UI provide suggested documents to users based on recent work and recommended content. The browsing design forces users to first select a model and stage of the experimental workflow for a faceted view of available documents.

Dissemination of scientific content consists of *exposing* content and providing the ability to *retrieve* specific content. *Exposing* is a composition of general services for *acquiring*, *cataloging*, *indexing* to support search functions, and *classifying* to support faceted browse functions. *Retrieval* consists of requesting a document,  $\exists C \in Coll : do_i \in C$ ; identifying handles for the requested document,  $h_i = do_i$ ; and provision of the document as output,  $\{do_i\}$ . The pre- and post-conditions for *retrieval* are  $\exists do_i, C : do_i \in C \wedge h_i = do_i$  and  $do_i = h_i$ , respectively. Exposing and retrieval are fully supported in our implementation of SimDL.

#### **Matching and Searching** (Below: Adapted from TPDL: Fingerprint article)

One algorithm for *matching* and *searching* attempts to use 3, 6, or 9-point triangles of high-quality minutiae locations to identify matches between two images as groups of minutiae are less susceptible to distortions. This matching algorithm stems from attempts to reduce the effects of small distortions on the identification of minutiae location and quality.

This process can be defined as a *binary* operation service  $f(do_i, do_j) = k$ ,  $k \in R$ , compared to a service such as rating and measuring which is a *unary* operation  $f(do_i) = k$ ,  $k \in R$ , where a real number  $k$  is a similarity score.

#### **Ranking** (Below: Adapted from JCDL: Short paper)

Search in SimDL is domain and model specific. Simple search allows users to filter content based on exact matches to input configuration parameters, metadata fields, and input or result parameters added to the metadata schema. Filtering involves selecting a schema and then metadata parameters of interest on which to filter. Exact values or a range of acceptable values for each parameter adds documents in a collection to the result set upon matches with the filter criteria. Thus, a novice user, desiring to reuse sub-configurations for a new experiment, selects a model, selects a node-level from the model-specific ontology, views and filters the set of sub-configurations archived at the specified node-level, and reuses an individual sub-configuration.

Ranked searches require the development of similarity distance metrics (*sm*) and a ranking system. The goal of a user's search is to discover a set of experiments that best fit given ranges or values of a collection's metadata terms. A user formulates a search query,  $q$ , of multiple clauses with Boolean 'and', 'or', and 'not' operations combining clauses. A clause,  $c_i$ , consists of a metadata parameter, a value or range of values, and clause descriptions (e.g., contains, equals, between). The clauses, in disjunctive normal form, are translated into a Boolean query and compared to documents in a collection. For a document,

each clause's value,  $v_i$ , is evaluated to '1' or '0' dependent on the document meeting the clause condition. Similarity scores for a document,  $d_i$ , are then produced by assuming each of the  $k$  clauses in a query have a weight of  $1/k$  and setting the total distance as  $sm(q, d_i) = \frac{\sum_{i=0}^k v_i}{k}$ . Advanced users may customize their query by assigning a clause weight,  $w_i$ , for each of the  $k$  query dimensions with a default of 1 and the summation of clause weights totaling  $k$ . The total distance is then calculated as  $sm(q, d_i) = \frac{\sum_{i=0}^k v_i * w_i}{k}$ . Typical distance functions based on natural language term frequencies are less useful than numeric metadata values for scientific data. Similarity metrics are currently in developmental stages as we attempt to integrate filtering, ranking, and annotation search.

**Collaboration** (Below: Adapted from TPDL: SimDL article) Collaboration services consist of tasks supporting multiple researchers at various stages of the simulation workflow. Scientific collaboration is a composition of generic and simulation-specific services, see Fig. 12.3. Collaboration consists of tasks for annotating, locating, rating, recommending, reviewing, and submitting content. Locating includes processes for user acquisition of content by browsing, searching, and retrieving. Submitting includes processes for indexing, managing, and disseminating content within a DL. *Collaboration* reuses generic services for annotating, rating, recommending, and reviewing; *locating* invokes simulation-specific services for browse, search, and retrieval; and *submitting* invokes generic services for indexing and disseminating. Generic versions of these services for text-based documents are defined in [250]. SimDL currently has implementations for all of the collaboration services except for rating. As all of the epidemiology content in a model-specific collection is produced by the same simulation model, ratings for individual documents are gratuitous.

(Below: Adapted from JCDL: Short paper)

The experimentation workflow assists in sharing and collaborating within a simulation model's user community. In designing new experiments, users may share, reuse, and recommend portions of existing input sub-configurations. In epidemiology, high quality population datasets are shared and reused as a highly recommended input sub-configuration. Novice users of a SimDL system might heavily reuse existing sub-configuration blocks generated by model experts. Advanced users, with detailed knowledge of the model, contribute by sharing detailed input sub-configurations. This approach encourages collaboration by experts in different fields (e.g., population modelers, disease modelers, entomologists, and public health officials). The provenance stream of content allows participation by researchers with different interests to contribute at different stages of the experimentation process. SimDL and its interface remove the need to receive training on how to generate inputs to each model, connect to backend resources, execute simulation or analyses code, or transfer data. Researchers' direct efforts are concentrated on retrieving staged content and performing non-automated activities, e.g., specialized analyses and publication efforts. As shown in Figure 12.4, many of SimDL's internal services make use of the collection structure that

### 336 12. BIOINFORMATICS, SCIENTIFIC, AND SIMULATION DIGITAL LIBRARIES

is organized through ontologies. SimDL's use of a generic interface provides a domain-free, model-exposing system.

#### Curation (Below: Adapted from TPDL: SimDL article)

Scientific content produced as output from HPC simulation systems may scale to infeasible storage requirements. Large simulation results datasets may be summarized or reproduced as needed. Content curation provides a means of establishing policies for content removal as well as arbitrary user-specified deletion. Simulation-based *curation* requires processes for identifying removable documents, removing documents, and either removing the document's index or defining a method of re-generating the document. As input, *curation* requires a set of document handles for arbitrary deletion,  $\{h_x, \dots, h_y\}$ ; a filter or query for pre-defined deletion,  $\{q, C_i\}$ ; and a specification for the type of curation removal,  $c_s$ . Specifications for documents that should remain indexed but physically removed for storage reclamation additionally require input for a regeneration method consisting of a soft-



Figure 12.3: 5S graph of collaboration and related services.

ware executable,  $sw_i$ , and input configuration,  $icfg$ . The output of *curation* is a modified collection,  $C_i \Rightarrow C'_i$ ; and altered index for the collection,  $I_{C_i} \Rightarrow I'_{C_i}$ . Pre-conditions for *curation* are  $\forall h_i \in \{h_x, \dots, h_y\} : \exists do_i, C_i : do_i \in C_i \wedge h_i = do_i : C_i \in Coll$ . Post-conditions for *curation* are *none* as removal criteria may exist without triggering document removal. SimDL's default policy is to maintain all information and relies on infrastructure designers to define processes for result summarization and dataset deletion.

#### Metadata Extraction (Below: Adapted from TPDL: SimDL article)

Content generated by the execution of simulation models must be automatically processed due to the volume of digital objects produced. Fortunately, metadata extraction need not require human-intensive efforts as ontologies describing a domain and type of object may be used to build a metadata description set for each type of content. Pairing the description set with a metadata extraction script supports automated, run-time metadata record harvesting. *Metadata extraction* for a specific type of document requires an ontology,

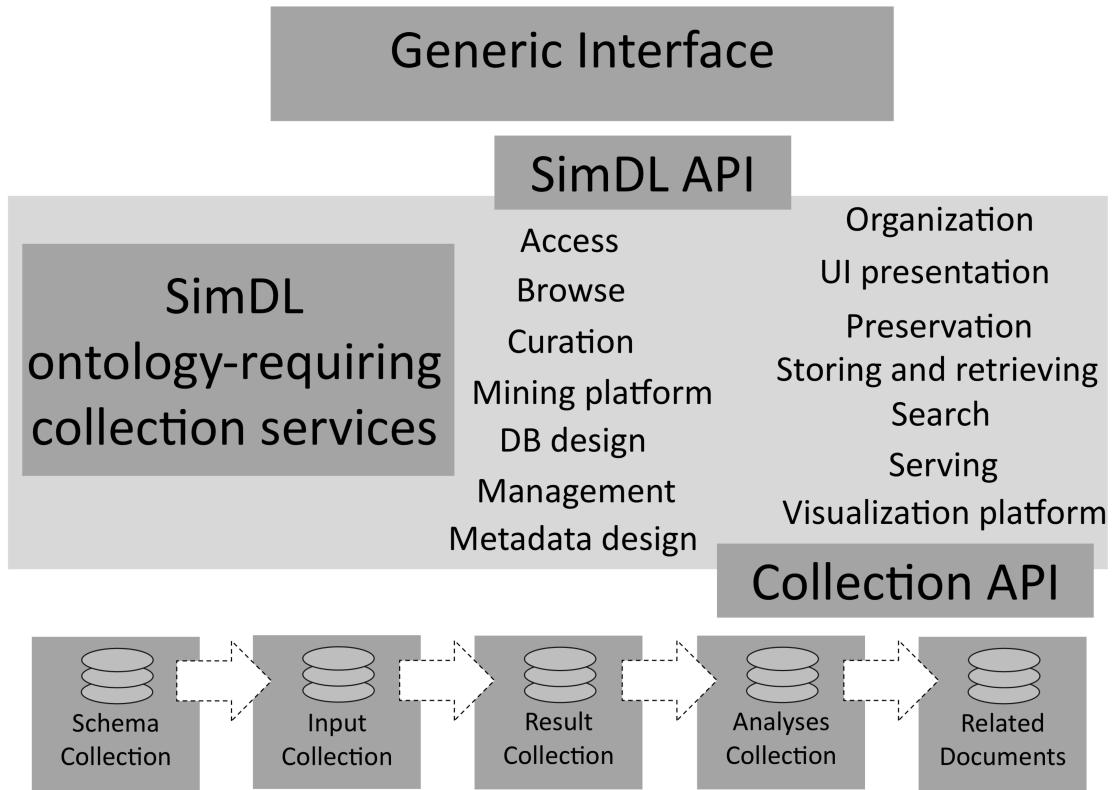


Figure 12.4: Ontology-requiring, model-independent services in SimDL.

### **338 12. BIOINFORMATICS, SCIENTIFIC, AND SIMULATION DIGITAL LIBRARIES**

*ont*; metadata description set,  $MD_s$ ; extraction script,  $ext_s$ ; and a document harvested by a content broker,  $do_i$ . The output of the extraction process is the inclusion of  $do_i$  in the collection index,  $I_{do_i}$ . The pre-conditions are an existing collection,  $C_i \in Coll$ , and script  $ext_s$  to extract content from  $do_i$  to  $I_{do_i}$  as described in  $MD_s$ . The post-condition is a modified collection index,  $I \Rightarrow I'$ . Metadata extraction in SimDL is handled by extraction scripts paired with a model ontology and are provided by simulation model developers. It is assumed that all input files or output files from a specific version of a simulation model may be processed by the same extraction script.

#### **Provenance Linking** (Below: Adapted from TPDL: SimDL article)

Simulation systems produce a stream of content at multiple stages including model schemas, input configurations, results, analyses, documentation, annotation, and publication. Preserving provenance information from findings and results is necessary for simulation researchers. Provenance is maintained by defining each type of content in a localized ontology and harvesting content collections for each stage of the experimentation process. Similar to indexing a document  $do_i$  in an index  $I_{do_i}$ , provenance information can be encapsulated in an index structure where two sets of documents,  $\{do_i, \dots, do_k\}$  and  $\{do_j, \dots, do_m\}$ , are indexed in  $P_{\{do_i, \dots, do_k\}, \{do_j, \dots, do_m\}}$ . Provenance streams may be defined by the tuple  $P = (h_{\{do_i, \dots, do_k\}}, C_i, h_{\{do_j, \dots, do_m\}}, C_j)$ , where

1.  $h_{\{do_i, \dots, do_k\}}$  represents handles for a preceding set or type of digital objects;
2.  $C_i$  is a collection in  $Coll$ ;
3.  $h_{\{do_j, \dots, do_m\}}$  represents handles for a successor set or type of digital objects;
4.  $C_j$  is a collection in  $Coll$ ;
5.  $\{do_i, \dots, do_k\} \in C_i$ ; and  $\{do_j, \dots, do_m\} \in C_j$ .

#### **Workflows** (Below: Adapted from TPDL: SimDL article)

SimDL interacts with this infrastructure's workflow management system, which is similar to other grid workflow processes, to communicate with other portions of the infrastructure. This communication interface is used to archive and manage input files, output files, analyses, plots, and publications. In other infrastructures, SimDL could interface with bioinformatics workflow systems such as Kepler [399], Taverna [488], Triana [409], and Pegasus [150]. Note that these systems exist to compose multiple stages of a workflow where each stage manages inputs, executes software, and organizes results. SimDL extends traditional workflow system functionality by using ontologies to provide simulation-specific services. A set of model-independent APIs and brokers supports the integration of SimDL with a single computational platform (e.g., grid or cloud), local institutional infrastructure (e.g., internal platform), or large-scale coordinated cyberinfrastructure (e.g., multiple, distributed platforms).

**Internationalization** (Below: Adapted from JCDL: Short article)

Scientific and simulation collaborations in a domain are often international efforts. The SimDL and generic interface pairing supports internationalization by using multiple multilingual glossaries. Each glossary provides multilingual translations used in user navigation of interface components. We rely on model builders to provide an XML Schema for each model and language pair. Domain ontologies and multilingual glossaries have previously been paired for searching and relating terms across documents in multiple languages [438]. SimDL uses domain ontologies and a multilingual glossary in the complementary process of UI presentation for content matching multilingual simulation model schemas. Our reliance on model builders for internationalization efforts mirrors the necessity in [438] of using a domain expert in the essential task of developing a classification ontology.

**Scientific Analysis and Experimentation** (Below: Adapted from TPDL: Fingerprint article)

To support a typical fingerprint algorithm analysis workflow, we have developed a DL services model and implemented a prototype instantiation. The workflow includes five stages: image harvesting, distortion image generation, algorithm execution, result harvesting, and algorithm performance analysis. With this model, researchers are able to investigate how an algorithm performs with synthetic, field-quality images. In particular, researchers are provided with an analysis framework that could be used to determine which image distortion parameters effect feature identification. The targeted workflow model framework pairs new algorithms with image collections to allow analysis on which image characteristics effect algorithm performance, as seen in Fig. 12.5. Previous work has defined the format for formally defining DL services that we will employ [250]. Formal definitions already exist for generic versions of several of our fingerprint-specific services, e.g., searching and visualizing [250]. The services in the following section are implemented in the prototype's initial workflow.

Each algorithm to be used in experimentation requires an algorithm-specific description of which outputs to process for analysis. The minutiae extraction algorithm requires analysis on the number, pixel location, and quality of identified minutiae. The ridge tracing algorithm requires quantification and analysis on ridge identification. The matching algorithm requires analysis to determine if the set of selected minutiae was sufficient in making a match. One required step to begin an experiment is to set up all the input parameters using the 'Selecting algorithm/images' service to choose which images and algorithms should be selected in an experiment run.

**Image Manipulation** (Below: Adapted from TPDL: Fingerprint article)

After high-quality real-world fingerprint images are submitted, a distortion generation algorithm takes a set of values for the ten distortion parameters used in [412]. Each distorted image then is added to the DL collection along with a link to the original image and the parameters used in the image generation. In environments with low amounts of disk storage

### 340 12. BIOINFORMATICS, SCIENTIFIC, AND SIMULATION DIGITAL LIBRARIES

available, the image generator may be used to generate distorted images on a just-in-time basis, and images may be removed after usage though their metadata or generation script is archived.

For the image processing service, there are two approaches to the extraction of minutiae and ridge features: *binarization* and *gray scale*. *Binarization* approaches typically use image processing techniques such as sharpening, histogram equalization, and enhancement, while *gray scale* approaches often exploit filtering with a Gabor filter to enhance gray scale fingerprint images. This service also can be seen as converting service, similar to the distortion service, and thus has the same 5S formalization [250]. See Table ?? for basic terms and definitions.

Informally, *distorting* and *image processing* take a digital object and produce a distorted version by changing its streams, structures, or structured streams as defined in the 5S framework [250], an alternative to the DELOS reference model<sup>3</sup>.

*Distorting/image processing* is a service defined as  $f : do_i \Rightarrow do_j$ , given a digital object  $do_i$ . The input and output structures for this service are  $do_i$  and  $do_j$ . The pre-condition and post-condition for this service are  $\exists C \in Coll : do_i \in C$  and  $\exists C \in Coll : do_j \in C$ .

<sup>3</sup>See [www.delos.info](http://www.delos.info)

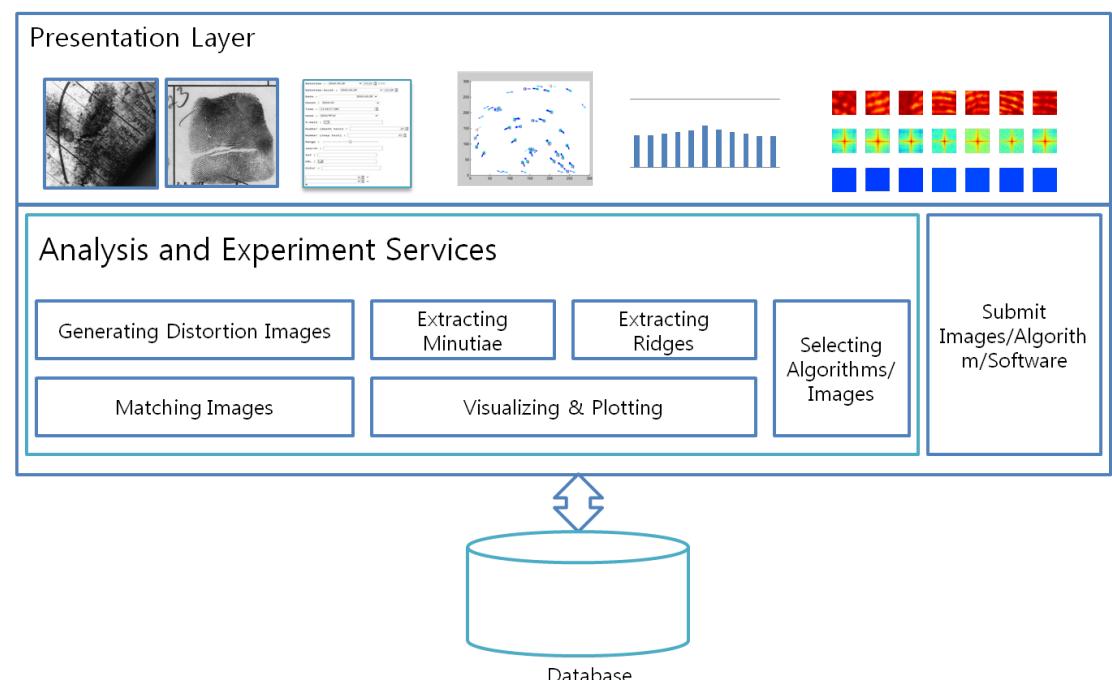


Figure 12.5: Analysis and Experiment Services in DL framework.

Table 12.1: Basic Terms and Definitions of 5S formalization [250]

Term	Definition	Term	Definition
$DO_i, DO_j$	digital objects $i, j \in C$	$V$	Vertex
$C$	a collection $\in Coll$	$Stm_i$	$\Psi_{ij}.Dom$
$Coll$	a set of collections	$\Psi_{ij}.Dom$	$V \times Streams$
$stm_j$	a stream	$S^3$	Streams $\cup$ Structures $\cup$ Spaces
$st_j$	a structure	$tfr$	$S^3 \times Spaces$
$\Psi$	$V \times Streams \Rightarrow (N \times N)$	$sp_j$	a space $j$
$St^2$	a set of functions $\Psi$		

**Image and Dataset Extraction** (Below: Adapted from TPDL: Fingerprint article)

A minutiae extraction algorithm is used to identify the locations and quality of major features, e.g., ridge bifurcation and termination. A third algorithm attempts to automatically trace the ridges in images resulting from smears, partial-smudges, or high humidity. High humidity refers to an overly oily or wet print that causes ridges to run together.

These two feature extraction algorithms form a service, *extracting*, that can be informally defined as *given a digital object, produce a descriptor from the object that represents the digital object*. User input is required as  $stm_i$  and outputs are  $(st_j, \Psi_{ij})$ . Pre-condition and post-condition are  $stm_i \in Streams$  and  $st_j \in Structs; \Psi_{ij} \in St^2; stm_i \in \Psi_{ij}.Dom; st_j.V \in \Psi_{ij}.Dom$ , respectively.

**Algorithmic Evaluation** (Below: Adapted from TPDL: Fingerprint article)

Evaluation is a critical service among these experimental services. Evaluation criteria can be 1) algorithm performance, 2) algorithm efficiency, 3) minutia reliability, and 4) image quality. First, performance metrics include indicators used in the FVC such as 1) number of rejected fingerprints during enrollment; 2) number of rejected fingerprints during genuine matches; 3) number of rejected fingerprints during impostor matches; 4) impostor and genuine score distributions; 5) FMR(t) / FNMR(t) curves, where FMR is the false match rate, FNMR is the false non-match rate, and t is the acceptance threshold; 6) ROC(t) curve, where ROC is a receiver operating characteristic; 7) equal-error-rate (EER), the value that EER would take if the matching failures were excluded from the computation of FMR and FNMR (EER\*); 8) the lowest FNMR for  $FMR \leq 1\%$ ; 9) the lowest FNMR for  $FMR \leq 0.1\%$ ; 10) the lowest FNMR for  $FMR = 0\%$ ; and 11) the lowest FMR for  $FNMR = 0\%$ . Second, metrics for measuring efficiency include 1) average enrollment time, 2) average matching time, 3) average and maximum template size, and 4) maximum amount of memory allocated. Third, minutia reliability and image quality can be measured with the combination of many different maps (e.g., low frequency, high frequency, and directional).

### 342 12. BIOINFORMATICS, SCIENTIFIC, AND SIMULATION DIGITAL LIBRARIES

Given a digital object, an evaluating service produces an evaluation (i.e., a real number) for it. Input is  $do_i$  and output is  $(do_i, w_i)$ . Pre-condition is  $\exists C \in Coll : do_i \in C$  and post-condition is  $w_i \in [a, b] \subset R$ .

**Visualizing & Plotting** (Below: Adapted from TPDL: Fingerprint article)

Analysis results can be visualized by projection to measurable spaces. Visualization techniques can be used to analyze the appearance and disappearance of minutiae over distortion degrees.

Visualizing and plotting can be described as processes that, given a collection, produce visualizations such as charts, histograms, plots, or meshes. Input for a visualizing service is a collection  $C$  and a transformation  $k$ , and output is a space  $j$ . Pre-conditions and post-conditions are  $C \in Coll$  and  $tfr_k(C) = sp_j \in Metric$ .

## 12.4 CASE STUDIES

The formally defined content, users, workflows, and services have lead to the design and implementation of a bioinformatics, scientific, and simulation supporting digital library (SimDL). SimDL has been integrated within large scale cyberinfrastructure and in local institutions to manage applications including OpenMalaria [604], EpiSimdemics [41], EpiFast [57], and GaLib. SimDL is a digital library that meets the minimal requirements for the computational epidemiology simulation community [376]. A prototype of SimDL has been integrated with a simulation infrastructure used by the Virginia Bioinformatics Institute at Virginia Tech.

### 12.4.1 PROTOTYPE: COMPUTATIONAL EPIDEMIOLOGY

(Below: Adapted from JCDL: Short paper)

Collaborations between public health researchers in the USA and Europe, each with decades of experience in simulations, led to SimDL's initial development. SimDL was designed to support the experimentation workflow for multiple models, including [604] presented in Figure 12.2, within a reusable model-independent design. Figure 12.2 displays the interface for gathering input parameters to a malaria model. Collaborating researchers had developed simulation systems which were complex for non-model developers to execute. Simulation software that did not use XML schemas to define inputs to models required wrapping scripts between XML and model-specific input files. The ontologies were found to aid in a model's UI presentation through the generic browser. A model-independent UI was built to rapidly expose new disease models and present input, result, study design, analysis, and document collections. Brokers also were developed to support the UI in submitting input configurations, compliant with XML Schemas, to available computational systems.

### 12.4.2 PROTOTYPE: FINGERPRINT ALGORITHMS

(Below: Adapted from TPDL: Fingerprint article)

We have implemented a basic prototype of this framework to conduct experiments with the feature extraction algorithms previously mentioned. The prototype consists of DL services to manage a distorted image collection, select and execute an algorithm, and execute analyses. The analysis processes allows a researcher to hold several parameters constant by careful selection of distortion parameters, e.g., x-axis translation, rotations, and skin plasticity. We are developing a plug-in system for easier integration of new algorithms.

We have developed a collection of real-world images. For several selected images, we have generated a range of distorted images and produced a service for generating new distorted images as required. An experiment was successfully designed, executed, and analyzed to determine the effects of humidity, x-translations, y-translations, rotations, and skin plasticity on minutiae extraction.

The prototype includes an online collection of original and distorted images and a system for selecting and composing service workflows. The Google chart API is used to present results of completed analysis tasks. A web-interface is used to browse the image collection, image information, distortion parameters used to generate specific images, extracted minutiae, and ridge information.

Currently, our prototype system contains 137,785 prints (FVC2000: 3520, FVC2002:3520, SD27: 516, self-collected: 629, and distorted: 129,600). For the preliminary experiments, we generated distorted images from real fingerprints. As a result of our experiments, the system yielded the following: 1) matching scores of a minutia extraction module MINDTCT and BOZORTH3 produced by National Biometric Image Software of BIST matching algorithm with distorted image sets (see Figure 12.6); 2) minutia counts of MINDTCT algorithm with distorted image sets (Figure 12.7); 3) minutia reliability of MINDTCT algorithm with distorted image sets (see Figures 12.8 and 12.9); and 4) improvement of schema presented in previous work [353].

Our framework is scalable but limited by file system storage space. Current terabyte storage devices can have roughly 1-10 billion images assuming 800 base images and 100,000 distortions at 100KB per file. Distorted image generation time is 1.0 sec. on a Pentium 4. Both time and space complexity are  $O(n)$ .

### 12.4.3 PROTOTYPE: LARGE-SCALE NETWORK SIMULATIONS

CINET: a Cyberinfrastructure for Network Science

## 12.5 SUMMARY

SimDL provides the minimal set of digital library services required to support a representative selection of bioinformatics, scientific, and simulation-based domains. These services

### 344 12. BIOINFORMATICS, SCIENTIFIC, AND SIMULATION DIGITAL LIBRARIES

include metadata definition, schema generation, metadata extraction, collection management, collection workflowing, and component communication. Further applications exists into other simulation domains, medical informatics, and scientific fields.

(Below: Adapted from ECDL: 2010 article)

SimDL currently supports input configurations as desired by one institution. Functional extensions and research efforts to improve SimDL are in continual, iterative devel-

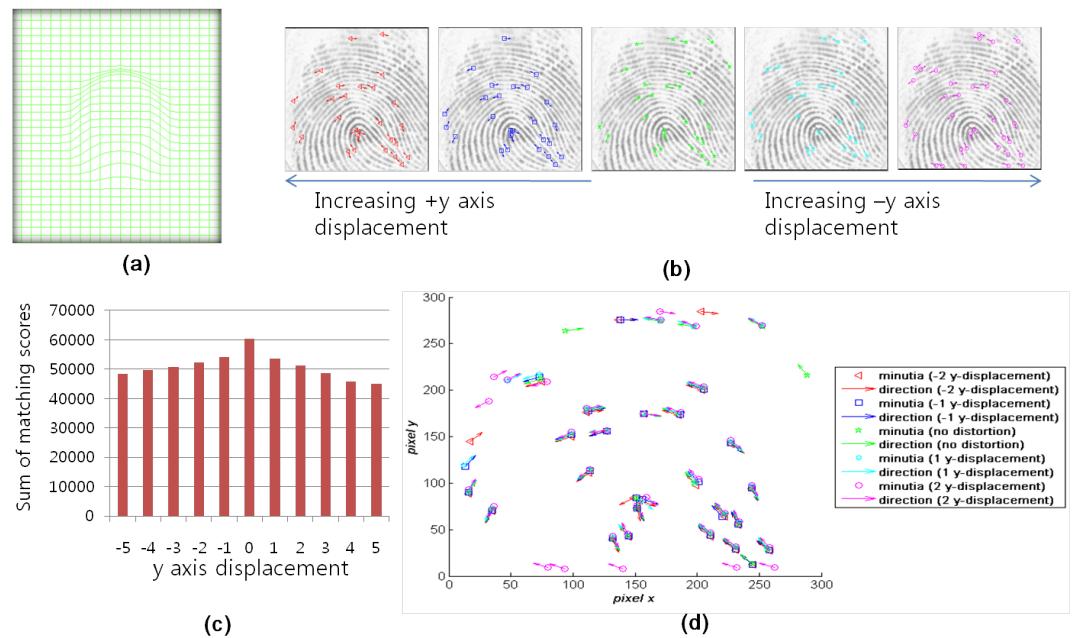


Figure 12.6: Example analyzing effects of y-axis displacements on matching quality: (a) skin distortion model selected; (b) distorted images; (c) histogram of y displacement versus sum of matching score; (d) plotting of minutiae spatial distribution.

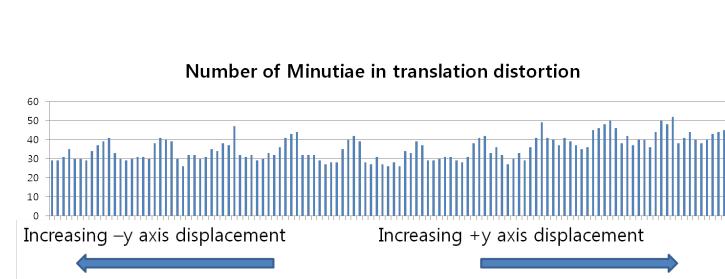


Figure 12.7: Number of minutiae in translation distortion.

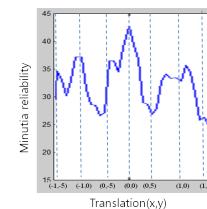


Figure 12.8: Average minutiae reliability: images distorted by translation.

opment. While the intention is to provide SimDL as an extensible platform for deploying digital libraries to manage scientific content, several components still in development are required to promote adoption within the epidemiology, simulation, and larger scientific communities. Plug-in components are in development to annotate and recommend digital objects; ask high level semantic search queries; query across SimDL instances; federate SimDL instances with other digital libraries through semantic schema mapping and domain ontologies; provide interoperability for accessing metadata and collections; and provide a method for communication within the DL system (e.g., message boards). Visualization components may be added to the generated UI to provide intuitive input configuration navigation; dragging and dropping of sub-configurations; fisheye overviews of a schema; and treemap views of datasets, analyses, documents, and annotations collections. Maintaining user sessions across SimDL instances also may provide more coherent use of user credentials, demographics, access rights, preferences, tailored views, expertise, classes of rights (access to content), and roles (access to system functionalities). The foremost expec-

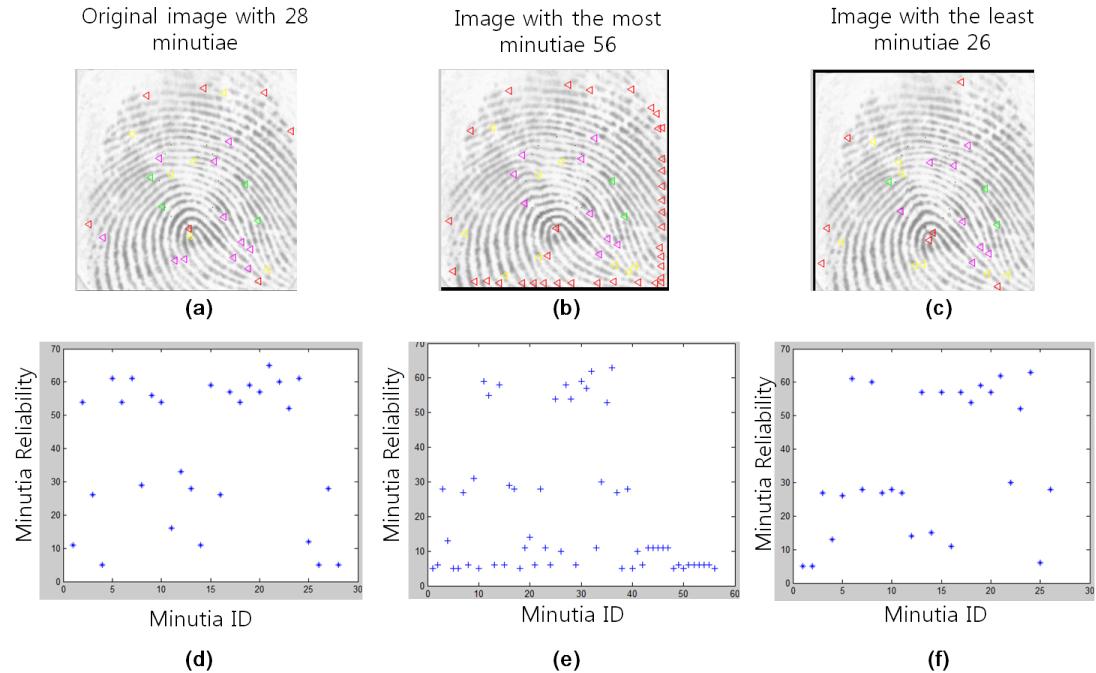


Figure 12.9: Spatial distributions of minutia reliability: (a) original image with 28 minutiae; images which increased (b) and decreased (c) the most in minutia points after distortion (56 and 26 minutiae respectively); and (d)-(f) minutia reliability of each image.

### **346 12. BIOINFORMATICS, SCIENTIFIC, AND SIMULATION DIGITAL LIBRARIES**

tations for a SimDL instance are to holistically provide DL services for interested simulation groups within a domain using multiple models as contexts.

(Below: Adapted from JCDL: Short article)

In this paper, we have presented a digital library design for supporting simulation communities, models, and infrastructure. Our preliminary work on developing a simulation DL system makes use of domain and model ontologies to manage and present epidemiological content. Having delivered a prototype of the system, we have a DL that supports a typical experimentation workflow and is capable of conducting and managing studies. The prototype DL successfully supports workflow specific user interactions and manages content at each stage of the workflow. The prototype indicates our model-independent approach to simulation DL design is deployable for model-specific environments, as new models may be rapidly exposed through ontology-based storage and interfaces. Our ongoing efforts include refinements to SimDL search and new deployments for managing multiple simulation systems in network science.

(Below: Adapted from TPDL: Fingerprint article)

Our main contribution is supporting collaborative research for researchers and trainers with services for generating distorted image datasets, testing different algorithms (e.g., for minutia detection and matching), and managing and workflowing scientific research datasets, algorithms, and analysis results. We have defined a framework for managing a large image collection, executing image-based algorithms, and analyzing an algorithm-specific experiment. We are integrating each implemented service under the proposed framework. We plan to verify this prototype in terms of algorithm correctness before and after integration. In addition, we will confirm that findings of experiments relate to the practice of researchers and fingerprint analysts. We also plan to incorporate (training and matching) algorithms from three other types of fingerprint DLs [353] with our collection of distorted images. We plan to create a publicly available online version of our fingerprint experimentation and analysis services and prototype, with the capability of allowing users to execute their own models on our distorted image collection. Astronomy and geo-location identification domains provide a parallel corpus of algorithms that compare images based on feature extraction. Comparisons of these algorithms would be useful for cross-domain generalization.

## **12.6 EXERCISES AND PROJECTS**

Select one of the following projects:

1. Bioinformatics, scientific, and simulation applications: Summarize the digital library service requirements you might expect for two of the following projects: earthquake simulation repositories [324], embedded sensor network DLs [75], community earth systems [165], mathematical-based retrieval [695], chemistry systems [387], fingerprints [412], and epidemiology [41].

## 12.6. EXERCISES AND PROJECTS 347

2. Workflow applications: Install one of the following and determine the types of digital libraries it may be suited to support: Kepler [399], Taverna [488], Triana [409], and Pegasus [150]
3. Formalisms: Create your own formal definitions for the following content: medical records, protein folding predictive simulations, and transportation modeling simulations.

# Geospatial Information

by Lin Tzy Li and Ricardo da Silva Torres

*Abstract:* Geographic information is part of daily life. There is a huge amount of information on the Web about or related to geographic entities – documents, photos, and videos that are related to somewhere on Earth – and people are interested in locating them on maps. This chapter surveys the geographic information world, the Geographic Information Retrieval (GIR) area, and Multimodal Information Retrieval that is focused on geographic information.

## 13.1 INTRODUCTION

Geographic information is characterized by the existence of an attribute which is related to a localization on Earth, for example a geographic coordinate, or a relationship to some other object whose geographic location is known. It might be a fully complete address (street name, number, and postal code) or even a single reference such as the airport name LaGuardia Airport which also indicates the name of the city where it is located (New York City).

There is daily use of geographic information. Thus it is not surprising to find a great amount of information on the Web about geographical entities and great interest in locating them on maps. There are many devices with a GPS unit embedded, such as cellphones and cameras, that add location tags to photos and other user published content like Twitter updates, Facebook posts, and other posts in social medias. Accordingly, location information is commonly stored as metadata. On the Web, tools like Google Maps<sup>1</sup> and Google Earth<sup>2</sup> are very popular, and partially meet the needs of Web users for geospatial information. By using these tools, users can, for example, find an address on a map, look for directions from one place to another, find nearby point of interest (e.g., restaurants, coffee shops, museums), and list the nearby streets.

An example of a query that most existing *Information Retrieval* systems do not support is: “Which are the Web pages of the cities which are neighbours of Blacksburg?” The reason is that spatial operators must be supported by spatial databases, and those are not integrated with Web search systems. This kind of problem is tackled in the Geographic In-

<sup>1</sup><http://maps.google.com/>

<sup>2</sup><http://www.google.com/earth/>

## 13.2. GEOGRAPHIC INFORMATION 349

formation Retrieval (GIR) area, which improves upon information retrieval (IR) by adding handling of geographic information found in Web documents and queries.

In this chapter we survey the Geographic Information Retrieval (GIR) area. Some of the concepts are related to geospatial (or geographic) information. Others are related to multimodal retrieval, as it integrates with geographic information. Additional insights come from computer vision and content-based image retrieval. A key challenge is recognizing places based on image or video content [401, 285, 333, 370]. Thus, there is considerable extension beyond text analysis based only on metadata [646].

### 13.2 GEOGRAPHIC INFORMATION

Fundamental concepts in this field are related to the world of geographic information, which is at the heart of a Geographic Information System (GIS).

A geographic entity/object (e.g., city, country, lake, etc.) can be localized on Earth because of the use of a coordinate system. Given an (x, y) coordinate point, x representing a horizontal position and y a vertical position, we can distinguish from other points in the coordinate system space.

The most popular and ancient coordinate system to locate points on Earth is the geographic coordinate system; every point is at the intersection of a meridian (longitude) and a parallel (latitude). The coordinates are measured in degrees in relation to the center of the sphere that represents the Earth (Figure 13.3).

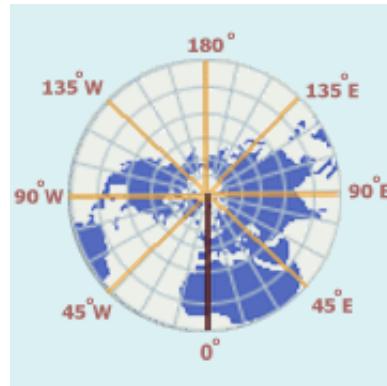


Figure 13.1: Top view of North Pole: longitude lines (radii) and latitude lines (concentric circles). Bold lines identify some longitude lines. *Source: nationalatlas.gov.*

A meridian is an imaginary arc on the Earth's spherical surface that is drawn from the North Pole to the South Pole. The meridians are vertical lines of longitude. Longitude 0 degrees is called the Prime Meridian, which is usually the Greenwich Meridian, that passes through the Greenwich Observatory in England. To the East of the Prime Meridian

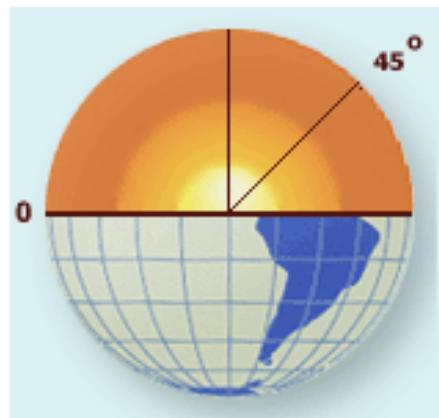


Figure 13.2: Cutaway view of Earth showing latitude  $45^{\circ}N$ , which is the angle measured from the center of the sphere. *Source: nationalatlas.gov.*

there are 180 degrees of longitude and to the West another 180 degrees. The east and west directions can be replaced by positive and negative signs, respectively. For example,  $105^{\circ}W$  is equal to  $-105^{\circ}$ . Figure 13.1 shows a schematic view from the North Pole to illustrate longitude lines and how they are drawn.

On the other hand, the Equator is an imaginary line around the Earth that divides it into two equal halves (north and south). It marks the 0 degree latitude line. All other latitude lines are parallel and equidistant from each other; thus the latitude lines are known as parallels. There are 90 degrees of latitude to the north and to the south. Parallels above (north of) the Equator are represented as positive degrees and conversely those below (south) appear as negative degrees. For example,  $45^{\circ}N$  is equal to  $+45^{\circ}$ . Figure 13.2 shows how latitude is measured.

### 13.2.1 RASTER & VECTOR DATA

There are essentially two kinds of geographic data format used in GIS (Geographic Information System):

**Raster data** comes from satellite images or digital aerial photos, for example, and it is stored as a matrix of cells (or pixels) arranged in rows and columns. Each cell stores some data value which is the target information. Raster data will have an origin point that will serve as reference for other cells' relative position. Based on its raster coordinate system, a GIS is able to calculate the real-world location for every cell in a raster. This kind of data is useful for continuous data where contours or well-defined shapes are not necessary.

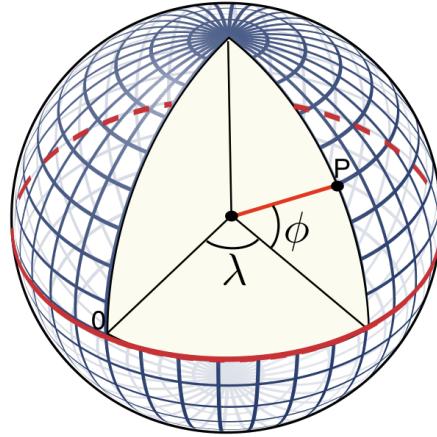


Figure 13.3: Cutaway view of Earth sphere: P is located at latitude  $\phi^{\circ}N$  and longitude  $\lambda^{\circ}E$ .

**Vector data** represents geographic objects like rivers, city boundaries, and houses as basic geometric forms of lines, polygons, and points. As we have seen previously, geographic objects have coordinates (such as latitude and longitude) that associate them with a location on Earth. A point is defined by a coordinate, a line by two coordinates, and a polygon by three or more.

Examples of these data formats, overlaid together, are shown in Figure 13.4.

Some current database management systems (DBMSs) support storing geographic vector data and provide special operators and functions to query them, e.g., MySQL and PostgreSQL (with PostGIS extension).

### 13.2.2 SPATIAL RELATIONSHIPS AND QUERIES

Spatial relationships refer to relative positions between objects in a space (geographically speaking, though clearly this also fits with the 'Space' part of 5S) and they can be classified as [68]:

**Topological:** this kind of relationship indicates connections between objects such as adjacent to, containing, or is contained, but it does not include measurement or direction. Egenhofer [173] classifies the topological relationships between two dimensional objects as: disjoint, meet, overlap, covers, contains, equal, covered by, and inside. Clementini et al. [123] summarize them as: disjoint, inside, touch, cross, and overlap (Figure 13.5).

**Metric:** this relationship expresses quantitative measurable attributes like area, distance, length, and perimeter.

352 13. GEOSPATIAL INFORMATION

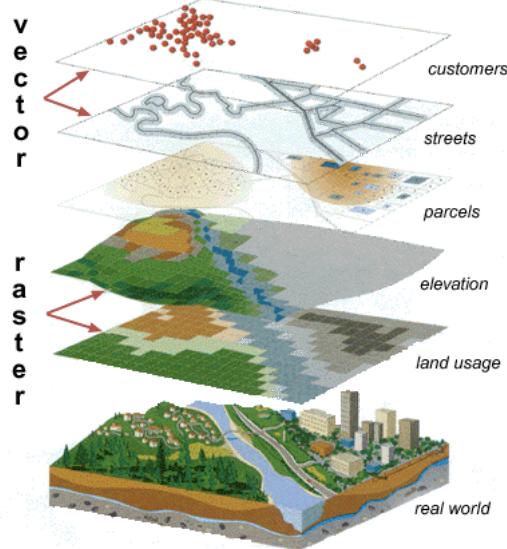


Figure 13.4: Vector and Raster data can be overlaid. Source ESRI.

**Directional:** this relationship is used to express orientation such as cardinal points (e.g., North, South, East, and West), as well as order or position like ahead, above, and under.

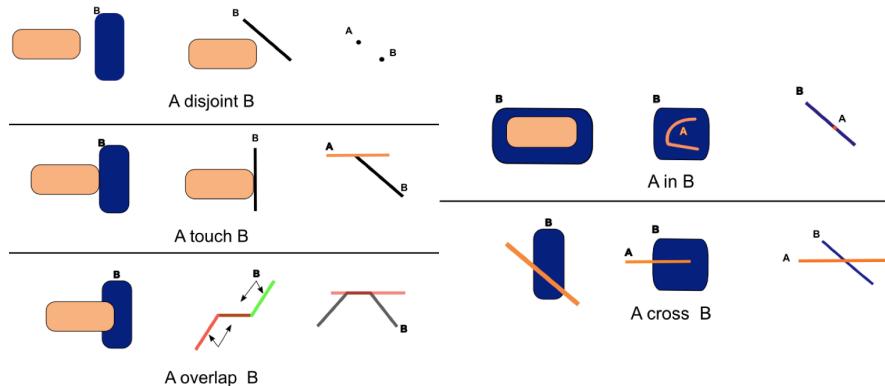


Figure 13.5: Examples of topographic relationship (from [92])

These concepts lead us to spatial queries, also known as geographic queries, which express spatial relationships between two objects in a very well defined space, either with

### 13.3. GEOGRAPHIC INFORMATION RETRIEVAL 353

or without geographic coordinates such latitude and longitude. According to [371], spatial queries can be:

**About a given point** inside a coordinate system, like “What can be found at the point given by the following latitude and longitude: 37.228, -80.423?”;

**About a region**, where you are interested in something inside it, e.g., “In which state or region is the Grand Canyon located?”;

**About distance and a buffer zone:** This is illustrated by queries like “Which are the cities 50 miles from Blacksburg’s boundaries?”;

**Path** involves searching along a structured network comprised of connected lines, such as electric lines and networks, water or gas pipes, or transportation lines. Examples include the shortest path between two points in a network and even a more complicate query like “What is the fastest path from Blacksburg to Washington, D.C.?” , which involves distinct variables as distance, direction, and even time.

**Multimedia**, when a query requires a variety of information types (e.g., text, image, and geographic), e.g., “In which rivers we can find fishes similar to a given picture, and that are from the darter family?”

## 13.3 GEOGRAPHIC INFORMATION RETRIEVAL

Geographical Information Retrieval (GIR) is an area concerned with challenges such as recognizing, querying, retrieving, and indexing geographical information. It combines research in database, human-computer interaction (HCI), geographic information systems (GIS), indexing, information retrieval (IR), and georeferenced information browsing [371], as well as visualization of information on maps. According to [320], GIR aims to improve information retrieval centered on geographic information in non-structured documents such as those found in the Web.

Two main concepts of this area are geoparsing and geocoding. Geoparsing is a process of recognizing references with locations inside documents, while ignoring false references (e.g., a place name that is also the name of an organization or person), while geocoding is a process to associate a document with some specific latitude and longitude based on locations recognized by geoparsing. Thus, geocoding consists of mapping a document to a location on Earth. For example, based on where its content refers to, we can assign a latitude and longitude to a document, so later user can retrieve this document based on geographical queries (e.g., “Give me all documents that refer to parks in the Blacksburg vicinity.”).

In the following subsection, why geographic information on the Web matters is discussed, and a GIR architecture is presented. This will serve as a baseline to discuss GIR ’s main concepts – geoparsing and geocoding – as well as research challenges.

## 354 13. GEOSPATIAL INFORMATION

### 13.3.1 GEOGRAPHIC INFORMATION ON THE WEB

As was introduced earlier, traditional search services are based on keyword matching and do not consider that keywords might represent geographical entities which are spatially related to each other. Yet, even though these relationships have not been explicitly used in a query, they are potentially relevant to users [319].

The screenshot shows a Google search results page. The search query "cities which are neighbors of Campinas?" is entered in the search bar. Below the search bar, it says "Aproximadamente 7.860.000 resultados (0,20 segundos)". The results list includes:

- Campinas - Wikipedia, the free encyclopedia**  
en.wikipedia.org/wiki/Campinas - Traduzir esta página [+1](#)  
Ir para **City twinning**: Campinas is a city and municipality located in the coastal interior of the state of São Paulo, Brazil. Campinas is the administrative ... [»](#)
- [PDF] ENTRE O BAIRRO E O LUGAR: EXPERIÊNCIA URBANA NOS ...**  
www.geografia.ufpr.br/.../Microsoft%20Word%20-... [+1](#)  
Formato do arquivo: PDF/Adobe Acrobat - Visualização rápida  
de EM JR - Artigos relacionados  
approach) in the DICs **neighborhood** at Campinas, São Paulo. From the place approach, we sought to apprehend the **neighborhood** meaning into the **city** ...
- Campinas travel guide - Wikitravel**  
wikitravel.org/en/Campinas - Traduzir esta página [+1](#)  
The "Maria Fumaca" [5] starts 25km away in the **city** of Jaguariúna and arrives in **Campinas** at Anhumas Station, in the **neighborhood** of Fazenda Anhumas. ...
- Real estate for rent in Loteamento Alphaville Campinas, Campinas ...**  
www.vivareal.net/.../campinas/neighborhoods/loteamento-alphaville-... [+1](#)  
Buy · Rent · Brazil Rentals · **Campinas** Rentals; Loteamento Alphaville **Campinas** Rentals ... **City Campinas** [Remove]. Neighborhood(s) Select Neighborhood ...
- Apartment for sale in Campinas, Brazil - Rua Ubatuba - VivaReal.net**  
www.vivareal.net/apartment-29112829/ [+1](#)  
16 set. 2011 – HOA fee: US\$ 66.00. External ID: 29112829. **City: Campinas** ...  
[+ Exibir mais resultados de vivareal.net](#)
- Alphaville Campinas - Neighborhood, Apartment/Condo | Facebook**  
www.facebook.com/pages/Alphaville-Campinas/138298589573166 [+1](#)  
Welcome to Alphaville **Campinas** on Facebook. Join now to write ... **Neighborhood** · Apartment/Condo · **Campinas**, São Paulo. Neighborhoods in Nearby **Cities** ...
- Getting to Campinas - São Paulo - iGuide**  
iguide.travel/Campinas/Getting\_There [+1](#)  
+100 items – Getting to **Campinas**, **Campinas** travel guide and map.  
Sítio das Pedras de Ouru 34 km (21 miles)  
Fazenda Santo Antônio 35 km (21 miles)

Figure 13.6: Google Search result for neighbors of Campinas.

For example, typing “pages of cities which are neighbors of Campinas” into Google search, will return web pages with the typed in terms (Figure 13.6). However, that query

### 13.3. GEOGRAPHIC INFORMATION RETRIEVAL 355

encompasses a geographic query (neighbors) meaning that all (ten) cities that share boundaries with Campinas (Figure 13.7), although not mentioned, should be returned in the result set too.

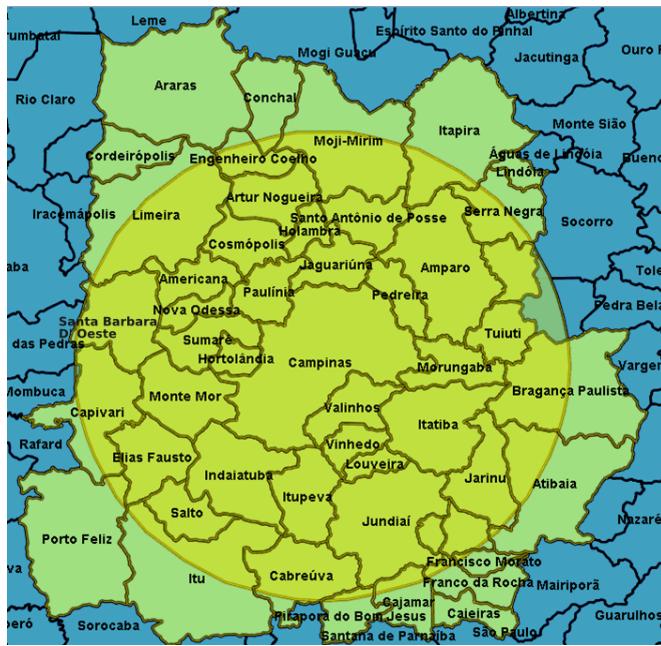


Figure 13.7: Campinas neighbourhood and cities within 50 km.

The difficulty in processing this kind of query comes from the need to combine traditional queries executed on Web search mechanisms with spatial operators usually implemented in spatial databases.

In [386], users were invited to use Web tools for performing tasks related to the search of geographic entities. All proposed tasks included at least one kind of spatial relationship. Obtained results indicate that: a) there is a tendency of users to break down the geographic searches into two or more steps when using the current multi-purpose search tools; b) the geographic relationship aspect of a spatial query on the Web is solved by users either inspecting visually the location on a Web map or by taking advantage of Web tools such as Wikipedia; c) geographic queries involving well-known or popular objects such as hotel locations in a city are solved easily by a single text-based search.

Let us take as an example one query used in that experiment: “Search for Web pages of cities in the neighborhood of Curitiba.”. To perform that task, users switched between keyword-based search, Web mapping, and encyclopedia tools.

### 356 13. GEOSPATIAL INFORMATION

It is not enough to send the city's name (e.g., Curitiba) and a term referring to a geographical relationship to a search tool, because it will just match the keywords with pages' textual contents. For those users who are used to geographic queries, it is common to rewrite the query into a form which the search tools can use to retrieve relevant results.

To sum up, as a rule, a more complex geographical query was broken down into two main steps. Keeping in mind the example we mentioned earlier, in the first step, the user processes the geographical part of the query by using a keyword search Web tool or Web mapping tool. Still This step consists of, for example:

- using a user's previous knowledge to associate a city to a region;
- visiting previously user-known Web pages (e.g., Wikipedia). From these Web pages, users could find the cities nearby, or the distance between cities. In this case, users go first to Wikipedia to find the city of interest and then create a list of candidate cities;
- submitting other words to the search tool in order to return the list of cities. This is the case in which the user first searches using the phrase "Curitiba metropolitan cities" to get the list of cities in the Curitiba metropolitan area, thus resolving the part of the query that refers to Curitiba's neighborhood;
- using a map service to localize a city used as reference, visually inspecting a map, and manually creating a list of cities that satisfy the target geographical relationship. An example involves going to a mapping tool like Google Earth or Google Maps first, and finding the city of interest by visually picking the neighboring cities.

Finally, the second step consists in searching for each city listed in the first step by:

- submitting the city name as keyword to the search tool to find the Web page of that specific city;
- reaching the city page by using a URL naming pattern previously known (e.g., the URL of a city's home page in Brazil is formed by *www.<city name>.<acronym>.gov.br*, where <city name> is the city name and <acronym> is the acronym of its state, so for Curitiba, a city in the state of Paraná (PR), its URL is *http://www.curitiba.pr.gov.br*);

There were some cases where a user just relied on the map tools, like Google Earth, that show the cities and their facilities locations (e.g., hotels and subway stations) on a map to answer the question posed by the task (e.g., "Barcelona's hotels which are near subway stations"). In fact, using these tools, the geographic relationship is resolved by the user,

who infers and inspects it visually on the map. Therefore there is no automatic list; in this case the user is in charge of building the list manually.

### 13.3. GEOGRAPHIC INFORMATION RETRIEVAL 357

Incorporating geographic relationships in web searches is not supported yet; as seen in this study, they are mostly processed by the user first. This could be explained by their inherent complexity, which is worsened by their imprecision and subjectivity. Thus, some geo-related concepts like near or south might depend on the user's search context [321, 702].

For some specific objects (e.g., hotels and city names) and relationships, when geographic terms (e.g., near) can be found on some Web pages, the use of current search tools is quite straightforward, as is illustrated by queries like: 'Web pages of Barcelona's hotels which are near to subway stations'. Such success is explained by those objects' search popularity [569, 290, 321]; Web pages that contain those keywords thus can be retrieved by popular keyword-based search tools.

The ideal Web search tool for geographic queries should be able to process the geographical relationships and retrieve all the relevant results on the Web that match the users' intention expressed by their query. This kind of query is common in a GIS (geographical information system) which works with structured data. Hence, there is need for investigation of strategies to integrate these technologies, so that Web queries that include this kind of feature can be easily processed, without undue user frustration or effort. One perspective is that geocoding Web pages can help, but the challenge is how to do that in light of the volume of data on the Web and the high level of ambiguity common in such queries.

#### 13.3.2 GIR ARCHITECTURE

As is shown in Figure 13.8, a GIR system can be divided into three layers: presentation, processing, and data. The main modules are:

**Query input, Result Presentation, and Feedback Presentation.** These modules are in charge of dealing with HCI: query input by the user, presentation of results returned by the system, and user feedback about the results. They forward data to lower level modules aiming at result improvement.

**Geoparsing** is a module responsible for recognizing references to geographic entities in a digital object and for disambiguating them based on their content, geo-ontologies [576], and semantic databases.

**Geocoding** is a module that will take care of associating appropriate geographic coordinates with a digital object, which can be one or more geographic points or even a geographic region.

**Query Processing** is responsible for interpreting and processing the input query. Besides, it handles subsequent interactions aiming to refine results.

### 358 13. GEOSPATIAL INFORMATION

**Relevance Feedback** will make improvements in retrieval and/or ranking algorithms by taking into account user assessment of results previously returned. The objective is to return better results through the course of system-user iterations.

**Ranking** is a module whose algorithm is in charge of ordering the results according to an estimation of their relevance to the user query.

**Semantic Repositories** store place names, how they are organized and related to each other, and other connected information, so they can be used in geoparsing and geocoding. The geographic knowledge can be represented by ontologies; these are called geo-ontologies [576, 319] or geographic ontologies [68]. More information about places can be found in gazetteers and/or thesauri, and even from previously geocoded Web pages. A gazetteer is a dictionary of geographic names consisting of a name and its variants, that place's location, and its category (populated place, school, farm, hotel, lake, etc.). An example of a gazetteer is Geonames<sup>3</sup>. A thesaurus is a list of structured and defined terms formally organized and with concept relations clearly drawn [82], which is what distinguishes thesauri from gazetteers. For example, the *Getty Thesaurus of Geographic Names*<sup>4</sup> organizes places/location based on their spatial relation and administrative area, gives their geographic coordinates and all other names a place has, and supports places with similar names assisted by ontologies [648].

**Spatial Database:** in addition to what a regular database management system (DBMS) offers, spatial database also stores and provides spatial operations and queries over stored geographic objects. These objects can be stored as point, line, or polygon in a given coordinate system and spatial indexes are built to later speed up spatial/geographic queries (Section 13.2.2). Examples of geographic objects that can be stored are: shapes representing boundaries of state, city, or country; other polygon, point, or line representing a specific area on Earth.

#### Geoparsing & Geocoding

Items in a collection can be associated with one or more region on Earth; thus we find their *footprint* [224]. Jones [318, 319] defines **geocoding** as the act of associating a *footprint* with a geographic reference, while recognizing geographic references inside a document is called **geoparsing**, as we introduced previously.

In GIR, a collection of documents that refer directly or indirectly to a place need to have their footprint identified and thus be indexed spatially. That is, documents should be geoparsed and then geocoded.

Geoparsing process should be able to identify and disambiguate a place name appearing in a document and rule out false reference to it. It can be seen as a particular case of

<sup>3</sup><http://www.geonames.org/>. Last accessed: October 25, 2011.

<sup>4</sup><http://www.getty.edu/research/tools/vocabularies/tgn/>. Last accessed: October 25, 2011.

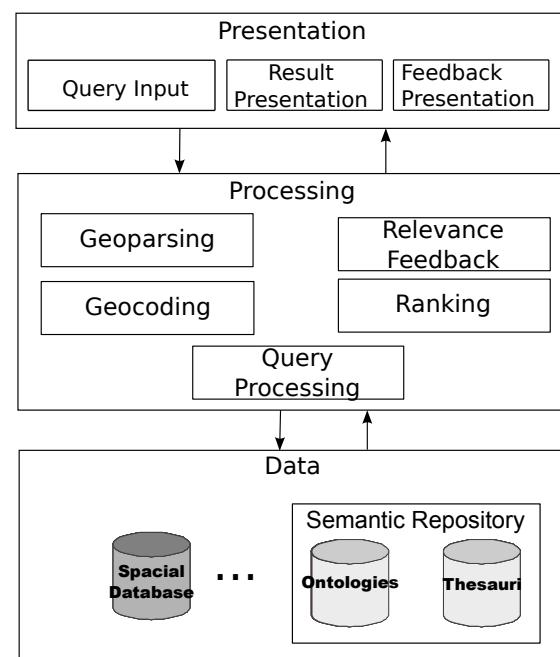


Figure 13.8: Architecture of a Geographic Information Retrieval

### 360 13. GEOSPATIAL INFORMATION

name entity recognition (NER), which identifies expressions in text and classifies them as person, event, organization, etc. [43].

However, there are challenges involved in recognizing place references and associating them with their coordinates [648, 372]. Examples include:

- Homonyms for places and person: For example, New York is a city name in Brazil as well as a state in the USA. Additionally, Luis Eduardo Magalhes is a Brazilian politician that has an airport, square, and city named after him;
- Descriptive place names change according to the historical context, culture, and customs that are in place, when a textual form is produced. For example, locating “North of the Russian capital” on a map would be difficult because the location of the capital of Russia has shifted repeatedly;
- Names of places change over time. For example, St. Petersburg, once Russia’s capital, was called Petrograd (1914-1924) and Leningrad (1924 - 1991);
- Geographic boundaries change over time. For example, Germany had different boundary over its history;
- Boundaries cannot always be clearly defined, for example in a conflict zone (e.g., Syrian-Turkish territory dispute).
- Names assigned to regions can refer to an area, rather than a well defined place, e.g., Southern California, or the Andes (in South America);
- Different names may refer to the same geographic entity, whether by error, language variations, or the legal existence of more than one valid way of writing it. For example, both Peking and Beijing refer to the capital of China, and Deutschland is commonly used to refer to Germany;
- Ambiguities arise due to different ways to describe a place, e.g., pseudonyms or expressions used in a specific context. For example, Saint Petersburg is called Piter by locals, while New York City is also referred to as the Big Apple. In Brazil the city of Sao Jose do Rio Preto, in Sao Paulo state, sometimes is called Rio Preto by locals, but, in another Brazilian state (Minas Gerais), Rio Preto is the official name of a different city;
- Names of famous buildings can lend their names to states; thus, New York is sometimes called the Empire State, after that tall building in NYC;
- Indirect references, such as to a road, like the Blue Ridge Parkway, may bring to mind both a region and event, e.g., due to a scenic drive in southwest Virginia and northwest North Carolina;

### 13.3. GEOGRAPHIC INFORMATION RETRIEVAL 361

- Imprecise references such as “100 km from Blacksburg” can refer to some point within 95 and 110 km. In another example, “South of Campinas” might include not just south, but also southeast and southwest locations.

Figure 13.9 illustrates accurate geoparsing. The names highlighted should be identified when geoparsing is applied to the text shown.

**Campinas** (Portuguese pronunciation: [kɐ̃pi'naʃ], *Plains*) is a city and municipality located in the coastal interior of the state of São Paulo, Brazil. Campinas is the administrative center of the meso-region of the same name, with 3,783,597 inhabitants as of the 2010 Census, consisting of 49 cities.

The municipal area of Campinas covers 795.667 square kilometres (307.209 sq mi). Campinas' population is 1,080,999 as of the 2010 IBGE Census;<sup>[1]</sup> while over 98.3% live in the urban region. The city's metropolitan area, as of 2000, contains nineteen cities and has a total population of 2.8 million people.

It is the third largest city in the state, after São Paulo and Guarulhos. The Viracopos International Airport connects Campinas with many Brazilian cities and also operates some international flights. The city is home to the State University of Campinas.

[Contents](#) [\[show\]](#)

#### Etymology

[\[edit\]](#)

Campinas means grass fields in Portuguese and refers to its characteristic landscape, which originally comprised large stretches of dense subtropical forests (*mato grosso* or thick woods in Portuguese), mainly along the many rivers, interspersed with gently rolling hills covered by low-lying vegetation.

Campinas was also known as "Cidade das Andorinhas" (City of Swallows), because it was a favorite spot for these migratory birds, which flocked annually in enormous numbers to downtown Campinas. However, they almost disappeared around the 1950s, probably because the church and plaza where they used to roost were torn down. Campinas' official crest and flag has a picture of the mythical bird, the phoenix, because it was practically reborn after a devastating epidemic of yellow fever in the 1800s, which killed more than 25% of the city's inhabitants.

An inhabitant of Campinas is called a campineiro.

Figure 13.9: Geoparsing example: Place names recognized in this extract of Wikipedia's page about Campinas (as of 11/03/2011).

On the other hand, Figure 13.10 illustrates geoparsing with errors, i.e., both true and false references [318]. In this case, false geographic references include personal names (Smedes Yok, Jack London), business names (Darchester Hotel, York Properties) and common words that are also places (bath, battle, derby, over, well). A possible strategy to distinguish between false and true references is to look for patterns and context [318]:

### 362 13. GEOSPATIAL INFORMATION

- For personal names, such as Jack London and Mr. York, the pattern is a first name or title followed by a location name;
- Business names like Paris Hotel have a location word preceded or followed by a business type;
- Detecting a spatial preposition helps validate a possible location; examples include: in, near, south of, outside, etc., as in “I lived in Blacksburg”;
- Street name can be distinguished from a city (e.g., Oxford Street) by verifying if its pattern is a location name followed by a road type.

JACK HAGEL, Staff Writer

Redevelopment of the World Trade Center site in New York is getting some input from a Raleigh real-estate maven.

York Properties President Smedes York was chairman of an Urban Land Institute panel at the World Trade Center and Lower Manhattan Summit last month.

The group heard presentations on how the area surrounding the site of the Sept. 11, 2001, terrorist attacks should be redeveloped. It suggested retail be a central focus for developers. The institute will issue a report based on the recommendations before the end of the year.

York was chairman of the Urban Land Institute, a Washington nonprofit organization, from 1989 to 1991. His dad, J.W. "Willie" York, joined the Urban Land Institute in 1947. That's where he met J.C. Nichols, the developer of Country Club Plaza in Kansas City, Mo. -- the center that inspired Willie York to build Raleigh's Cameron Village, the Southeast's first shopping center.

Figure 13.10: True and false references in geoparsing [318]

Some of the tools used in geoparsing and geocoding, to help detect and disambiguate place references, are geo-ontologies [319], gazetteers, and thesauri. Even Wikipedia has been used to enrich the knowledge base for geoparsing [146]. Thus, places can be referenced using an urban address, postal code, or the area code of a phone number [69].

According to the earlier discussion of modules, geocoding is a process to associate a document/digital object with some specific latitude and longitude, based on location references recognized by geoparsing. In fact, a document can be associated with one or more geographic objects, which in turn can be represented using a point, line, or polygon. Therefore, it is better to define geocoding as a process to associate a digital object with one or more footprints instead of just a point on Earth.

### 13.3. GEOGRAPHIC INFORMATION RETRIEVAL 363

As we observed in Figure 13.9, a set of place names can be recognized in a document. Therefore, one geocoding challenge is to determine which footprints should be associated with a given document. This often requires disambiguation of locations [43]. As was illustrated above, often the same name is used for different geographic locations (referent ambiguity), or the same location is described by different names (reference ambiguity).

The geographic knowledge required for this task is provided by a geo-ontology, supporting structuring, representation, and storage. It includes all suitable data types: place name, place type (city, state, country, etc.), footprint, relation (e.g., containment, adjacency) to other place names, population, historic names and dates, activities, etc. Given a set of geoparsed names, geocoding will find the corresponding matches in the geo-ontology. Then, based on related information, a decision can be made regarding a location to contribute, along with a document footprint: keeping, merging, creating, or discarding. Related information could specify how two extracted locations in a document are spatially associated with each other: are they close to each other? Another type of related information might address: what is their closest common ancestral node (e.g., state, county, or country)?

Consider an example borrowed from Batista et al. [43], where a document ( $D$ ) is geoparsed, yielding the result: Lisboa and Santa Catarina. Then, the first step of geocoding, checking the geo-ontology, yields these results:

- (i) Lisboa is a municipality;
- (ii) Lisboa is somewhere in the municipality of Mono;
- (iii) Santa Catarina is a civil parish in the Lisboa municipality;
- (iv) Santa Catarina is a street in the Porto municipality.

In this case, the closest spatial relation is (i) and (iii). Hence, the geocoding can result in associating the footprint of (iii) to that document. However, sometimes in a DL one might want to associate more than one footprint, in order to represent geographic concepts; this would ensure that the various possible scopes of a document are captured (geographic signature) [43].

#### Research Challenges

Research is required to address challenges in the presentation, processing, and data layers. Regarding the presentation layer, key concerns relate to how humans can express their information needs through queries, and how they can browse through the results returned by a GIR system. In the processing layer, important challenges are related to the identification and elimination of place name ambiguity, and the design of effective (as well as efficient) algorithms: for search, result classification, and ranking. Finally, in the data layer, considering the Web as a huge data repository, inconsistent and unstructured data make it difficult to identify and geocode the documents that are found.

### 364 13. GEOSPATIAL INFORMATION

**Presentation Layer** Early computer interfaces forced users to formulate structured queries, similar to what can be supported by a typical database query language (e.g., SQL). However, the majority of users lack knowledge and skill regarding proper use of such structured languages. As a result they do not completely express their needs, and the retrieved information does not fulfill their expectations. Considering also that users have to express spatial notions in words, more complexity and indirection are added to this problem. Often, system results do not fulfill user expectations. All this leads to the question: Does a query need to be expressed only by words/terms?

The difficulty in designing an interface where users can express themselves informally is related to problems that natural language processing researchers have been tackling for years: ambiguity, imprecision, and human language context dependency. Adding to these problems, the imprecision and temporal dependencies attached to the spatial data can make designing a good interface even more challenging.

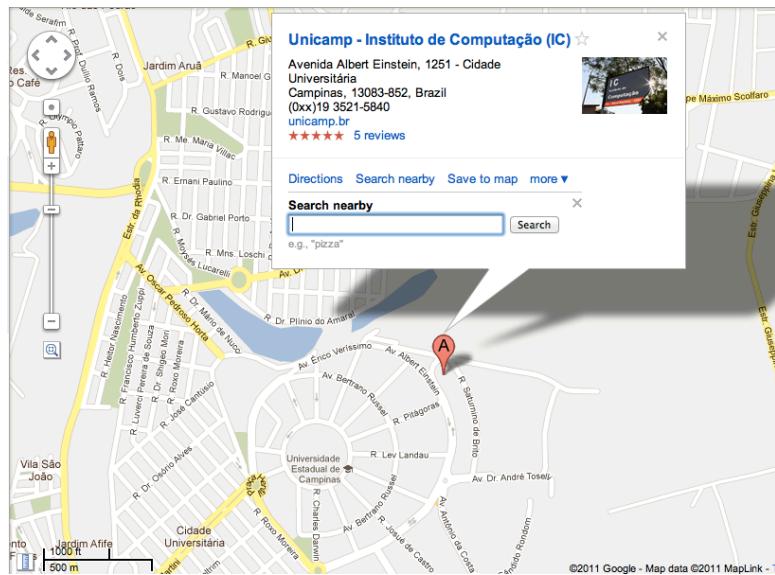


Figure 13.11: Example from Google Maps with a Point of Interest (POI) selected and search for something nearby enabled.

Consider, as an example of GIR results presentation, the execution of local searches such as those found in Google Places Search. In this case, a set of geocoded pages is retrieved by keyword-based search and they are pinpointed on Google Maps, yielding a geographic view and distribution of the results over a space (Figure 13.12). Moreover, when a point on map is selected, users are allowed to specify what they want to “Search nearby”. One

### 13.3. GEOGRAPHIC INFORMATION RETRIEVAL 365

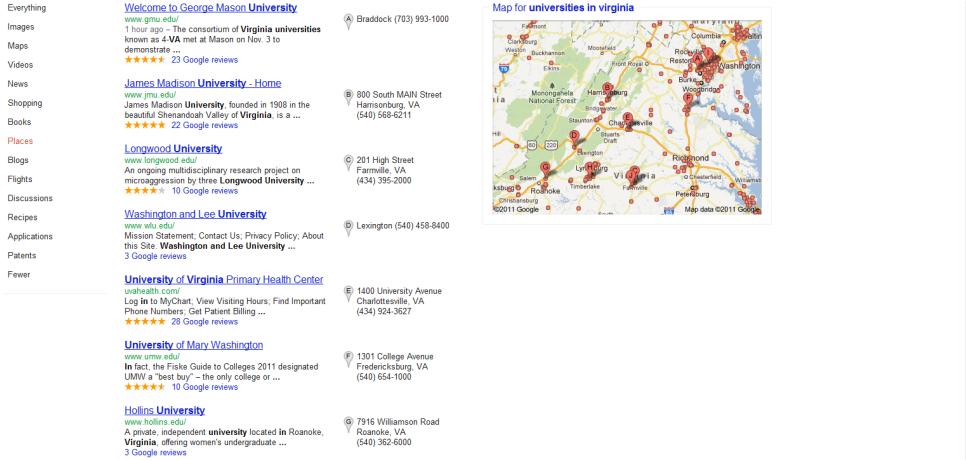


Figure 13.12: Example of results returned by Google Place Search.

strategy for a GIR query input interface could be using this kind of map-based interface, where it is easy to aggregate queries that involve spatial relationships (Figure 13.11).

There are still many challenges in the presentation layer relating to how geographical results are presented, how users can indicate which results are really relevant (so the system learns how to refine its searching), and how queries interface can be improved even using visual query interfaces [526, 340] such as those inspired by works developed for GIS: Spatial-Query-by-Sketch [59] and visual formulation of queries based on set of icons representing geographic features and relationship that are combined to build a geographic query [269, 183].

**Processing Layer** Challenges in the processing layer include identification and elimination of place name ambiguity (e.g., when a common name is used for a variety of places and objects). In this case, the system presents alternative choices (similar names) to the user. According to user feedback, a new query is sent to the system [286, 686].

Thus, supposing users are offered a interface where they can express their queries through semi-structured or natural language, it will be challenging to identify [642], extract [43], and manipulate references to places as well as intended geographical relationships between them [100, 569], and to deal with those derived imprecise and ambiguous references [226, 500, 86, 493, 382, 586].

Even if a geographic knowledge base is properly assembled and geocoded, in light of the huge amount of data spread over the Web, two challenges stand out: efficiently processing queries in a geographical Web search engine [114] and designing effective algorithms to predict if a document is relevant [690, 421, 181].

### 366 13. GEOSPATIAL INFORMATION

Moreover, more study is required of users' needs for geographical information, trying to understand their search behavior. For example, by analyzing search logs, Henrich and Luedcke [290] show that searches for geographic information in the USA are often about places to stay and visit, and users intend to buy or rent something from there besides learning how to reach it. In another investigation, the reason why users rewrite Web queries was studied. This aimed to identify users' preferences for physical distances between places searched, as well as the location from which a query was sent [321]. Finally, the most used geo-words and how they are reused by users was studied by Sanderson and Han [569].

**Data Layer** Considering the Internet itself as a big data repository, it is challenging to create and index automatically a geographic knowledge base from what is available on the Web [521, 319]. That involves dealing with inconsistent data and trying to identify and geocode data found in Web pages [69, 9, 62, 93, 16].

The data layer includes collections of target documents, as well as supporting structures such as ontologies, thesauri, and geospatial database, to assist with spatial operations and queries (as discussed earlier). Further research is needed regarding how to build and use such structures and repositories.

## 13.4 MULTIMODAL RETRIEVAL FOR GEOGRAPHIC INFORMATION

The discussion above of GIR focused on text geocoding and geoparsing. We also argued that only after documents are geoparsed and geocoded they can be placed on a map or queried by geographic location. However, in digital libraries digital objects go beyond text documents, e.g., to include images and videos. The quantity and space involved for these are growing rapidly. Besides, there are more devices connected with GPS and camera, such as smartphones, that embed location data in pictures and videos' metadata, along with other data such as date, time, and camera details. Therefore, it is useful to combine CBIR (Chapter 9), multimedia, and GIR techniques in DLs.

The process of associating a geographic location with photos and videos' metadata, which is called geocoding in GIR, in the multimedia field is often referred to as geotagging or georeferencing (also spelled geo-referencing) [401]. Further, in GIS, georeferencing is a term largely used to refer to defining a location where something exists, in a physical space, in term of a coordinate system (e.g., latitude and longitude).

Geotagging photos and videos is possible not only when you take or record them from a device with GPS, but it is also enabled by applications and services such as Flickr<sup>5</sup> and Panoramio<sup>6</sup>. In addition to supporting annotation, they allow users to organize and manually assign locations, using a map interface or geographically relevant keywords [401].

<sup>5</sup><http://www.flickr.com/>

<sup>6</sup><http://www.panoramio.com/>

## 13.4. MULTIMODAL RETRIEVAL FOR GEOGRAPHIC INFORMATION 367

Accordingly, the quantity of geotagged photos and videos is growing rapidly. For example, in Flickr, there were about 4.7 million geotagged items in 2010 [401], but this number increased to more than 165 million geotagged items on November 2, 2011.

### 13.4.1 IMAGE/VIDEO RETRIEVAL FOR GEOGRAPHIC INFORMATION

According to Luo et al. [401], “geographic information has been embraced by multimedia and vision researchers within a contextual modeling framework”, such as for event detection and classification, semantic scene content understanding, and annotation propagation. Their discussions are around the modalities of geographic information used and around geotagging driven applications in multimedia and vision research. They divided up the geotagging multimedia applications:

**Semantic multimedia understanding** encompasses social and cultural semantics, as well as annotation, organization, and retrieval of events, scenes, or objects. For example, white colors associated to a photo of somewhere in NE of USA during winter indicates snow;

**Geolocation and landmark recognition** aim to determine the location of an image, video, or series of images. In this case, collections of geotagged images are used as training and matching data to help predict the location of unknown images. Landmark images recognition can be seen as detection of somewhat unique objects in unknown images, that match images in a collection of geotagged images. Therefore its matched images’ geolocation will aid in the prediction of the location of a given unknown image;

**Media visualization** for: collections and landmarks, camera viewing directions, travel trajectories and routes, and photos in large collections (that can be browsed for tourism in 3D fashion);

**Recommendation for location-based services or products** such as planning vacations and identifying attractions based on users’ locations and interests. This category of applications can be divided further into: real-time recommendation, recommendation inference via geotagged images (considering spatial and temporal patterns), travelogues, and GPS trajectories;

**Social network applications:** Luo et al. [401] cite works that use tweets or Flickr uploads to discover time and location information related to an event. Users are seen as social sensors; their reports can document the spread of the consequences of an event (such as a flu epidemic or the movement of a typhoons) Therefore, it is important to predict the location of Flickr users. One strategy is based on their social connections’ public locations, since users tend to communicate more with closer friends;

**Mapping applications** can use geocoded photos to produce different kinds of maps, for example, on land use (park, green area, under/super developed area).

### 368 13. GEOSPATIAL INFORMATION

#### Geolocation and landmark recognition

As is explained above, landmark image recognition is based on detecting unique objects in images and matching them against a knowledge base (collection of geotagged images). This is called landmark recognition with feature point matching, as interest points from a test image are matched to interest points in one or more training set images [401]. However, interest point matching in urban areas is difficult, since some structures (e.g., windows) may repeat frequently.

In non-landmark location recognition, image exact match on a training dataset may not occur or may not be reliable. So, Hays and Efros [285] find a probability distribution of images over the globe and base their strategy on that information, as well as on a dataset of over 6 million geotagged images (their knowledge base) from all over the world. Unknown images are described by selected image descriptors (e.g., color histograms, GIST) and compared to the big knowledge base. The top  $k$  most similar returned geotagged images are used to estimate the location of a given unknown image. Although this strategy will not be precise most of the time in finding an exact location, it will indicate roughly where an image was captured. For 16% of the time their method correctly predicted an image location to within 200km. Extensions of this approach rely solely on the text tags associated with the images [646, 591], or apply Hays and Efros' method to the visual content of images and to their associated user tags [236]. Gallgher et al. [236], besides using a collection of over a million geotagged photographs, also built location probability maps of user tags over the globe to study the picture-taking and tagging behaviors of thousands of users. Applying the local tag probability maps and image matching of Hays and Efros [285], Gallgher et al. showed that their method yielded improvements over pure visual content-based methods.

Similar strategies have been employed for multimedia retrieval. MediaEval<sup>7</sup>, a benchmarking initiative to evaluate a “new algorithm for multimedia access and retrieval” which is a spin-off of VideoCLEF, launched the Placing Task in 2010, along with other tagging tasks [370]. This task requires participants to automatically assign latitude and longitude coordinates to each of the provided test videos.

Participants in the Placing Task at MediaEval 2011 were allowed to use image/video metadata, audio and visual features, as well as external resources, depending on the run submitted. The organizer [528] of this task released a set of geotagged Flickr videos as well as the metadata for geotagged Flickr images, such as title, tags, and descriptions provided by the owner of that resource, comments of her/his friends, users’ contact lists, and other uploaded resource on Flickr. Data released included 10,216 geotagged videos, along with their extracted keyframes and corresponding pre-extracted low-level visual features, and metadata for 3,185,258 CC-licensed Flickr photos, uniformly sampled from all parts of the world. Test data comprised of 5,347 videos with its related metadata (without latitude and longitude information). Evaluation was based on the distance to the ground truth

<sup>7</sup><http://www.multimediaeval.org/>

### 13.4. MULTIMODAL RETRIEVAL FOR GEOGRAPHIC INFORMATION 369

geographic coordinate point, in a series of widening circles: 1 km, 10 km, 100 km, 1000 km, and 10000 km. Thus, an estimated location is counted as correct at a particular quality level if it lies within a given circle radius.

Although this year, a minimum of one run that uses only audio/visual features was required, most of the participants focused on modeling and solving the problem based on text metadata associated with the videos.

In 2010 the Placing Task data set was divided into 5091 videos for training (with the same additional Flickr photos) and 5125 videos for testing. There were three main approaches: (a) geoparsing and geocoding texts extracted from metadata assisted by a gazetteer of geographic name such as GeoName; (b) propagation of the georeference of a similar video in development database to the test video; (c) training set is divided in geographical regions determined by clustering or fixed-size grid using a model to assign items to each group. The model estimation was based on metadata text data and visual clues. The best result in 2010 for this task was accomplished by VanLaere et al. [646] by only using metadata for images and videos, combining approaches (b) and (c): first a language model identified the most likely area of the video and then the most similar resources from the training set will give the exact coordinates.

The only group in 2010 that also made use of visual features was Kelm et al. [333]; they reported that combining visual and textual results can yield better results than just relying on one of the modalities of information (just text or visual content).

#### Proposed multimodal geotagging framework

In order to begin tackling this problem we participated in Mediaeval 2011 and reported our idealized architecture for multimodal geocoding as depicted in Figure 13.13. The proposed architecture for dealing with multimodal geocoding is composed by three modules:

1. text-based geocoding is responsible for all text processing and GIR geocoding technique to predict location for videos' metadata;
2. content-based geocoding: this module will predict location based on visual similarity of the test images/videos against the knowledge database composed by development dataset; and
3. data fusion/rank aggregation: the rank aggregation-based module combines the geocoding results generated by the previous modules and gives the final result of the geocoding. The idea is to rely on text and image data whenever possible.

The final result is a combination of the results from each modality, treated by a data fusion module which deals with the geocoding results of textual (metadata) and visual (frames) parts of a video.

The challenge is to have good textual geocoding, good visual content-based geocoding, and an effective data fusion algorithm for geographic information.

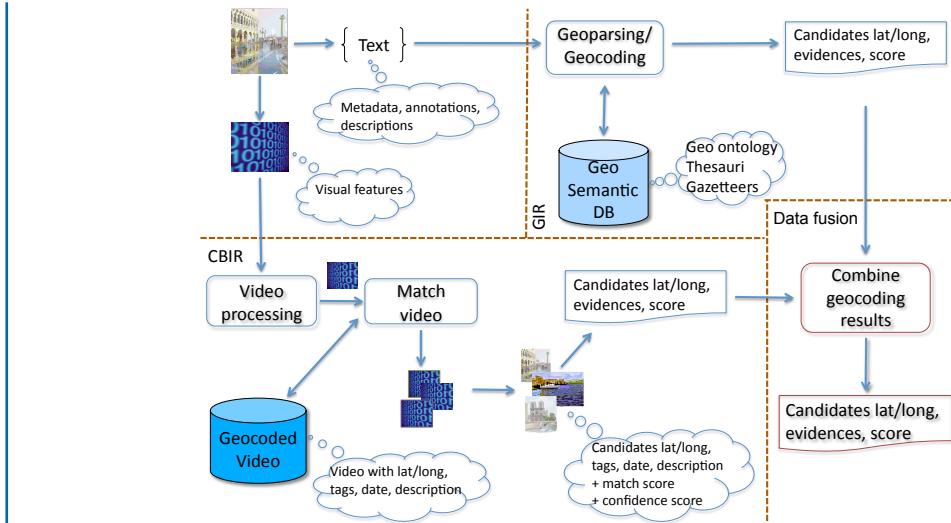


Figure 13.13: Multimodal geocoding architecture proposal.

For MediaEval2011, we focused on the second module, exploring a method to identify similar videos whose visual content indicates where those videos were filmed. Although it was legal to use all the metadata associated with the given video, such as descriptions and tags provided by users, we focused on geocoding based on visual features of the videos, since many others have studied geotagging based on textual evidence. The next step is to focus on data fusion for combining image and textual evidence in the context of geocoding digital objects.

### 13.5 RELATED WORK

Related works are cited throughout this chapter. This chapter relates to and employs Content-based Image Retrieval (CBIR), GIR, multimedial retrieval, data fusion, and ontologies.

There are two main projects that are worth to mention due to their impacts in IR for geographic information:

**ADEPT** – Alexandria Digital Earth ProtoType came from Alexandria Digital Library (ADL), which is a project led by University of California (Santa Barbara, USA) from 1995 to 2004. It is a distributed digital library comprising of a collection of geo-referenced material that could be searched in [312, 219]. Its search was focused on its digital library contents.

**SPIRIT** – *Spatially-Aware Information Retrieval on the Internet* project's aim was to develop a spatially-Aware web search engine. They tackled problems of assigning a footprint to web pages, indexing, searching, ranking, and also user interfaces for geographic queries and results [526, 319]. This project was led by Cardiff University (UK) from 2002 to 2005 and their focus were on Web documents.

## 13.6 FORMALIZATION

The content and discussion of this chapter are mainly related to 5S framework on these concepts:

**Space:** Besides the space of vector models of digital objects, geospatial information in DL will require the space that represents the Earth, a geographic coordinate system, since digital objects can be associated to that.

**Structures** are involved to store and use geo-ontologies, geographic objects, images and videos content, their metadata, and related information.

**Streams** of text, image, and video (which comprises of frames of images, associated sound and text) are mentioned in this chapter and we showed that they can be related to some place on Earth.

There is no need to extend the 5S framework formalization to house the geospatial information in DL.

Some topics to discuss here include: How does geospatial information discussed in this chapter relate to the 5S framework? Do those concepts need to be extended? How can we relate the extended set of concepts?)

## 13.7 CASE STUDY

(Some topics to discuss here include: CTRnet web archives and tweets. CTRnet applications. MediaEval2011. VT April 16 event and geocode locations.)

*[Section on this to be added]*

## 13.8 SUMMARY

In this chapter, geospatial information concepts were introduced along with the set of operations and possible queries for geographic objects.

The Geographich Information Retrieval (GIR) area was also presented in this chapter. That area comprises researches that aim to recognizing, querying, retrieving, and indexing geographical information present in documents, such as those in a DL or on the Web. Existing challenges concerning the implementation of GIR systems were introduced by a quick review of existing works for presenting , processing, and storing geographic data.

### **372 13. GEOSPATIAL INFORMATION**

Documents in both traditional DL and Web include text, images, and videos. Collections of images and videos are growing really fast due to numerous devices that take pictures or record videos. Furthermore, devices connected with GPS and camera, such as smart phones, that embed location data in pictures and videos' metadata are also growing. That scenario poses new challenges in different areas such as IR, multimedia, and computer vision. Some of those challenges were discussed in this chapter as well.

The 5S concepts highly referred in this chapter are space, structure, and streams

### **13.9 EXERCISES AND PROJECTS**

1. What is the best way to add a geospatial component to an existing digital library?  
Hint: Consider adding a module to handle geo-related 'knowledge', and supporting geo-enabled services (e.g., searching, browsing) that operate on all appropriate digital objects.

## CHAPTER 14

# Security

by Noha ElSherbiny

*Abstract:* Security is an important issue in digital library design. Security weaknesses in digital libraries, coupled with attacks or other types of failures, can lead to confidential information being inappropriately accessed, or loss of integrity of the data stored. These in turn can have a damaging effect on the trust of publishers or other content providers, can cause embarrassment or even economic loss to digital library owners, and can even lead to pain and suffering or other serious problems if urgently needed information is unavailable. In this chapter, security requirements that are essential for any digital library are explored along with models and mechanisms to provide them.

## 14.1 INTRODUCTION

Computer security is a broad term that refers to the protection of computer systems from threats. There are various domains of security, such as network security, information security, physical security, personnel security, operational security, and Internet security. In this chapter we are concerned with information security and logical aspects of security.

From the previous chapters we saw how varied and rich the content of digital libraries can be, as well as the complexity of their architecture. Some of the content stored in a digital library may be free for use, while other content is not. There are many different actors working with a digital library; each of these may have different security needs [118]. Thus, a digital library content provider might be concerned with protecting intellectual property rights and the terms of use of content, while a digital library user might be concerned with reliable access to content stored in the digital library. Requirements based on these needs sometimes are in conflict, which can make the security architecture of a digital library even more complex.

That architecture must be designed so that security concerns are handled holistically. A security system designer must view the whole architecture and consider all of the applicable security factors when designing a secure digital library. The nature of a security attack may differ according to the architecture of the digital library; a distributed digital library has more security weaknesses than a centralized digital library.

**374 14. SECURITY**

**Table 14.1:** Definition for 5 security services

Security Service	Definition
Authentication	The assurance that the communicating entity is who it claims to be.
Access Control	The prevention of unauthorized use of a resource (i.e., this service controls who has access to a resource, under what conditions access can occur, and what those accessing the resource are allowed to do)
Confidentiality	The protection of data from unauthorized disclosure
Data Integrity	The assurance that the data received is exactly as sent by an authorized entity (i.e., contains no modification, insertion, deletion, or replay)
Non-repudiation	Provides protection against denial by one of the entities involved in a communication of having participated in all or part of the communication

#### **14.1.1 BASIC CONCEPTS**

According to the X.800 recommendation for the security architecture for OSI [606], there are 5 main security services that are required to provide system security (see Table ??).

Support of the security of digital libraries is not limited to these 5 services. As discussed in previous chapters, digital libraries cover broad areas of information systems; therefore there are other services to consider. These are discussed below.

- **Availability:** a property that allows the system to be accessible and usable upon demand by an authorized system entity
- **Privacy:** is concerned with the collection and distribution of data and the legal issues involved.

- **Identity Management:** a process where every person/resource is assigned unique identifier credentials, which are used to control access to any system/resource via the associated user rights and restrictions of the established identity
- **Trust:** An entity is said to “trust” a second entity when the first entity assumes the second entity behaves exactly as the first entity expects.
- **Intellectual property rights:** is the protection of digital content from disclosure, unauthorized distribution, or copying. Some types of intellectual property rights include copyrights, patents, and trademarks.

All these security requirements are essential to protect digital libraries from different security attacks.

## 14.2 RELATED WORK

The CIA triad is a common model used to describe the security of information systems [529]. As shown in Figure 14.1, there are 3 properties in the triad: confidentiality, integrity, and availability. These 3 properties are necessary to provide a secure system. Security also



Figure 14.1: CIA triad

can be described using the DELOS reference model [98]. According to it, there are 6 main concepts in a digital library universe: content, user, functionality, architecture, quality, and policy. Each of these concepts has security issues that affect it. These issues are explored below.

### 14.2.1 CONTENT

The content of a digital library includes the information objects that a digital library provides to the users. Some of the security issues involved are integrity and access control. Integrity requires that each object/resource has not been altered or changed by an

### 376 14. SECURITY

unauthorized person. Access control encompasses two security requirements. The first is authentication, where the user must log into the system, while the second is confidentiality, which means that the content of an object is inaccessible to a person unless they have authorization. Not all digital libraries are free; often content is provided to digital library users for a certain fee, whereupon access control is needed to protect the content. Further, some content is inappropriate for some users, or targeted to particular user groups; there are a whole host of such reasons for access control. Logical attacks such as hacking and message tampering can affect the integrity and confidentiality of the content. Improved information access in digital libraries has raised many issues that affect the management of digital libraries. Content Management, or more specifically Digital Rights Management, refers to the protection of content from the different logical security attacks, as well as a range of issues relating to intellectual property rights and authenticity.

#### Digital Rights Management

DRM provides content protection by encrypting the content and associating it with a digital license [643]. The license identifies each user allowed to view the content, lists the content of the product, and states the rights the user has to the resource in a computer readable format using a digital rights expression language (DREL) or extensible Rights Markup Language (XrML), that also describes constraints and conditions.

There are 7 technologies used to provide DRM [186]. Figure 14.2 below summarizes the DRM components and supporting technology.

Each of these components involves mechanisms used to provide DRM:

- **Encryption:** Encryption techniques such as symmetric and asymmetric ciphers can be used to provide access control; public-key encryption is used in payment systems that control how and by whom the content is used. Symmetric ciphers using DES, 3DES, AES, and RC4 algorithms require the use of a shared secret key to encrypt data before it is sent. At the receiver's end the cipher text is decrypted using the same secret key. Symmetric ciphers depend on both the sender and receiver knowing the shared key. Asymmetric ciphers use a pair of keys, public and private, for each of the sender and the receiver. The public keys of both the sender and the receiver are known but the private key is kept secret. If encryption is performed using the public key then only the private key can be used for decryption and vice versa.
- **Passwords:** Stored strings must be matched by users desiring access.
- **Watermarking:** Characters or images are added to reflect ownership. Steganography is used to conceal data inside audio, video, or image [316]. Different watermarking techniques have different aims; some watermarks might be visible while others invisible. Some watermarks are reversible [432]; it depends on the desired use of the watermark and what is being protected.

Component	Protection Technology
Access and usage control	Encryption (e.g., symmetric, asymmetric), passwords
Protection of authenticity and integrity	Watermarks, digital signatures, digital fingerprints
Identification by metadata	Allows description of an object in suitable categories, covering the digital content, rights owner, and conditions.
Specific hardware and software	Includes all hardware and software used by the end-device through which the digital content is being played, viewed, or printed.
Copy detection systems	Search engines, which search the network for illegal copies and use watermarking.
Payment systems	Can be seen as a certain type of protection technology as it requires user registration, or credit card authentication, which also require a trust relationship between the content provider and the customer.
Integrated e-commerce systems	DRMS must include systems, which support contract negotiation, accounting information, and usage rules.

Figure 14.2: DRM components and protection technologies, adapted from [186]

- **Digital signature:** Asymmetric encryption can be used. Likewise, hash algorithms such as MD5 and SHA can be used to create a signature [606].
- **Digital fingerprint:** Digital fingerprints are a more powerful technique involving digital signatures and watermarking. The creator of the content creates a unique copy of the content marked for each user; the marks are user-specific, hence called fingerprints. Should a user illegally distribute the content, the creator can use search robots to find those copies [587].
- **Copy detection systems:** Search engines also can help locate such copied objects. Copy-detecting browsers can protect digital content too.
- **Payment systems:** Users must divulge personal information to pay for content. Installing payment systems can help protect digital content.

There is no standard mechanism for providing DRM, mainly due to the lack of regulations [118], however there are various systems and protocols introduced to provide content management and support fair usage policies.

### 14.2.2 PERFORMANCE

There is a tradeoff between security and performance. Nadeem and Javed [456] used a Pentium-4, 2.4 GHz machine running the Microsoft Windows XP operating system, to encrypt 20527 bytes to 2323398 bytes of data using DES, 3DES, and AES. For 20527 bytes of data it took 2 seconds to encrypt using the DES algorithm and 4 seconds to encrypt using the AES algorithm [456]. It can be seen that the more complex the encryption algorithm the longer it takes to encrypt the data. In another study, encrypting data with the RSA algorithm using a key size of 1024 took 0.08 milliseconds/operation on an Intel Core 2, 1.83 GHz processor under Windows Vista in 32-bit mode, while using a key size of 2048 took 0.16 milliseconds/operation [1].

### 14.2.3 USER

Users in a digital library refers to “the various actors (whether human or machine) entitled to interact with digital libraries” [98]. Digital libraries connect the different actors with the information they have and allow the users to consume old or generate new information. Security issues relating to the users of a digital library intersect with content issues discussed above. A main logical security issue relating to users and content is access control. Different access control requirements arise for distributed systems [633] to ensure both confidentiality and authentication:

- **Access control must be applied and enforced at a distributed platform level, so should be scalable and available at various levels of granularity.**
- **Access control models should allow a varied definition of access rights depending on different information and must be dynamic where changes to policies are easily made and easy to manage.**
- **“Access control models must allow high-level specification of access rights.”**  
[633]

Digital library users may need to be authenticated before they can access content in a digital library. Global/universal identification may not suffice. A service provider that provides content based on a non-identity based criteria like age will not benefit from global identification because there is no way to verify the authenticated user’s personal information. Usernames and passwords are not efficient ways to provide authentication.

One of the most widely used authentication protocols is Kerberos. It [467] is a client/server model, which secures communication with servers on a local network. Developed at MIT in the 1980s to provide security across a large campus network, it is based on the Needham-Schroeder protocol and has now been standardized and included in many operating systems such as UNIX, Linux, Windows 2000, NT, and XP. Kerberos is used as an authentication protocol in cases where attackers monitor network traffic to intercept

passwords. It secures communication, provides single sign on and mutual authentication, and does not send a user's password in the clear on an insecure network. An alternative solution suitable for digital libraries [675], is to represent information about an individual using credentials. Credentials are "abstract objects which contain statements expressing knowledge or information from a definite context." Credentials do not specify direct information about a client and their attributes; they describe the local environment and context in which the requests originate [116].

Digital credentials can be used as a means of authentication in providing DL access control [675]. Two agents can be used to assist in the management— a personal security assistant and a server security assistant— to manage digital credentials using a client/server model. The server must notify the client of the credentials required for the current request. The client must have some trust of the server to give its credentials, which raises privacy issues. The personal security assistant is used to obtain credentials on behalf of the client, store the credentials, parse and interpret the required credentials, and manage the acceptance policies [675]. A server security assistant is available to specify the credential acceptance policies and their usage.

There is a tradeoff between flexibility and security that must be considered when choosing an access control model, as is discussed below.

### **Access Matrix Models**

This conceptual model specifies the rights that each subject possesses for each object [633]. Actions on objects are allowed or denied based on the access rights specified. There are 2 implementations of the AMM:

- **An Access Control List provides a direct mapping of each object the subjects are allowed to access, and their usage rights (owner, read, or write).**
- **A Capability List defines the objects each subject is allowed to access and the usage rights.**

Access control lists and capability lists are not suitable for distributed systems. Their limitations lead to multiple problems [457]. ACL provides limited expressibility of policies. Any change in the policies will propagate in the system/application. Authentication in a system that uses ACL solely is a problem because using username and password in a distributed system is not practical. In a distributed system, administration of the system should be decentralized by delegation to reduce the overhead. The owner of the object specifies a policy in ACL. If an overall policy is specified by an entity higher than the object owner, then conflicts may occur in the access rights. The number of administrative entities in a distributed system can be very large. Not all the administrators may have trust amongst themselves, resulting in incorrectly defined policies. For example, admin A may

## 380 14. SECURITY

trust B but not C, however B may trust C. If A were to define a policy for B then it would be implicitly applicable to C, causing problems.

### Role-based Access Control

Role-based access control involves policies that regulate information access based on the activities the users perform. Such policies require the definition of roles in the system: “a set of actions and responsibilities associated with a particular working activity” [570]. Permissions are assigned to roles instead of individual users. Specifying user authorization involves 2 steps: first assigning the user to a role, second defining the access control that the role has over certain objects. RBAC is easier to manage and is more extensible than ACL. However RBAC does not flexibly handle constraints, where a user with a specific role may need specific permission on an object. An example of an RBAC architecture addressing key limitations is OASIS [35], for use in distributed systems. Role management in OASIS is decentralized and service specific. OASIS is integrated with an event-based middleware that notifies applications of any environmental changes. Roles are parameterized by applications and services to define their client roles, and to enforce policies for role activation and service invocation within each session. Role membership certificates (RMC) are returned to each user on successful login, to be used as a credential to activate other roles [35]. RBAC is suitable for use with digital libraries because it supports decentralized architectures and varying roles, however RBAC does not allow for the definition of different roles in a collaborative group.

### Task Based Access Control

The Task based access control model extends subject/object access control by allowing the definition of domains by task-based contextual information [633]. Steps required to perform the task are used to define access control; the steps are associated with a protection state containing a set of permissions for each state, which change according to the task. TBAC uses dynamic management of permissions. TBAC systems are limited to defining contexts in relation to activities, tasks, or workflow progress. Since it is implemented by recording usage and validity of permissions, therefore, TBAC requires a central access control module to manage permissions activation and deactivation in a just-in-time fashion.

### Team Based Access Control

RBAC does not address cases where group members of different roles want to collaborate in a single group. The TMAC model defines collaboration by user context and object context. “User context provides a way of identifying specific users playing a role on a team at any given moment” [633] while object context defines the objects required. TMAC offers the advantages of RBAC along with ability to specify fine-grained control on users and on object instances. A scalable access control data structure can be used with large

collections, applying concepts of team based access control, focusing mainly on the access control data structure, and employing an access control framework called Document Access Control Method (D ACM) with a Document Storage System (DocSS) [244]. D ACM allows the decentralized administration of privileges, the definition of different rule sets to control a single collection, and different delegation patterns as models. Current object access control policies use an array of rules to record the privileges each subject is allowed, with regard to each object. This is impractical to manage in the large data collections found in digital libraries. D ACM solves this problem by finding symmetries in a permission function to allow a brief expression without losing important distinctions.

#### **Content Based access control**

Another approach to access control models involves defining models according to content. This approach is applicable in digital libraries and distributed systems [7], where the access rights to the user are dynamic and may change with each login. Content based access control policies are very well suited for digital libraries and distributed systems. Recent research has proposed different models; most use digital credentials for authentication, but vary in the definition/storage of the policy. An important content based access control model [185] introduces a content-based access control system, Digital Library Authorization System, that utilizes the Digital Library Authorization Model (DLAM). Subject, object, and privilege sets cannot be used to define policies in digital libraries mainly because DLs are dynamic with large collections of data and subjects. It defines access control policies based on subject qualifications and characteristics. DLAM provides a means to specify the qualifications and characteristics of subjects. It uses content dependent and independent access control and allows the definition of policies with varied granularity.

#### **14.2.4 FUNCTIONALITY**

The concept of functionality encompasses the services that a digital library offers to its users. The minimum functions of a digital library include adding new objects to the library, or searching and browsing the library, as well as other functions relating to DL management. A security attack that can affect the functionality of the digital library is a Denial of Service attack, which can affect the performance of the system and prevent users from accessing the system.

#### **14.2.5 ARCHITECTURE**

Digital libraries are complex forms of information systems, interoperable across different libraries, and so require an architectural framework mapping content and functionality onto software and hardware components [98]. There are various models for architecture, e.g., client-server, peer-to-peer, and distributed. All these require the protection of the communication channels between 2 parties, where sensitive data might be transferred [344].

## 382 14. SECURITY

Securing the connections involves different layers – Internet, transport, or application layer – depending on the architecture of the system.

The distributed model is scalable and flexible. It is useful when building a digital library with changing content from different sources and offers potential for increased reliability. The security requirements for a distributed digital library are challenging, since the content and operations are decentralized. Fault tolerance and error recovery are issues that affect a distributed system. Replication is used to increase the availability of the system. While this approach solves problems with denial of service attacks, it complicates the protection of the content because one or more replicas of the content exist.

The client-server model does not have the same security problems as a general distributed model, however, it presents a major security weakness, the server being a single point of failure. Attacks can be concentrated on one server rather than on the multiple replicas of a distributed model. In Chapter 1, the 3-tier architecture of the DELOS reference model was explained. There are security issues relating to each of the 3 tiers; on the Digital Library (DL) tier issues of intellectual property rights, digital rights management, data confidentiality, and data integrity affect the digital content. On the Digital Library Systems (DLS) tier availability and access control are security issues because the DLS is the interface between the DL and the users. The user interface of the digital library, the DLMS, is the third tier; I suggest security issues such as access control and privacy of the data should be considered.

### 14.2.6 QUALITY

The content and behavior of a digital library is characterized and evaluated by quality parameters. Quality is a concept not only used to classify functionality and content, but also used with objects and services. Some of the parameters are automatically measured and are objective while others are considered subjective; some are measured through user evaluations.

### 14.2.7 POLICY

Policy is the concept that represents the different regulations and conditions that govern the interaction between the digital library and users. Policy supports both extrinsic and intrinsic situations [98], and their definition and modification.

Examples of security issues relating to policies include providing digital rights management, privacy, and confidentiality of the content and users, defining user behavior, and collection delivery.

An important security issue relating to policy is trust. For example, there may be various contributors to a digital library, each of them wishes to protect their content; if there is trust between the contributors and the DL managers then they will be more willing to share their content. There are 3 basic models of trust [64].

- **Implicit Trust Model:** In this model there is no explicit way to validate the credentials of the communicator. It is sometimes called the assumptive trust model. For example A receives an email from B, A knows both the email address and domain name, therefore A assumes that B is the sender and trusts B. However there is no way to prove that B actually sent that email and it was not a replayed message. In current electronic transactions, implicit trust is not a reliable model for use.
- **Explicit Trust Model** In the explicit trust model, credentials are used to validate the communicating entities. For example in Figure 14.3, A issues B a credential such as a username and password. B then uses this username and password to log onto the system. Here we have explicit trust, where A trusts B because of the credentials it just verified, and B trusts A the issuer of the credentials.

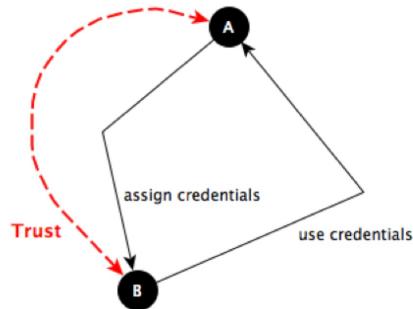


Figure 14.3: Explicit Trust

- **Intermediary Trust Model:** This model is commonly used in distributed and peer-to-peer systems, where the trust is “transmitted” through intermediaries. For example from Figure 14.4 we can see that A and B have explicit trust, while B and C have explicit trust, therefore a third trust is inferred. A and C have intermediary trust.

Digital libraries should be secure. This is an important quality that affects all aspects, as has been shown previously using the DL characterization of the DELOS Reference Model [98]. Many of the security issues discussed affect more than 1 concept, suggesting the overlapping of issues between digital library concepts.

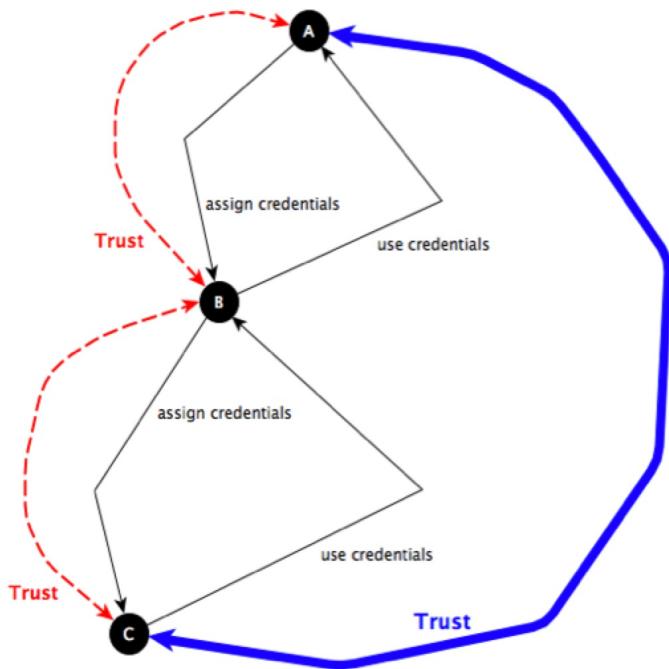


Figure 14.4: Intermediary trust model

## 14.3 FORMALIZATION

In the previous chapters, how the 5S can be used to define various aspects of digital libraries was explored. Having discussed the different security requirements that are needed in a digital library, we now look at how each of them maps to the 5S.

### 14.3.1 STREAMS

The streams model describes the different formats of the digital content that is to be stored in the digital library. As discussed in the previous section, publishers or suppliers of the digital content must have certain access rights defined for each digital object. A major security threat is the disclosure of content, meaning that a person has violated an access right defined for the digital object. To prevent this attack we need to provide data confidentiality where data is protected from unauthorized disclosure. Encryption of the data is a mechanism used to provide data confidentiality. As discussed in the prior section, there are various encryption methods available for use. Another security vulnerability in the streams model

is the modification of data; again digital content access rights might not allow a certain user to modify the contents of a digital object. Any change in the content would be a violation of the access rights defined. Therefore to detect modifications made to the digital object we must have data integrity, which is the assurance that the data received is exactly as sent by an authorized entity [606] (i.e., contains no modification, insertion, deletion, or replay). There are various methods to provide data integrity, some of which are hash functions such as SHA-1 and MD5, or message authentication codes (MAC) such as DAA and keyed Hash MAC (HMAC). These mechanisms can be used with encryption methods to provide data confidentiality.

#### 14.3.2 STRUCTURES

The structures model covers the organization of the digital content and its metadata in the digital library. Again disclosure is another security threat where the data objects may be revealed, violating a predefined access right; to combat this attack we can use each of these services on its own or a group of them together. Authorization is a desired service of the digital library: giving permission to use a resource by defining its access rights. Access control is another required security service; it is the prevention of unlawful use of a resource [606] (i.e., this service controls who has access to a resource, under what conditions access can occur, and what those accessing the resource are allowed to do). In the previous section, various access control models were described; any of these models can be used to provide access control. Another security vulnerability in the structures model is the modification of catalog data. The catalog data has all the information about the metadata; any change in the content would cause a variety of problems. An attacker might change the access rights of certain documents, causing a violation of the access rights defined by the owner of the resource. Therefore, to detect modifications made to the catalog data, we must have data integrity. Again, this can be provided by using hash functions, encryption techniques, or MAC. Another important feature that is required is data consistency, which is the integrity, validity, and accuracy of data between applications. In the structures model, organization tools are used to describe the structure in the digital library. An attacker may change the relationships between certain entities, which would make the model inconsistent. By applying the same mechanisms described for data integrity we can provide data consistency for the structures model. Illegal use of data is another security attack that can occur in the structures model. An attacker may have gained access to a resource by legal means, but then abuses his privileges by illegally using the data, such as making a copy of the document. Here an attacker would be violating his access rights; this can be prevented by providing access control as well as protecting the intellectual property rights of the data. A term is used to describe the protection of content from the different logical security attacks and issues relating to intellectual property rights and authenticity: digital rights management (DRM). In order to have digital rights management, authorization and privacy of the data

## 386 14. SECURITY

are implied. To provide DRM we can use a group of technologies such as encryption, passwords, digital signatures, and watermarking.

### 14.3.3 SPACES

The spaces model describes the user interface of the digital library and the index retrieval model. The index retrieval model is not being used in the generation of the DL; it is only described for purposes of documentation [251]. Disclosure is a major security concern in the spaces model; only authorized personnel should be able to view the user interface details. Modification of the data is also another concern; any changes in the model could cause major changes in the interface. Therefore data integrity, data confidentiality, and authorization are each a requirement. Mechanisms such as hash functions and message authentication codes could be used to provide data integrity, allowing us to detect any changes made to the data. Encryption can be used to provide data confidentiality, where all the data can be encrypted and stored along with authorization of the user before he/she is allowed access to the data; this can prevent disclosure of the data.

### 14.3.4 SCENARIOS

Scenarios describe how the digital library actors behave and how the services of the digital library are carried out. Another security attack is denial of service, which would prevent the digital library from providing any services to its users. Availability is a security requirement, that describes a system property that allows authorized users to access and use the system upon demand. There are various types of DoS attacks; pingflood attack involves sending large amounts of ICMP echo command (ping) data packets to the server to attempt to overload it. A counter mechanism requires a network administrator to obtain the IP address of the attacker and block access to the network. TCP smurf is another DoS attack. It involves the attacker communicating with the victim using the victim's IP address. This causes confusion on the victim's network resulting in a flood of traffic sent to the victim's network device. Firewalls can be used to prevent TCP smurf. A similar DoS attack is UDP fraggle, which is also used to confuse the victim's network, but using UDP. Again UDP fraggle can be prevented by having a firewall or simply by blocking any ports that could be used for fraggle such as port 7, echo port, or port 17. Distributed denial of service attack is more complex than the other attacks discussed; it involves ping flood but from various computers. The computers attacking might not be aware they are being used; a Trojan or a virus might have given a hacker control over the devices. There is no simple solution to overcome DDoS attack, but buying an intrusion detection system would help prevent the attack.

### 14.3.5 SOCIETIES

As discussed before, the societies model describes the entities that make up the community of the digital library. Masquerade is another security attack where an attacker falsifies his identity, pretending to be someone else. This is a major problem because certain users may be allowed access to certain content while others are not; if an attacker masquerades as an entity that is allowed to access certain content then a violation of the access rights of the content occurs. Authentication is the guarantee that the entity communicating is who it claims to be, thus preventing masquerade. Another form of masquerade that occurs on a lower level (Network Layer of TCP/IP model) is IP address spoofing. IP spoofing is the creation of Internet protocol (IP) packets with a fictitious source IP address to hide the identity of the sender or impersonate another computer system, thus possibly gaining access to confidential content. Providing authentication in the system by using packet filtering can prevent IP spoofing. A different security attack is misuse of privileges where a user violates the access rights of a digital object. This can be prevented by providing access control, using any of the methods discussed previously. Another security attack could be session hijacking, which is the use of a session to gain unauthorized access to information or services. Mainly, it refers to the theft of an HTTP magic cookie, which a user uses to authenticate to a remote server. Here authentication is an important feature but it is not enough to authenticate the user at the start of the session; authentication should continue throughout the session. A mechanism to prevent session hijacking is encryption of the traffic between the communicating entities, such as SSL. Authentication bypass is a security threat where an attacker could perform some action that is restricted to authenticated users without providing authentication. This could lead to various other vulnerabilities such as disclosure of protected data or modification of data. Authentication bypass is easy to avoid by providing authentication of the user. We need to ensure the credentials submitted are valid before performing any action. Mechanisms such as Kerberos and the X.509 Authentication service verify the user's identity once before the data exchange starts. Source/Sender non-repudiation is preventing sender or receiver repudiation. Repudiation is the “denial of by one of the entities involved in a communication of having participated in all of or part of the communication” [606] This is a problem when a user denies communicating in an e-commerce environment. Some digital libraries may require payment to access certain information; the payment process must be secure and must prevent any repudiation by the user. Non-repudiation is a desired requirement that can be satisfied by having users digitally sign any transaction. In the case of payment, an authorized third party may be used to handle the communications, such as PayPal [2].

Figure 14.5 shows the different security attacks that can occur at each of the 5Ss, what service is required to prevent or detect the attack, and what corresponding mechanism is required to provide that service.

388 14. SECURITY

5S	Security Attack	Security Service
Streams	<b>Disclosure</b>	Data Confidentiality
	<b>Modification of Data</b>	Data Integrity
Structures	<b>Disclosure</b>	Access Control + Data Confidentiality + Authorization
	<b>Modification of catalog data</b>	Data Integrity + Data Consistency
	<b>Illegal use of data</b>	Digital Rights Management + Privacy + Authorization
Spaces	<b>Disclosure</b>	Data confidentiality + Authorization
	<b>Modification of Data</b>	Data Integrity
Scenarios	<b>Denial of Service Attacks:</b> Ping flood/ TCP smurf/ UDP fraggle/ DDoS.	Availability
	<b>Disclosure</b>	Access Control + Data confidentiality + Authorization
Societies	<b>Masquerade</b>	Authentication
	<b>IP Spoofing</b>	Authentication
	<b>Session hijacking</b>	Access Control + Authentication
	<b>Authentication bypass</b>	Authentication + Access Control
	<b>Source/Sender repudiation</b>	Non-repudiation
	<b>Misuse of privileges</b>	Access Control

Figure 14.5: The possible security attacks that can occur at each of the 5S, the Ss are color-coded: Red for Scenarios, Blue for Streams, Green for Spaces, Orange for Structures, and Purple for Societies

The security services mentioned above are related to one another. For example, by definition, in order to have access control, one must have authentication of the user and confidentiality of the data. Therefore access control involves authentication and confidentiality. Other broad terms such as intellectual property rights include having copyrights on content or more broadly having digital rights management. Figure 14.6 shows a concept map of the different security issues of a digital library and how they all relate to each other. Secure digital libraries require data consistency of the content and the catalog of the digital library. Data consistency as defined before is the integrity, validity, and accuracy of data between applications. Here, data integrity is of the different digital library components and

is a security requirement that can prevent replay attacks, data insertion, data modification, and data deletion. Another security issue in digital libraries is availability, which is a security requirement used to prevent denial of service attacks. The possible types of DoS attacks that can occur on a digital library are distributed denial of service attack, ping flood, TCP smurf, and UDP fraggle. Intellectual property rights is a major security concern in digital libraries since the content stored on the digital library might not be for free; the creator of the content might wish to enforce certain access rights on the content. The content might have copyrights that determine how the content is to be used.

Digital Rights Management is a sub-category of intellectual property rights, which refers to the protection of content from the different logical security attacks and issues relating to intellectual property rights and authenticity. In order to enforce DRM, certain sub-requirements are needed, for example, digital signatures and access control. These can be used to preserve the authenticity of the object. Access Control as stated above is basically having data confidentiality and authentication of the user along with a series of access usage definitions for each of the objects. These definitions describe who can access a resource, under what conditions, and what can be done with the resource. As seen in the DELOS reference model an important aspect of digital library design involves the different policies, including the security policies. Security policies are the different regulations and conditions that govern how a system stores, manages, protects, and distributes sensitive information. There are various security policies in digital libraries that define the terms and conditions for use of the digital library; some policies may define the privacy issues relating to a digital library. This includes the privacy from the user's perspective as well as the collection and distribution of the data. There are various legal issues concerning privacy, especially the personal information being stored, e.g., in certain digital libraries that store sensitive personal information about people, their salaries and their income. DL designers must make sure that the personal information is confidential and not available to anyone, or may risk facing legal charges.

During the requirements gathering phase, various digital libraries were studied to understand the nature of their security requirements. In the CTRnet project (<http://www.ctrnet.net>), Professor Donald Shoemaker, from the Sociology Department at Virginia Tech, described the sensitive nature of information stored on CTRnet. Information such as survey results, which may include sensitive information such as the income of studied subjects, must not be revealed to just anyone. Certainly some members of the study group may view this information, however, the information must be stored in a way that would give an abstract view of the study findings without revealing sensitive personal information to someone who does not have the right to view the information. An example of privacy violation could result when personal information could be deduced after multiple queries are conducted, using the digital library search agent. All these issues are privacy issues that need to be addressed. An important aspect of privacy is authorization, which in

### 390 14. SECURITY

turn requires specifying access control. Another security issue that affects digital libraries is SPAM. During the discussion with the CTRnet project members, it was stated that SPAM was a serious security issue. The digital library offers forums for victims of crises or tragedies to exchange ways to deal with grief and recovery. The DL is free to any user; it is only required that they create an account the first time they use the DL. Spammers have been using the forum to publish adverts and other spam messages.

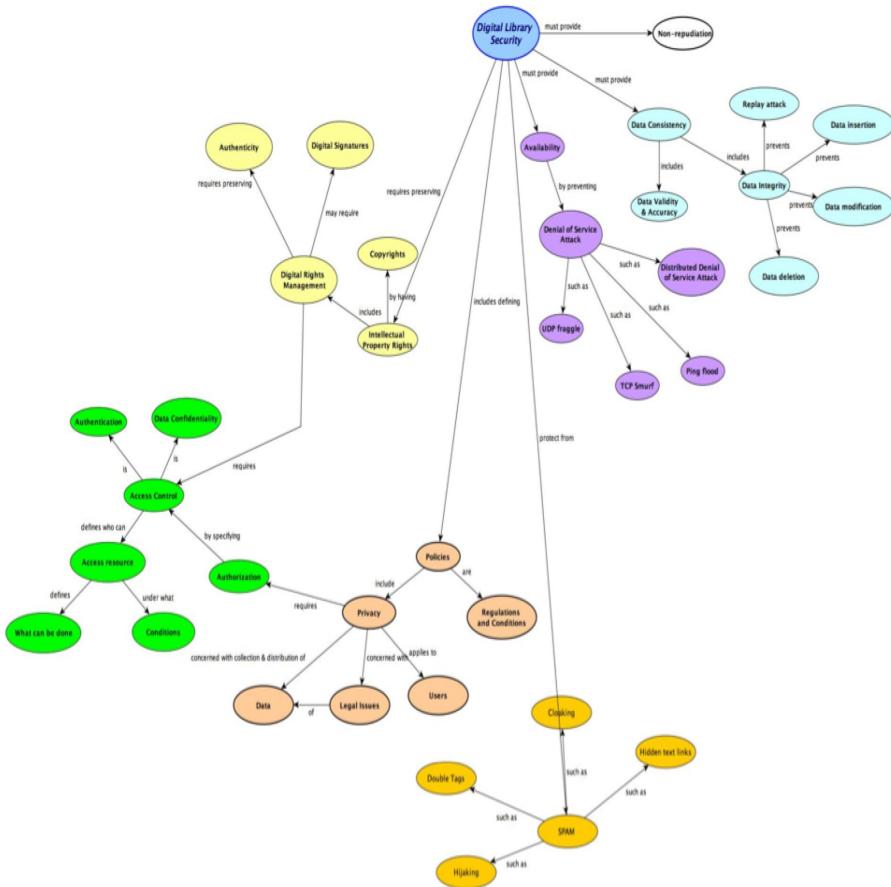


Figure 14.6: Concept map of the issues related to digital library security

## 14.4 CASE STUDY

In Chapter 12 the CINET project (<http://ndssl.vbi.vt.edu/CINET>) was described. CINET, cyber infrastructure for network science, is a web portal/service, which provides a common repository for managing both data and models, through a digital library that maintains metadata, and provides users with tools to analyze and run these models. CINET hides the details of computation and data management, thereby minimizing the learning effort required. It allows easy extension by integrating off-the-shelf network analysis suites for analysis and visualization; this means new algorithms can be added easily over time. CINET aims to foster research, teaching, and collaboration by building a broad user base, from multiple disciplines, including incorporation into courses on network science at many different universities.

CINET has 4 partners: Argonne, Indiana University, University of Houston, University of Chicago, and Virginia Tech, which provide clusters and grids, on which to run the computations. The architecture of CINET (see Figure 14.7) is distributed, where the computing resources are at the various partners' sites. The middleware manages the repository and is responsible for running the models and graphs using the selected computing resources. A user interface is available to facilitate the interaction with the system. Thus, there are various interfaces for the various users that use the system.

The 5S model can be used to describe the CINET digital library, along with the necessary security features.

### 14.4.1 SOCIETIES

The CINET societies involve various entities. CINET is used for educational purposes and therefore part of the CINET societies are the students that will use the portal as well as the professors and lecturers that will use the portal; they will have teaching materials to conduct certain courses. The security issues involved here are mainly identity management, authentication, and access control. Professors may wish to authenticate the students that use their resources, therefore authentication and identity management is required. The CINET system also should require authentication of the professors and students in order to give them access and privileges to appropriate resources.

CINET is maintained by administrators that ensure the services are running and who are improving the system; these administrators are also part of the society. They would need to authenticate their credentials in order to access the system, but also they would have different access rights depending on the role(s) that they play. This applies to the partners that offer their resources to run the computations; they are also part of the society.

The CINET repository is constantly updated by contributors; these contributors are varied; some of them require their content to be paid for, while others offer it to the public for free. These contributors are another example of societies. Again, access control and

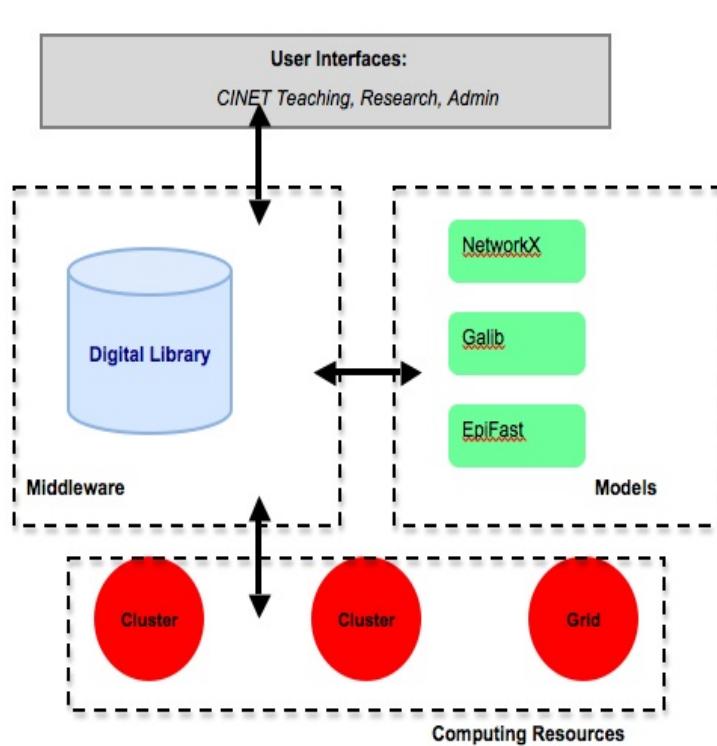


Figure 14.7: Architecture of CINET

authentication are an issue here but so also privacy; some entities using CINET may wish to remain anonymous.

Another example of societies is the algorithms and programs that are run on the CINET resources. These are not necessarily human but are still part of the societies model. Authentication is important to validate who can use the resources; there needs to be a mechanism to detect bots, SPAMMers, or other harmful programs that would use the CINET computational resources inappropriately.

Therefore, the security issues involved in the societies model are:

- access control and authorization
- identity management
- authentication
- privacy

#### 14.4.2 STREAMS

CINET stores in the repository graphs and models of networks; both (in encoded form) are examples of streams. Some of these graphs are not for public use, some are provided by researchers that wish to keep the models confidential. Confidentiality is an important security issue, to ensure the content is protected from unauthorized disclosure. Integrity also would be an important issue: to be able to detect any changes to the streams if they occur.

#### 14.4.3 STRUCTURES

The metadata of the models and graphs stored are examples of structures in CINET. This information may not be available to everyone. Again, like in the streams model, confidentiality and integrity of the content are important security issues to consider.

Digital rights management is also an important issue. The ownership, access rights, and copyrights of the content stored in the digital library must be preserved otherwise the “trust” between the content providers and the CINET system will be lost. Similarly, authorization would be required to ensure the access control of the content stored.

#### 14.4.4 SCENARIOS

CINET has various scenarios that might have vulnerabilities that need protection. We will explore the different scenarios and the security requirements for each.

- **Log-in:** Any society member in CINET will need to log into the CINET system before he/she is given access to the resources. In this scenario identity management and authentication are the security issues to consider.
- **Using Resources:** The societies in CINET use the different resources available; they can run tests and visualizations on the existing data using the CINET computational resources or they can use graphs and algorithms that they have written themselves. This scenario has various security concerns. First, there is access control: who has the right to access what content in the repository, as well as who has the right to execute on the grids. Second, there is the issue of trust: can the content the user is providing be trusted to run on the computational resources? A user can be an “untrusted” user and run a virus on the grid. Also the communications channel must be secure, to prevent any changes to be made to the data being sent to/from the grids; therefore data integrity is an issue to consider.

Therefore, the security issues involved in the scenarios model are:

- access control and authorization
- identity management

## 394 14. SECURITY

- authentication
- trust
- data integrity

### 14.4.5 SPACES

The availability of the portal is an important security issue in CINET. Different users have different views of the system; all the interfaces must be available for users to use at all times. CINET has a distributed architecture, with different computational resources at different sites. These resources need to be protected from denial of service attacks as well as other attacks on the communication channels.

## 14.5 SUMMARY

Digital libraries should be secure. This is an important quality that affects all aspects, as has been shown above using the DL characterization of the DELOS Reference Model.

The 5S framework supports Societies and their needs, covering all aspects mentioned above about Users and related Policies. Since Societies cover software actors, agents, components, modules, etc., this also encompasses related Architectural issues. Thus, security with regard to Societies covers issues like client/server, commerce, identity, peer-to-peer, privacy, rights, roles, teams, and trust. Scenarios cover functions, operations, requirements, services, and tasks. Examples include access, access control, authentication, browsing, copying, denial of service attacks, encryption, payment, recovery, searching, usage, and watermarking. Spaces cover distributed aspects, as well as representations related to 1D, 2D, 3D, and higher dimensional spaces. These include feature, measure, metric, probability, vector, and topological spaces. They are used throughout computer and human systems. Structures cover all types of organization, including data structures and databases, with lists (e.g., access control or capability), graphs, and networks. Structures are overlaid on other constructs in the 5S framework, especially on Streams. Thus, documents are structured streams, while protocols involve scenarios applied to structured communication streams. Structures and Streams cover all types of content, and the many security issues related, including digital rights management, fingerprints, and watermarks.

Clearly, DL security support can be complicated, but the above discussion should help readers organize their thinking and make sure that DL systems meet security requirements.

## 14.6 EXERCISES AND PROJECTS

1. What are some problems with taking an existing digital library and converting it so it will be completely secure?

## CHAPTER 15

# Text Extraction

by Sung Hee Park, Venkat Srinivasan, and Pranav Angara

*Abstract:* To support many digital library activities, it is useful to extract data, information, and knowledge from text. Text processing (including tokenization), natural language processing, and machine learning are key technologies involved. When one begins with large documents, document segmentation also is a necessary precursor, and can directly address needs for extracting images and captions. A case study, illustrating the promise and complexity of the service, describes reference string parsing. This requires feature extraction, training, and classification of extracted entities.

## 15.1 INTRODUCTION

### 15.1.1 RATIONALE AND SCOPE

Text extraction is useful in supporting many digital library activities. The term *text extraction* refers to identifying entities of interest from character strings found in any media type, such as text documents, video sequences, or images. However, in this chapter we focus on text information extraction from documents. Since documents may vary in size, so too may the size of what is extracted: from a big chunk of text, e.g., chapter, section, or paragraph, to a word or part of sentence, e.g., a particular part of speech (POS, such as a noun or verb) or a named entity such as a person, institution, location, time, date, or monetary amount.

### 15.1.2 PATTERN RECOGNITION, CLASSIFICATION, AND STRUCTURING

Text extraction is related to topics such as 1) *pattern recognition*, 2) *classification*, and 3) *structuring*. *Pattern recognition* is a research area that focuses on finding repeated patterns in the input. To solve this pattern recognition problem, a classification technique is sometimes used. *Classification* is a technique that assigns input elements to a corresponding specific class. Classification can be largely categorized into general classification and sequence tagging.

Additionally, there is a *structuring* concept in document processing, analysis, and engineering. *Structuring* is a text processing technique, which tags a plain text (an input) with an entry from a predefined tagset. Examples of such tags or labels are based on allowable parts of speech, or on the set of tags in a markup language like HTML.

### 396 15. TEXT EXTRACTION

These three practices describe some of the activities that are closely related to text extraction. For example, structuring helps with breaking text into pieces, which makes segmentation easier. Further, in some cases, structuring works down to a fine level, which is much the same as text extraction, though it is inside a document, and extraction may need to later separate out tagged items. For another example, text extraction often seeks to find character strings that fit into a particular pattern, like a phone number, address, or person name; pattern recognition applies in such cases. Likewise, classification often is a key part of text extraction, since identifying a character string of interest frequently is based on it fitting into a class or category of interest.

#### 15.1.3 PROBLEMS AND APPLICATIONS

Text extraction can be defined as a classification problem in which the extracted text should be tagged with a predefined tagset. One application of text extraction to digital libraries, an example of general classification, is document segmentation. Three other such applications, exemplifying sequence tagging, are part of speech tagging (POS), bibliographic information extraction, and reference metadata tagging. Information extraction can be applied to digital libraries, such as to extract information from digital objects that can be added to metadata records, as shown in Figure 15.1.

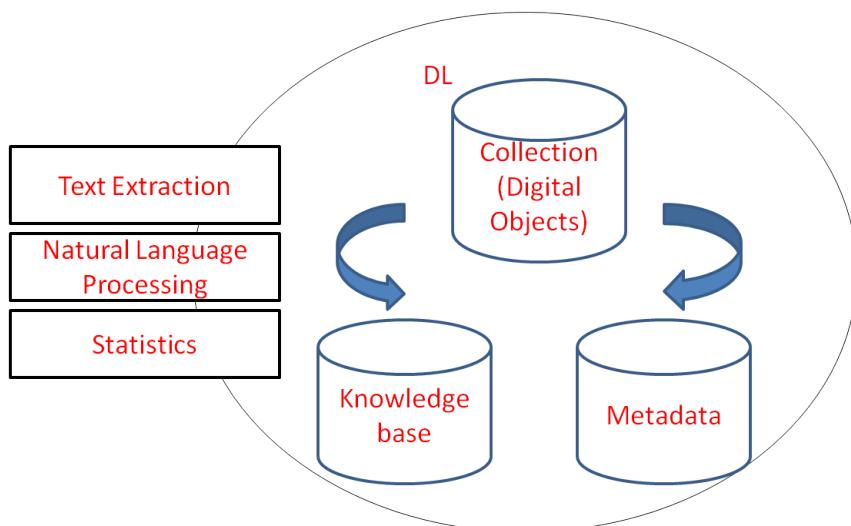


Figure 15.1: Text extraction in digital libraries

## 15.2 RELATED WORK

General Information Extraction (IE) techniques from documents have been widely researched. Naomi Sager directed work on an early IE system in the Linguistic String Project,

Table 15.1: Comparison of previous extraction approaches

Approach	Author & Year	Uns/Su
Rule-based	Day et al. (2006) [145]	S
	Cortez et al. (2007) [126]	U
	Afzal et al. (2010) [10]	U
Machine learning	Councill et al. (2008) [130]	S
	Hong et al. (2009) [293]	S
	Hetzner (2008) [291]	S

focused on the medical domain [562]. The Message Understanding Conference (MUC), sponsored by the U.S. Defense Advanced Research Projects Agency (DARPA), encouraged IE research from 1987 to 1998 [266]. At MUC-7, there was evaluation of extraction of useful information from news messages about airplane crashes and rocket/missile launches. MUC encouraged a focus on extracting four types of elements: named entities, co-references, template elements, and template relations. Subsequently, the Automatic Content Extraction (ACE) evaluation project was organized by the National Institute of Standards and Technology (NIST) from 2000 to 2008. An aim of the ACE program was development of technologies that extract entities from language data and then infer relations among them. Those two programs, MUC and ACE, also contributed to the development of a variety of indicators for deep evaluation. [498]).

In recent years, text extraction has become a popular tool. For example, a great deal of knowledge is being extracted automatically from the WWW [178].

In this section, we review prior work regarding text information extraction in terms of: 1) classification methods and 2) features in classification. In the next section, we look into 3) the knowledge bases that have been studied so far.

### 15.2.1 ALGORITHMS

Much research has addressed improving canonical representation extraction performance. Such studies can be categorized into two types of approaches: 1) Rule or other knowledge-based approaches [145, 126, 10] and 2) Machine learning approaches [130, 293, 291, 691]. Table 15.1 summarizes prior work, making clear which approaches are supervised vs. unsupervised.

#### Rule/Knowledge Based Approaches

Rule based approaches to canonical reference representation extraction are classification methods that exploit rules to mark tokens in the questioned sequence with proper semantic

### 398 15. TEXT EXTRACTION

labels. These approaches are particularly appropriate in situations where human experts in the specific area also would apply rules.

Some prior research has successfully applied rule based approaches to reference metadata extraction. However, that has been done in a limited problem space (e.g., less than ten reference output styles, two or three disciplines). For example, Day et al. [145] adopted a rule-based approach using a knowledge representation frame, INFOMAP, with respect to six output styles. Their frame describes the layout appearance of semantic labels such as *author*, *title*, *publisher*, and *year*. Ding et al. [158] used different templates designed to deal with the specific citations from digital contents.

Cortez et al. [126] proposed a knowledge-based approach for extracting citation metadata in a flexible way called FLUX-CiM, using blocking, matching, binding, and joining processes. Unlike other knowledge/ontology based approaches such as [145][126], this approach used a knowledge base to gather frequencies of terms that occur in each field like *authors*, *titles*, *journals*, etc. To evaluate the effectiveness of the method on output style-free extraction, three disciplines were used: computer science, health science, and social science. However, term frequency, which they used as a primary feature, is likely to be dependent on discipline. In addition, Embley et al. [175], used a conceptual model like Ding's template based approach, not only in citation parsing but also in general information extraction.

A general disadvantage of rule based approaches is that it is not easy to extract rules. Although previous research has shown that they are effective in the case of limited numbers of output styles and disciplines [145, 126], they are not easily adapted to our problem; we require solutions that are discipline-independent and that have a style-free complexity.

#### Machine Learning Approaches

Surface and semantics mapping problems can be viewed as instances of a sequence labeling problem in a natural language processing context. Figure 15.2 shows a concept map for machine learning based text extraction.

Machine learning approaches have used for sequence labelling. Examples include: hidden Markov models (HMM) [291], conditional random fields (CRF) [293, 130, 510, 548, 395], and Support Vector Machines (SVMs) [275, 469].

Machine learning approaches generally fit into two main categories: *kernel function based* such as support vector machine, and *probabilistic graphical models* such as CRF, HMM, and the Maximum Entropy Markov Model (MEMM). Fundamentally, SVM is a binary classifier, but it has been extended to solve the sequence labeling problem. Thus, *SVM<sup>struct</sup>* is one of the support vector machine implementations for sequence labeling [315]. Probabilistic graphical models can be grouped further into *generative models* and *discriminative models*. A *generative model* is a probabilistic graphical model in which an input sequence generates an output sequence. Generative models calculate a joint probability  $P(X, Y)$ , whereas a *discriminative model* directly calculates a conditional probability of

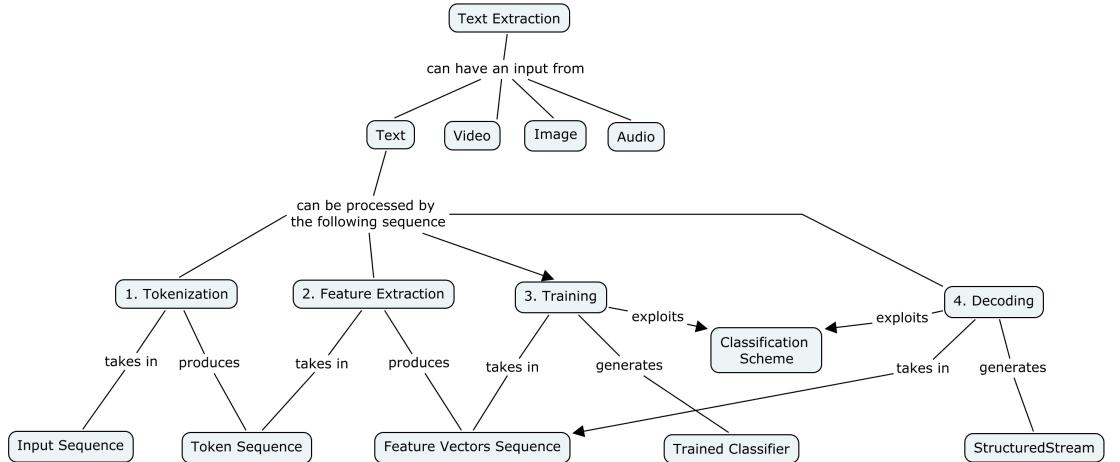


Figure 15.2: Text extraction from the 5S perspective

$P(Y|X)$ , where  $X$  is an input sequence and  $Y$  is an output label sequence. An example classifier for a single classification is a naive Bayesian network. HMM [527] is a typical generative model for a sequence data segmenter and labeler. On the other hand, a *discriminative model*, e.g., CRF, directly calculates the conditional probability of  $P(Y|X)$ , where  $X$  is an input sequence and  $Y$  is an output label sequence.

All machine learning techniques need to learn from a training dataset. Since these sets are labeled by human effort, this process is called supervised learning; it often is both time consuming and expensive. To alleviate this burden in supervised learning methods, some efficient methods called semi (weak)-supervised learning have been proposed and evaluated [417, 314, 388]. In this chapter, machine learning approaches will be the main focus.

### 15.2.2 FEATURE SELECTION

With classification methods, features are considered to play a critical role, so that it is possible to discriminate between data belonging to one class, and data belonging to other classes. Novel and effective features have been proposed and utilized.

In an open source reference parsing tool, *ParsCit*, Councill et al. [130] used 23 features named Token identity (3 features), N-gram prefix/suffix (9 features), Orthographic case (1 feature), Punctuation (1 feature), Number (1 feature), Dictionary (6 features), Location (1 feature), and Possible editor (1 feature). *ParsCit* is currently used in CiteSeerX. In similar fashion, in the lightweight real-time reference string extraction & parsing system called *FireCite*, Hong et al. [293] used a set of ten features, which are categorized into lexical (dictionary) features, local features, contextual features, and layout features.

Table 15.2: Features for canonical representation extraction

Features	Description	References
Local features	Non-lexical information about the token	[333, 130, 293, 701, 691, 510]
Lexical features	Information about the meaning of the words within the token	[130, 293, 701, 691, 510]
Contextual features	Lexical or local features of a token's neighbours	[130, 293]
Layout features	Relative position of a word in the entire reference string	[130, 691, 293, 510]

Similarly, Zou et al. [701] used 14 binary features consisting of three dictionary features (Author Name, Article Title, and Journal Title), plus an additional 11 binary features, describing local and orthographical information (e.g., pagination pattern, name initial pattern, four digit year pattern, etc.). Yu and Fan [691] used 15 features in total: nine local features (ALLCHINESE, CONTDIGITS, ALLDIGITS, SIXDIGITS, CONTDOTS, CONTAINS@, SINGLECHAR, NAME, EMAIL), three layout features (LINE\_START, LINE\_IN, LINE\_END), and three lexicon features (FAMILYNAME, AFFILIATION, ADDRESS) for metadata extraction from Chinese research papers. On the other hand, Peng et al. [510] investigated state transition, unsupported vs. supported, local, layout, and lexicon features.

To ensure beneficial use of features, automatic reference metadata labeling methods have used the following four types of features: 1) local features (dictionaries), 2) lexical features, 3) contextual features, and 4) layout features. Comparisons of existing features are described in Table 15.2.

### Local Features

*Local features* represent orthographic information about a token (e.g., a single capital letter surrounded by spaces, acronyms, and numbers). For example, among the local features, one single capital letter followed by a dot would appear to be the initial of a middle name or a first name in the *author* field, while numerals would appear in the *pages* or *year* attributes.

### Lexical Features

*Lexical features* represent information about the semantic category of a token. For instance, New York is a city name and May is the fifth month of a year. A lexical feature of the token New York can be LOCATION and similarly a lexical feature of May can be MONTH.

### Contextual Features

*Contextual features* are state transition features. From a probabilistic perspective, they can be defined as the probability of transition from a state at time  $t - 1$  to a state at time  $t$ , i.e.,  $P(y_t|y_{t-1})$  where  $y_{t-1}$  is the state at time  $t - 1$ , just before the state at time  $t$ . For example, the probabilities will differ in the two cases where the current state lexical feature is *TITLE*, and either the previous state is *AUTHOR* or *YEAR*.

### Layout Features

*Layout features* reflect the relative positions of a word in a reference sequence. They are similar to contextual features, in terms of structural information. But those features encode physical position whereas contextual features are concerned with semantic transitions. Layout features can be binary features or can have real values between 0 and 1.

## 15.3 FORMALIZATION

In this section, we give informal and formal definitions related to text extraction, and explain how new terms relate to the 5Ss.

### 15.3.1 INFORMAL DEFINITIONS

Text extraction is a pipeline service which produces structured streams from plain streams, using spaces and structures.

**Societies:** First, a specific society may set some constraints on the streams and structures. For example, university librarians are interested in ETDs and references in them, as well as classification schemes for bibliographical information found in such scholarly documents. The societies that are likely to be interested in text extraction research include those working on: 1) natural language processing, 2) pattern recognition, or 3) artificial intelligence.

**Streams:** Two main types of streams are of interest: 1) video/image/text documents, and 2) general structured streams. For more detail definition in structured streams, refer to 8.3.

**Scenarios:** From the scenarios point of view, information extraction from text consists of a pipeline of scenarios: 1) tokenizing, 2) feature extracting, 3) training, and 4) tagging/classifying/segmenting.

**Structures:** For each domain of interest, a **classification scheme** is necessary, so it can be used to guide conversion of a plain input stream into a suitably structured stream.

**Spaces:** To classify the segments of the input stream using proper labels, digital libraries may use metric spaces to calculate, for each label, the similarities between a centroid or other representative of the training data, and an unknown item of data. The following types of spaces can be used: probabilistic or vector.

The relationship among informal definitions is illustrated in Figure 15.3.

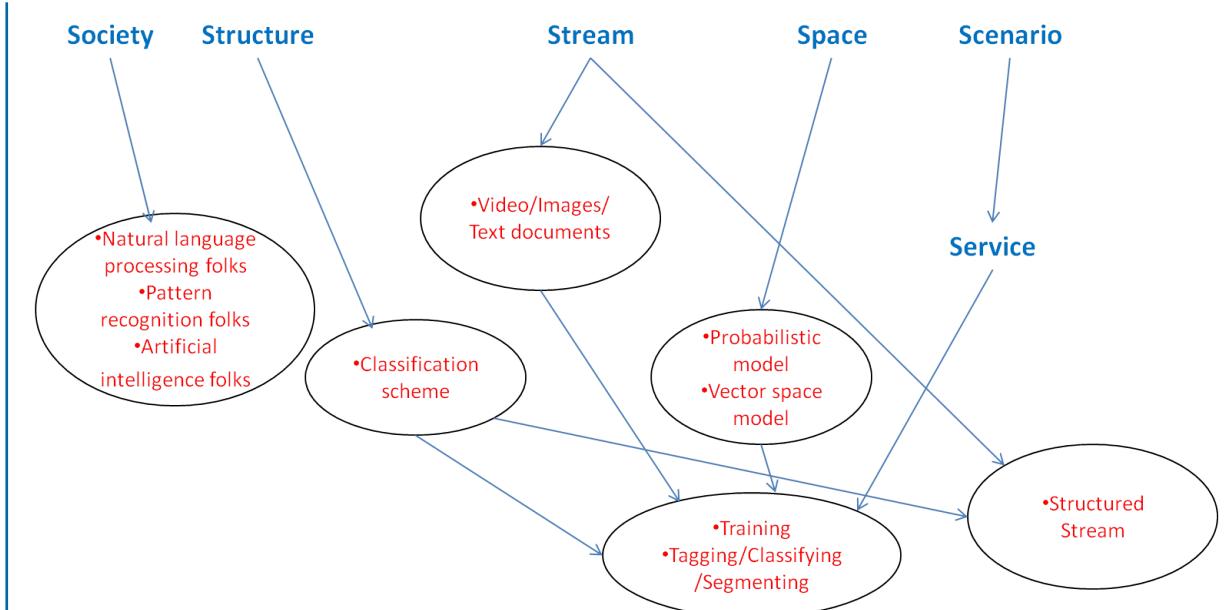


Figure 15.3: Text extraction from the 5S perspective

### 15.3.2 FORMAL DEFINITIONS

#### Classification Schemes

**Definition 15.1:** A **classification scheme** is a structure.

A classification scheme can be described using a set of labels, assigned to a graph representation of the relationships between the concepts and categories in a specific domain. For example, taxonomies and ontologies can be considered as classification schemes. In the case of text information, the semantics of a classification scheme indicate the category or categories to which an information unit should be assigned. For example, parts of ETDs can be described using a logical structural classification scheme made up of chapters, sections, subsections, paragraphs, etc.

#### Extraction and Segmentation

Text extraction consists of a pipeline of general processes for text processing, such as tokenizing, feature extraction, training, and extracting.

**Definition 15.2:** **Tokenization** is a scenario made up of transition events that begin from a state of having a *character sequence* and end with a state of having a *sequence of tokens stream*.

Tokenization divides an input text stream into smaller tokens. A token generally is a word. A tokenizer identifies tokens between delimiters such as blank, comma, semi-colon, colon, or space, and separates the tokens from each other.

**Definition 15.3:** **Feature extraction** is a scenario made up of transition events that begin from the state of having a *token sequence* and end with a state of having a *token feature sequence*.

Feature extraction generates feature vectors, considering each token of a token sequence stream. In some cases, feature selection (Def. 8.1) may be employed to reduce the number of features, so that the most important ones are employed.

**Definition 15.4:** **Text segmentation** is a service that takes in a stream, and produces a structured stream.

**Definition 15.5:** **Text extraction** is a service that takes in a stream, and produces a substructured stream.

This service usually consists of a set of scenarios such as *training* defined in (Def. 8. 5) and *decoding*, each of which is composed of a sequence of scenarios, such as *tokenization* (Def. 15.2), *feature extraction* (Def. 15.3), and *training/decoding*. Prior to feature extraction after tokenization, **stop word removal** and **stemming** may be performed.

## 15.4 CASE STUDIES

In this section, three case studies are described. Document segmentation is described in the first subsection below, in terms of dividing a long (multi-page) document into small parts, e.g., chapters and sections. In the second subsection below, reference section extraction is explained; this involves extracting the reference section from a text document. For more details, please see [605] and [498], respectively. The third one is related to sequence tagging or named entity recognition. For this case, refer to the geoparsing in th section 13.3.2.

### 15.4.1 DOCUMENT SEGMENTATION

We used several open source tools in order to segment documents and extract images (see Table 15.3). In the following subsections, we discuss our methodology in detail. The various steps are shown in Figure 15.4.

#### Extracting Chapters

In order to get to the text from ETDs, we used pdf2xml. It decodes the structure of PDF documents, and produces very fine grained font related information (metadata) about every token (word) that occurs in a PDF file. Sample metadata produced for an example token in a PDF file is shown in Figure 15.5. As can be seen, this output is not readily usable. Instead of producing lines or paragraphs that occur in PDF files as output, it generates tagged tokens (words). A page in a PDF file is treated as a co-ordinate space, with the

## 404 15. TEXT EXTRACTION

Table 15.3: Open source software used

Tool	Source	Function
pdf2xml [7]	SourceForge	Extract font related metadata
pdfimages	XPDF (Linux)	Extracts images (in PPM PBM format) from PDF files
pnmtojpeg	Linux utility	Converts PPM and PBM images into jpeg

origin (0,0) being at the top left corner of the page. Every token is encoded as a point in XY space, relative to the origin (as seen in ‘x’, ‘y’ in Figure 15.5).

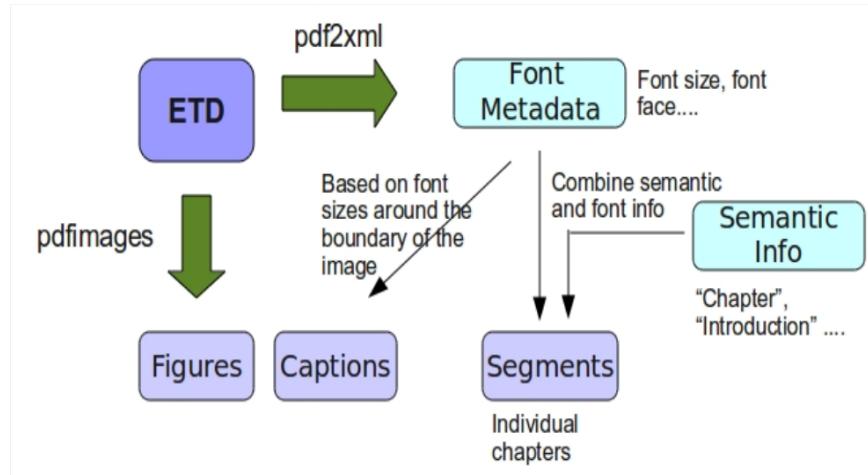


Figure 15.4: Various steps in text and image extraction

In order to identify chapter boundaries, we first wrote parsers to decode the pdf2xml output and to identify Title Case (TC) tokens.

```

<TOKEN sid="p1_s8" id="p1_w4" font-name="liberationserif" symbolic="yes"
bold="no" italic="yes" font-size="9" font-color="#000000"
rotation="0" angle="0" x="121.843" y="36.381" base="44.4" width="19.485"
height="9.963">name</TOKEN>

```

Figure 15.5: XML metadata for the token ‘name’ occurring in a PDF file

**Table 15.4:** Major reference styles used in ETDs

Style	Example
Bracketed	[Fox2011]
Numeric	[1]
Bracketed	(Fox2011)

We define TC tokens as those tokens that are likely to be chapter titles. ETDs, though, commonly contain fonts with heterogeneous sizes and characteristics. Hence, identifying TC tokens is not straightforward. In order to do this, we detect candidate TC tokens that occur near the top of a page, and use these to identify chapter boundaries. Often, ETDs also have one or more words like “Introduction”, “Summary”, “Chapter”, “References”, etc. in chapter headings. We use this cue also to identify TC tokens.

#### Extracting Images and Captions

In order to extract images from PDF files, we make use of pdfimages, which comes bundled in Linux, as part of the XPDF package. We process the PDF file page by page, and extract images that occur in each page. We are also interested in extracting image captions. Hence, once a page is found to contain image(s), we extract the XML metadata information for the page using pdf2xml. Using this metadata information, we identify the XY location of the image(s) on the page and identify small sized texts in the neighborhood of the images as possible image captions. The images extracted by pdfimages, however, occur in PPM, PNM, or VEC formats. In order to convert them to JPEG (for ease of display on the web), we use pnmtojpeg.

#### Extracting Other Miscellaneous Information

As an add on, we also attempted to extract individual entries in the bibliography section and their corresponding references within the body of the ETD. We identified 3 major styles used in ETDs (see Table 15.4), and wrote parsers to scan through the body of the ETD to identify locations of citations to references.

#### Web Prototype Design

We developed our web prototype using the content management system Drupal [161]. We used several Drupal modules, like those for taxonomy, image gallery, etc. (see Table 15.5) in order to achieve the desired functionality. Users can browse by chapter, page, figure, reference, etc. (see Section 15.4.2).

Table 15.5: Drupal modules

Module Name	Function
Views	Image Gallery
Taxonomy	Taxonomy for browsing by chapter
Vocabulary	Creating navigation block for Taxonomy

### 15.4.2 RESULTS

#### Web Demo

Our web demo allows for browsing by separate streams (chapters, images, etc.), as well as presents a unified view of the entire document. The screenshots in Figure 15.6 show the use of the taxonomy and image gallery features. The web demo can be accessed at <http://zappa.dlib.vt.edu/etd/>.

#### Evaluation

One critical issue in the development of such a system, besides its usability, is the performance of the backend methods. In order to understand how accurate our text, image, and caption extraction methods are, we ran several experiments. To evaluate the accuracy of the document segmentation technique, we randomly selected 10 ETDs each from the Engineering, Arts, Business, and Mathematics disciplines from the Virginia Tech ETD collection. Our algorithm achieved an accuracy of 70%, 50%, 70%, and 60% respectively on this data set. We consider the algorithm to be accurate when it successfully identifies every single chapter boundary for an ETD - so, for example in this case, our algorithm perfectly identified every single chapter boundary in 7 out of the 10 Engineering ETDs we had selected. If we relax the criterion a little bit, and allow for identification of some but not all chapters in ETDs, the accuracy goes higher, although more experiments are needed to get a good estimate. Nevertheless, out of 40 ETDs used in the experiment, our algorithm perfectly segmented 25 of them. Evaluating the performance of our image and captions extraction tool is a little harder. One problem is that pdfimages extracts certain extra ghost images from ETDs. These are just small sized spurious PPM or VEC files, and without visual inspection, it is hard to tell whether it is a real image extracted from the ETD or just some ghost image. We observed, however, that in the case of ETDs in our collection, the ghost images are mostly less than 1KB in size. So, in order to perform a reasonable evaluation of our method, we ignore all extracted image files that are less than 1KB. Another problem is that pdfimages segments the images themselves under certain circumstances. For example, when the image is a flowchart, pdfimages extracts certain segments separately, and returns multiple images instead of the entire flowchart as a whole. To get a rough estimate of the

## 15.4. CASE STUDIES 407

performance of our image and caption extraction tool, we selected 10 ETDs at random from the Virginia Tech ETD collection, and extracted images (and captions) from them. These ETDs were found to contain a total of 91 images, out of which we were able to recover 36 images. The rest of the images either could not be recovered at all, or were recovered only partially or were segmented. Of these 36 images, we were able to extract captions for 24 of them. More experimentation (including user studies) is needed to get better estimates.

The screenshot displays a web-based interface for a digital dissertation. The interface is organized into several sections:

- Contents**: A sidebar on the left containing a tree view of the dissertation structure:
  - Chapters: Chapter 1, Chapter 2, Chapter 3, Chapter 4, Chapter 5, Chapter 6, Chapter 7, Chapter 8, Chapter 9
  - Figures: Figure 1, Figure 2, Figure 3
- ETD CONTENTS**: A section titled "ETD FIGURES GALLERY" showing thumbnails for figures 1.1 through 8.1.
- Figures**: A callout pointing to the figure gallery.
- Context Sensitive Interaction interoperability For Distributed Virtual Environments**: The main content area, which includes:
  - Posted: Tue, 06/14/2011 - 12:37 by Mohamed
  - Author: Hussain Mohammad Ahmed
  - Title: Context Sensitive Interaction interoperability For Distributed Virtual Environments
  - Date: May 2010
  - Type: Dissertation
  - Department: Computer Science
  - Advisor: Graeme
- Metadata**: A callout pointing to the metadata section.
- HMA\_Chapter 1**: A section titled "HMA\_Chapter 1" containing:
  - Posted: Wed, 05/18/2011 - 11:45 by Mohamed
  - The main premise of my research is that it is feasible to abstract input devices and interaction tasks and provide context sensitive mapping between devices and tasks in large scale distributed virtual environments (DVEs) without compromising performance. In this dissertation, entitled Context Sensitive Interaction Interoperability for Distributed Virtual Environments, I elaborate on this premise and describe how to abstract input devices and interaction tasks in order to achieve suitable many to many mappings between devices and tasks given their descriptions. Using this abstraction I proceed to show how context sensitive mapping is done in a scalable way, without compromising system performance, to facilitate integration in DVEs with a large user base. This dissertation demonstrates then how abstraction, mapping and context awareness are all achieved in a distributed manner to ensure scalability and provide support for heterogeneous platforms.
  - The focus of this dissertation is on DVEs given the scalability demands of these environments with multimillion sized user bases, thousands of concurrent users and distributed infrastructures. More specialized types of applications and virtual environments (VEs) such as Collaborative Virtual Environments (CVEs), Serious VEs, Computer Games and others can be included in the presented framework.
  - With an ever growing number of innovative input devices the motivation is that users should be able to interact with applications using what they like from that vast variety of input devices that includes touch screens, accelerometer sensors, brain interfaces and a lot of other innovative controllers and input devices. Providing a personalized user interface and interactions has been shown to improve the user's performance [Benford et al., 1997]. The idea of personalization is more evident in CVEs. In a CVE, a number of users collaborate.
- Chapter Fulltext**: A callout pointing to the chapter fulltext section.

Figure 15.6: Web demo

### 15.4.3 REFERENCE SECTION EXTRACTION

Automatic reference section extraction is a module of the ETD-db system for exposing references to the public. Figure 15.7 illustrates its system architecture. The automatic reference section extraction consists of the following: pdf2txt, feature extraction, learning (training), and reference section extraction (see Figure 15.8).

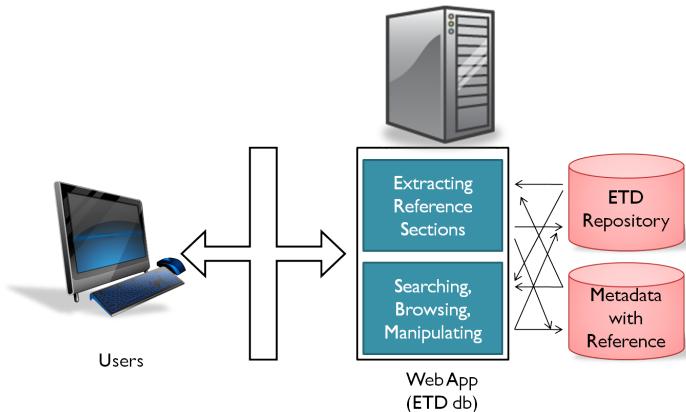


Figure 15.7: System architecture

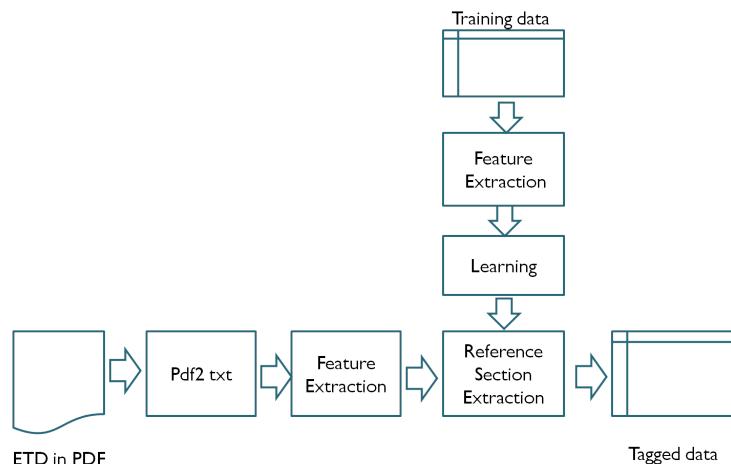


Figure 15.8: Dataflow diagram of reference section extraction

#### PDF2Txt

The Apache PDFBox API is an open source, Java-based library that supplies components for working on and manipulating PDF files. In the context of our project, the operations required dealing with stripping content from a PDF document and writing it to a text

## 15.4. CASE STUDIES 409

file. The version used in this project was 1.4.0. Figures 15.9 and 15.10 show an example of chapter reference and end reference, respectively.

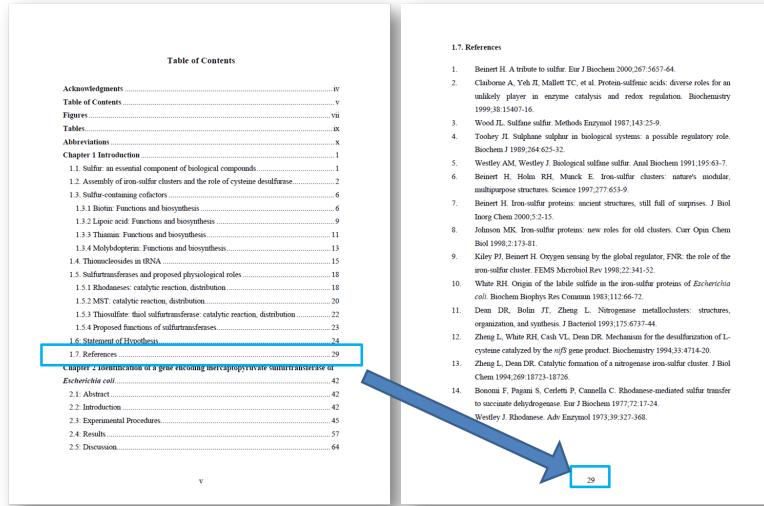


Figure 15.9: An example of a chapter reference

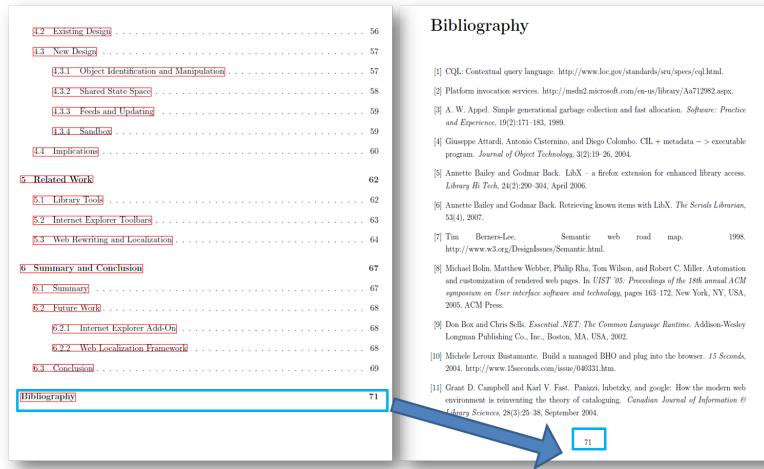


Figure 15.10: An example of an end reference

### Feature Extraction

Features that we use can be categorized into three types: Word local features, line features, and contextual features. Table 15.6 describes each feature with simple examples. 28 different

## 410 15. TEXT EXTRACTION

Table 15.6: Feature sets

Feature Name	Descriptions	Examples
Word local features	28 different string patterns	Types of punctuation, capitalization, etc.
Line feature	Patterns in a line	Number of word in the line, percentage of capitalized words
Contextual feature	Patterns of a neighborhood	Class (REF or NON-REF) of neighbor lines before and after the current line

string patterns (e.g., types of punctuation, capitalization, etc.) are used as word features. The token vector is given a bit string for each pattern it does or does not have (e.g., 0,1,0,0,1,0,0,...,0,0,1,0). Each feature vector is based on a line of text with a token vector for the tokens it has and a null vector for all of the tokens it lacks. For training data, each line is labeled as ‘REF’ if it is a reference, or as ‘NON-REF’ otherwise.

### Training

After extracting features, a machine learning based classifier is trained to identify reference sections from texts. It is important to create training sets for WEKA (Waikato Environment for Knowledge Analysis). First, some lines from a reference section and from the body of the text are extracted to a separate file. Figure 15.11 illustrates the training text lines corresponding to a reference section and the body of the text, respectively. For training, features also should be extracted. The feature vectors are then created to measure the similarity between each line in the vector space. In particular, there exists a class tag (i.e., ‘REF’ or ‘NON-REF’) indicating if that line is a reference or not, as an attribute of each vector.

### Reference section extraction (Classification)

In this way, the program can “learn” over time various patterns that references follow. Once a classifier object is trained, any feature vector can be tested against previous classifications to see if it “matches” any previous patterns. To implement this, WEKA, an open source machine learning toolkit, is used to deal specifically with data mining operations. The operations provided by WEKA are crucial to the machine-learning techniques employed by our software. The version used in this project was 3.7.3. Figure 15.12 illustrates VT

The diagram illustrates a machine learning training process. On the left, a green 'PDF' icon points to a document page. The page contains text about a user interface design, a copyright notice, and a bibliography section. The bibliography section is highlighted with a red border and lists five references. To the right of the document is a table titled 'Line\_id', 'Label', and 'Text\_line'. The table rows correspond to the lines of text in the document, with the first few lines showing the abstract and the last few lines showing the bibliography. The references from the bibliography section are also listed in the 'Text\_line' column, each preceded by a red number [1] through [5].

Line_id	Label	Text_line
2168	1.	We also hope that the work we have done on interacting with the user proves to be a significant contribution to the understanding of how the Web facilitates user interface design. By making use of client-side technology to integrate communication between user and program into the standard workflow, we hope the result is a feature set which feels natural and well-integrated to the user while being unobtrusive when its services are not needed.
2169	1.	Ultimately, these developments will put more power and control into the hands of the users, where it belongs. By providing users with easy access to resources, an intuitive interface, and the power to customize their web environment and share those customizations with others, we try to make the
2170	1.	full range of resources available from university and local libraries more accessible to the patrons of those libraries. In doing so, we improved the experience of using library resources and provided a way through which other services and resources can be exposed naturally to the user.
2171	1.	69
2172	1.	1. We also hope that the work we have done on interacting with the user proves to be a significant contribution to the understanding of how the Web facilitates user interface design. By making use of client-side technology to integrate communication between user and program into the standard workflow, we hope the result is a feature set which feels natural and well-integrated to the user while being unobtrusive when its services are not needed.
2173	1.	Ultimately, these developments will put more power and control into the hands of the users, where it belongs. By providing users with easy access to resources, an intuitive interface, and the power to customize their web environment and share those customizations with others, we try to make the
2174	1.	full range of resources available from university and local libraries more accessible to the patrons of those libraries. In doing so, we improved the experience of using library resources and provided a way through which other services and resources can be exposed naturally to the user.
2175	1.	1. 69
2176	1.	1. it belongs. By providing users with easy access to resources, an intuitive interface, and the power to customize their web environment and share those customizations with others, we try to make the
2177	1.	1. Bibliography
2178	2 [1]	CQL: Contextual query language. <a href="http://www.loc.gov/standards/sru/specs/cql.html">http://www.loc.gov/standards/sru/specs/cql.html</a> .
2179	2 [2]	Platform invocation services. <a href="http://msdn2.microsoft.com/en-us/library/Aa712982.aspx">http://msdn2.microsoft.com/en-us/library/Aa712982.aspx</a> .
2180	2 [3]	A. W. Appel. Simple generational garbage collection and fast allocation. <i>Software: Practice and Experience</i> , 19(2):171–183, 1989.
2181	2 [4]	Giuseppe Attardi, Antonio Cisternino, and Diego Colombo. CIL + metadata > executable program. <i>Journal of Object Technology</i> , 3(2):19–26, 2004.
2182	2 [5]	Annette Bailey and Godmar Back. LibX – a firefox extension for enhanced library access. <i>Library Hi Tech</i> , 24(2):290–304, April 2006.
2183		
2184		
2185		
2186		
2187		
2188		
2189		

Figure 15.11: An example of a training data set

ETD-db with Reference Metadata. ‘References’ after an ‘Abstract’ are included to show the users the references to which the ETD refers.

#### 15.4.4 EVALUATION

We evaluated our machine learning approach to reference section extraction. We used six documents randomly selected from the VT ETD-db system, and marked their reference sections and non-reference sections, manually. Table 15.7 shows the statistics of documents used in this evaluation. In the first column of the table, *# of lines* indicates the number of total lines in the text, *# of reference lines* indicates the number of reference lines in the text, *Percentage of reference lines* indicates the ratio of reference lines to the total lines in the text, and *# of features* indicates the number of features used in the training. Incidentally, all reference sections were found at the end of documents.

We evaluated two tokenization methods: Support Vector Machines (i.e., with a normal tokenizer and a simple tokenizer) and one existing method, ParsCit. Our ‘normal tokenizer’ considers delimiters (space, tab, carriage return, period (.), comma (,), semicolon (;), colon (:), single quotation mark ('), double quotation mark ("), parentheses, and question mark (?)). Our ‘simple tokenizer’ drops period, comma, semicolon, colon, double quotation mark, and parentheses, as compared with the normal tokenizer. ParsCit is based on heuristics using regular expressions. Table 15.8 shows precision, recall, and F1-score of these three

## 412 15. TEXT EXTRACTION

**Title page for ETD etd-02092005-171659**

Type of Document	thesis
Author	Aamir Anwar
URN	etd-02092005-171659
Title	Low Frequency Finite Element Modeling of Passive Noise Attenuation in Ear Defenders
Degree	MS
Department	Mechanical Engineering
Abstract	Noise levels in areas adjacent to high performance jets have increased monotonically in the past few years. When personnel are exposed to such high noise fields, the need for better hearing protection is inevitable. Adequate hearing protection may be achieved through the use of circumaural ear defenders, earplugs or both. This thesis focuses on identifying the dominant physical phenomena, responsible for the low frequency (0 to 300 Hz) acoustic response inside the earmuffs. A large volume earcup is used with the undercut seal for the study. The significance of this research is the use of finite element methods in the area of hearing protection design. The objectives of this research are to identify the dominant physical phenomena responsible for the loss of hearing protection in the lower frequency range, and develop FE models to analyze the effects of structural and acoustic modes on the acoustic pressure response inside the earcup. It is found that there are two phenomena, which are primarily responsible for the lower frequency acoustic response inside the earmuffs. These modes are recognized in this thesis as the piston mode and the Helmholtz mode. The piston mode occurs due to the dynamics of the earcup and seal at 150 Hz, which results in loss of hearing protection. The Helmholtz mode occurs due to the presence of holes. The resonant frequency of the Helmholtz mode and the pressure response depends on the leak size.
References	<ul style="list-style-type: none"> <li>[1] L.E. Kinsler, A.R. Frey, A.B. Coppens, J.V. Sanders, Fundamentals of Acoustics, 4 th ed., John Wiley &amp; Sons Inc. New York, 2000.</li> <li>[2] F. Fahy, Sound and Structural Vibration, Academic Press, 1985.</li> <li>[3] "ABAQUS Theory Manual" ver 6.4, ABAQUS, Inc. 2003</li> <li>[4] Von Gierke, H.E., and Warren, D.R., "Protection of Ear From Noise: Limiting Factors," Benox Report, 1953 contract N6 ord-020 Task Order 44, University of Chicago.</li> </ul>

Figure 15.12: VT ETD-db with reference metadata

Table 15.7: Data, used in evaluation, randomly sampled

Items	Doc1	Doc2	Doc3	Doc4	Doc5	Doc6
# of lines	4,818	4,899	2,237	6,178	2,369	2,254
# of reference lines (location)	324 (end)	291 (end)	63 (end)	214 (end)	145 (end)	73 (end)
Percentage of reference lines	6.7%	5.9%	2.8%	3.5%	6.1%	3.2%
# of features	5,185	5,493	3,208	6,061	3,393	4,097

tokenizer methods of interest, run against six test datasets. For more details in metrics, see definitions in 8.2.1

When we informally did experiments on chapter reference section extraction through picking some documents with chapter references, ParsCit failed saying Citation text cannot be found: ignoring. The ETD selected used Literature Cited as the reference section header in each chapter. ParsCit does probably not include Literature Cited as a starting word of a reference section. Even though we did an experiment with chapter reference sections starting with ‘References’, ParsCit extracted only the references in the last chapter; it also had some extraction errors where it failed to find the end of the reference section. When we

**Table 15.8:** Result of reference section extraction (P=Precision, R=Recall, F1=F1 score)

Docs	Doc1			Doc2			Doc3			Doc4			Doc5			Doc6		
Metric	P	R	F1															
Normal	.99	.80	.88	.97	.75	.85	.92	.54	.68	1.0	.83	.91	.86	.65	.74	.83	.35	.49
+ SVM	1.0	.74	.85	.97	.66	.78	.92	.38	.54	.99	.67	.80	.89	.51	.65	.95	.26	.41

did an experiment considering contextual features, against document 6 (which showed the worst performance), the performance was improved, resulting in: precision, 0.973; recall, 0.986; F1, 0.979.

## 15.5 SUMMARY

In this chapter, we discussed the concepts, literature review, 5S formalism, and case studies of text extraction. Text extraction is a necessary process for extending metadata (i.e., special descriptive structures), to be more comprehensive and precise. Text extraction is one type of pattern recognition. Specifically, it involves structuring using classification technique. We also looked into other techniques related to reference metadata extraction, which is a sequence tagging problem, that is a type of text extraction. Further, we formally described text extraction from the 5S perspective. Finally, we discussed two small studies in terms of the 5S related concepts for text extraction: 1) document section segmentation and 2) reference section extraction. Thus, we demonstrated the effectiveness of the 5S perspective to text extraction.

## 15.6 EXERCISES AND PROJECTS

- When information is extracted from text automatically, imprecision may be introduced. How should that be handled?

## APPENDIX A

# Mathematical Preliminaries

Here, we briefly review the mathematical foundations necessary for the development of the following discussion. Since the goal is complete precision, all terms used in the other definitions must be carefully and unambiguously defined. Authors' definitions of terms even as basic as "function" often disagree, so (for completeness) we begin at the most fundamental level, with set notations, relations, functions, sequences, tuples, strings, graphs, and grammars [125]. Readers familiar with these concepts can skip this section or simply refer to it as needed when some of the concepts are used in other definitions.

Formally, *set* and  $\in$  ("element of") are taken as undefined terms in the axioms of set theory. We remark that a set cannot contain itself and the "set of all sets" does not exist. That  $x$  is an element of set  $S$  is denoted  $x \in S$ . There is an "empty" set ( $\emptyset$ ). The notation  $S = \{x|P(x)\}$  defines a set  $S$  of precisely those objects  $x$  for which the logical proposition  $P(x)$  is true. Standard operations between sets  $A$  and  $B$  include union:  $A \cup B = \{x|x \in A \text{ or } x \in B\}$ ; intersection:  $A \cap B = \{x|x \in A \text{ and } x \in B\}$ ; and Cartesian product:  $A \times B = \{(a, b)|a \in A \text{ and } b \in B\}$  where  $(a, b)$  is called an *ordered pair*.  $A$  is called a *subset* of  $B$ , denoted by  $A \subset B$ , if  $x \in A$  implies  $x \in B$ . The set of all subsets of set  $S$  (including  $\emptyset$ ) exists, is called the *power set* of  $S$ , and is denoted  $2^S$ .

**Appendix Definition 1** *A binary relation  $R$  on sets  $A$  and  $B$  is a subset of  $A \times B$ . We sometimes write  $(a, b) \in R$  as  $aRb$ . An  $n$ -ary relation  $R$  on sets  $A_1, A_2, \dots, A_n$  is a subset of the Cartesian product  $A_1 \times A_2 \times \dots \times A_n$ .*

**Appendix Definition 2** *Given two sets  $A$  and  $B$ , a **function**  $f$  is a binary relation on  $A \times B$  such that for each  $a \in A$  there exists  $b \in B$  such that  $(a, b) \in f$ , and if  $(a, b) \in f$  and  $(a, c) \in f$  then  $b = c$ . The set  $A$  is called the *domain* of  $f$  and the set  $B$  is called the *codomain* of  $f$ . This is shown as  $f : A \rightarrow B$ . We write  $b = f(a)$  as a common notation for  $(a, b) \in f$ . The set  $\{f(a)|a \in A\}$  is called the *range* of  $f$ .*

**Appendix Definition 3** *A **sequence** is a function  $f$  whose domain is the set of natural numbers or some initial subset  $\{1, 2, \dots, n\}$  of the natural numbers and whose codomain is any set.*

**Appendix Definition 4** *A **tuple** is a finite sequence that is often denoted by listing the range values of the function as  $\langle f(1), f(2), \dots, f(n) \rangle$ .*

**Appendix Definition 5** *A **string** is a finite sequence of characters or symbols drawn from a finite set with at least two elements, called an **alphabet**. A string is often denoted*

by concatenating range values without punctuation. Let  $\Sigma$  be an alphabet.  $\Sigma^*$  denotes the set of all strings from  $\Sigma$ , including the empty string (an empty sequence  $\epsilon$ ). A **language** is a subset of  $\Sigma^*$ .

**Appendix Definition 6** A **graph**  $G$  is a pair  $(V, E)$ , where  $V$  is a nonempty set (whose elements are called **vertices**) and  $E$  is a set of two-item sets of vertices,  $\{u, v\}$ ,  $u, v \in V$ , called **edges**. A **directed graph** (or **digraph**)  $G$  is a pair  $(V, E)$ , where  $V$  is a nonempty set of vertices (or nodes) and  $E$  is a set of edges (or arcs) where each edge is an ordered pair of distinct vertices  $(v_i, v_j)$ , with  $v_i, v_j \in V$  and  $v_i \neq v_j$ . The edge  $(v_i, v_j)$  is said to be **incident** on vertices  $v_i$  and  $v_j$ , in which case  $v_i$  is **adjacent to**  $v_j$ , and  $v_j$  is **adjacent from**  $v_i$ .

Several additional concepts are associated with graphs. A **walk** in graph  $G$  is a sequence of not-necessarily distinct vertices such that for every adjacent pair  $v_i, v_{i+1}$ ,  $1 \leq i < n$ , in the sequence,  $(v_i, v_{i+1}) \in E$ . We call  $v_1$  the origin of the walk and  $v_n$  the terminus. The **length** of the walk is the number of edges that it contains. If the edges of the walk are distinct, the walk is a **trail**. If the vertices are distinct, the walk is a **path**. A walk is **closed** if  $v_1 = v_n$  and the walk has positive length. A **cycle** is a closed walk where the origin and non-terminal vertices are distinct. A graph is **acyclic** if it has no cycles. A graph is **connected** if there is a path from any vertex to any other vertex in the graph. A **tree** is a connected, acyclic graph. A **directed tree** or (DAG) is a connected, directed graph where one vertex - called the root - is adjacent from no vertices and all other vertices are adjacent from exactly one vertex. A graph  $G' = (V', E')$  is a **subgraph** of  $G = (V, E)$ , if  $V' \subseteq V$  and  $E' \subseteq E$ .

**Appendix Definition 7** A **context-free grammar** is a quadruple  $(V, \Sigma, R, s_0)$  where  $V$  is a finite set of symbols called non-terminals,  $\Sigma$  is an alphabet of terminal symbols,  $R$  is a finite set of rules and  $s_0$  is a distinguished element of  $V$  called the **start symbol**.

A **rule**, also called a production, is an element of the set  $V \times (V \cup \Sigma)^*$ . Each production is of the form  $A \rightarrow \alpha$  where  $A$  is a non-terminal and  $\alpha$  is a string of symbols (terminals and/or non-terminals).

**Appendix Definition 8** A **deterministic finite automaton** is a 5-tuple  $(Q, q_0, A, \Sigma, \delta)$  where  $Q$  is a finite set of symbols called states,  $q_0 \in Q$  is the **start** automaton state,  $A \subseteq Q$  is a distinguished set of accepting states,  $\Sigma$  is an alphabet (defining what set of input strings the automaton operates on), and  $\delta$  is a function from  $Q \times \Sigma$  into  $Q$ , called the **transition function** of the automaton.

The finite automaton begins in state  $q_0$  and reads characters of an input string one at a time. If after reading the string the automaton is in a state  $q \in A$  the string is **accepted**.

**Appendix Definition 9** Let  $X$  be a set. A  **$\sigma$ -algebra** is a collection  $\mathbb{B}$  of subsets of  $X$  that satisfies the following conditions:

## 416 A. MATHEMATICAL PRELIMINARIES

1. every union of a countable collection of sets in  $\mathbb{B}$  is again in  $\mathbb{B}$ , i.e., if  $A_i \in \mathbb{B}$  ( $i = 1, 2, 3, \dots$ ), then  $\bigcup_{i=1}^{\infty} A_i \in \mathbb{B}$ ;
2. if  $A \in \mathbb{B}$ , then  $\tilde{A} \in \mathbb{B}$ , where  $\tilde{A}$  is the complement of  $A$  with respect to  $X$ .

One consequence of the definition of  $\sigma$ -algebra is that the intersection of a countable collection of sets in  $\mathbb{B}$  is again in  $\mathbb{B}$ .

**Appendix Definition 10** A **measurable space** is a tuple  $(X, \mathbb{B})$  consisting of a set  $X$  and a  $\sigma$ -algebra  $\mathbb{B}$  of subsets of  $X$ .

A subset  $A$  of  $X$  is called **measurable** (or **measurable with respect to  $\mathbb{B}$** ) if  $A \in \mathbb{B}$ . A **measure**  $\mu$  on measurable space  $(X, \mathbb{B})$  is a nonnegative real-valued function defined for all sets of  $\mathbb{B}$  such that the following conditions are satisfied:

1.  $\mu(\emptyset) = 0$  where  $\emptyset$  is the empty set, and
2.  $\mu(\bigcup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} \mu(A_i)$  for any sequence  $A_i$  of pairwise disjoint measurable sets.

**Appendix Definition 11** A **measure space**  $(X, \mathbb{B}, \mu)$  is a measurable space  $(X, \mathbb{B})$ , with measure  $\mu$  defined on  $\mathbb{B}$ .

**Appendix Definition 12** A **probability space** is a measure space  $(X, \mathbb{B}, \mu)$ , such that measure  $\mu(X) = 1$ .

**Appendix Definition 13** A **vector space** is a set  $V$  (whose elements are called **vectors**) together with a field of “scalars”<sup>1</sup> with an addition operation  $+ : V \times V \rightarrow V$  and a multiplication operation  $* : S \times V \rightarrow V$  such that if  $x, y, z$  are in  $V$  and  $\alpha$  and  $\beta$  are in  $S$  then:

1. there is a unique vector  $0 \in V$  such that  $x + 0 = x$  for all  $x \in V$  (additive identity);
2. for each vector  $x \in V$  there exists a vector  $-x \in V$  such that  $x + (-x) = 0$  (additive inverse);
3.  $(x + y) + z = x + (y + z)$  (associativity of  $+$ );
4.  $x + y = y + x$  (commutativity of  $+$ );
5.  $1 * x = x$  (identity);
6.  $(\alpha * \beta) * x = \alpha * (\beta * x)$  (associativity of  $*$ );
7.  $(\alpha + \beta) * x = \alpha * x + \beta * x$  (distributivity of  $*$  over  $+$ , right); and
8.  $\alpha * (x + y) = \alpha * x + \alpha * y$  (distributivity of  $*$  over  $+$ , left).

<sup>1</sup>In this context, the field of real numbers.

**Appendix Definition 14** A **topological space** is a pair  $(X, \mathcal{T})$  consisting of a set  $X$  and a family  $\mathcal{T} \subset 2^X$  of subsets of  $X$  such that:

1.  $\emptyset$  (the empty set)  $\in \mathcal{T}$  and  $X \in \mathcal{T}$ ;
2. for any collection of sets in  $\mathcal{T}$ ,  $\{A_i \in \mathcal{T} | i \in I\}$ ,  $\cup_{i \in I} A_i$  is also in  $\mathcal{T}$ , and if the index set  $I$  is finite,  $\cap_{i \in I} A_i$  is in  $\mathcal{T}$ .

$\mathcal{T}$  is said to be a topology for  $X$ , and elements of  $\mathcal{T}$  are called **open** sets. The complement of an open set is called a **closed** set.

## APPENDIX B

# Glossary

**5S** A framework for digital libraries, building upon five constructs that each start with the letter ‘S’: Societies, Scenarios, Spaces, Structures, and Streams

**5SGen** Software for generating a digital library, using a pool of components, based on a description in XML using the 5S language (5SL)

**5SGraph** A graphical user interface allowing a digital library designer to load a metamodel for a class of digital libraries, according to the 5S framework, and then to specify details of a particular digital library, so that 5SGen can generate that digital library

**5SL** An XML-based language used to specify a digital library

**5SQual** A software toolkit to assess the quality of a (federated) digital library, building upon the 5S framework

**5SSuite** The suite of tools supporting digital library work according to the 5S framework

**Access Control** The prevention of unauthorized use of a resource (i.e. this service controls who has access to a resource, under what conditions access can occur and what those accessing the resource are allowed to do)

**Third** The third etc ...

**Access Control** The prevention of unauthorized use of a resource (i.e., this service controls who has access to a resource, under what conditions access can occur and what those accessing the resource are allowed to do)

**Accuracy** It is a degree of how close a measurement of a quantity is to the true value of the quantity.

**Algorithm**

**Analysis of Content**

**Analyst (user)**

**Anchor**

**Animation**

**Annotation**

**Annotator (user)**

**API**

**Archive**

**Assignment**

**Audio**

**Authentication** The assurance that the communicating entity is who it claims to be

**Authorization** Specifying access rights to a resource

**Availability** A system property that allows the system to be accessible and usable upon demand by an authorized system entity

**Base document** A base document refers to information existing as whole documents for which subdocuments have been defined. A digital object becomes a BD upon creation of the first subdocument.

**Bias**

**Browse**

**CBIR**

**Classify**

**CLIR**

**Cluster**

**Collaboration**

**Community**

**Compound object**

**Composite object**

**Comment**

**Completeness**

**Compress**

## 420 B. GLOSSARY

**Consistency** The integrity, validity and accuracy of data between applications

**Content Management System**

**Copyright** It is a law that assures an ownership over a person's creations such as paintings, poems, novels, etc. If a piece of work is neither in the public domain nor the permission to use it is acquired from the creator (or copyright owner), using the work is violating the copyright law.

**Course**

**Courseware**

**Coverage**

**Crawl**

**Crawler**

**Curation**

**Curriculum** It is a designed courses to help students specialize in a certain field of study. In higher education, a curriculum often includes specialized courses as well as general courses within a field.

**Database**

**Data confidentiality** The protection of data from unauthorized disclosure

**Data Integrity** The assurance that the data received is exactly as sent by an authorized entity (i.e., contains no modification, insertion, deletion or replay)

**Denial of Service (DoS)** A security attack that prevents the normal use of communications facilities

**Descriptor**

**Digital object**

**Digital library**

**Digitize**

**DRM - digital rights management** Refers to the protection of content from the different logical security attacks and issues relating to intellectual property rights and authenticity

**Dimension**

**Dimensionality reduction**

**Dissemination**

**Dublin Core**

**Education**

**Educational Institution**

**Entities** It is something that has a separate existence in general. In data modeling or an ontology, an entity is a unit of data or concept such as a person, a place, or a thing. Entities can have relationships with other entities.

**Evaluation**

**Explore** By exploring we mean searching, browsing, investigating, studying, or analyzing for the purpose of discovery, e.g., pursuing truth or facts about something.

**Explorer user**

**Experiments**

**Functionality**

**Feature**

**Feature extraction**

**Feedback**

**Filter**

**Gazetteer**

**Generation**

**Geo-coding**

**Geo-parsing**

**Geographic entity**

**Geographic relationship**

**Geographic query**

**Grammar**

## 422 B. GLOSSARY

**Graph**

**Harvest**

**Hierarchy**

**Hypertext**

**Hypermedia**

**Hyperlink**

**IDF**

**IPR**

**Image**

**Index**

**Input configuration content**

**Interoperate**

**Knowledge base**

**LDA**

**Learner**

**Lecture**

**Lexicon**

**Link**

**Location**

**LSI** Latent Semantic Indexing is a method used in Information Retrieval field to analyze relationships of words and documents in a collection, and then index them accordingly. Documents that are indexed with LSI might have high similarity value even though the two documents do not share exact words. This is because LSI does not require an exact match to return useful results.

**Mapping**

**Mark**

**Metadata**

**Metadata extraction****MIT OpenCourseWare**

**Module** Within the scope of this book, modules mean educational modules, which consist of lesson plans, lectures, exercises, evaluation of student achievements, etc., to help students gain knowledge about a certain area of study.

**Modeling****Multimedia****Multi-modal search****Navigate**

**Non-repudiation** Provides protection against denial by one of the entities involved in a communication of having participated in all or part of the communication

**NSDL pathway****OAI-PMH****Ontology****OSR****Parse****PDF****Personalization****Policy****Precision**

**Presentation specification** A presentation specification is a descriptive metadata specification conforming to a presentation-based metadata format and is used to specify how the content in a digital object translates into a particular view/presentation.

**Preserve**

**Privacy** Is concerned with the collection and distribution of data and the legal issues involved

**Probability****Profile**

## **424 B. GLOSSARY**

Provenance  
Quality  
Query  
Query expansion  
Ranking  
Rating  
RDF  
RDF-A  
Recall  
Recommend  
Record  
Reducing  
Regular expression  
Relationships  
Relevance  
Repository  
Representation  
Result content  
Retrieval  
Robustness  
Scenario  
Schema  
Score  
Search  
Search, weighted boolean

**Search engine**

**Security** A collection of tools designed to protect data and thwart attackers

**Security Policies** The different regulations and conditions that govern how a system stores, manages, protects and distributes sensitive information

**Semantic Web****Semantic network****Sequence****Similarity****Similarity function****Simulation****Smoothing**

**Social network** It is a social structure consists of nodes and their links. Often nodes represent persons and organizations, and links show relationships between nodes connected by the links.

**Society****Space**

**SPAM** A security attack where an attacker (spammer) sends irrelevant or inappropriate messages to a large number of recipients on the Internet

**Span****Stem**

**Stemmer** It is a program that reduces derived words to their stem (i.e., root form). For example, a stemmer converts 'runs' and 'running' to their root form 'run'.

**Stream****Structure****Study designer (user)**

**Subdocument** A subdocument is a digital object that is part of a base document. A subdocument is referenced by an address, which indicates a range or span in the base document.

## 426 B. GLOSSARY

**Superimposed document** A superimposed document is a complex object, in which at least one of the constituent digital objects is a subdocument.

### SVD

**System and DL administrator (user)**

### Tagging

**Tag cloud** It is also called 'word cloud'. It is one of visualization methods that presents more frequent words in a given text with bigger fonts, thus emphasizing the potential importance of the frequent words. Less frequent words are visualized using smaller fonts instead.

### Taxonomy

### Teacher

### Text

**Term** It is a noun or a compound words in the context of Information Retrieval.

**Term extraction API** It is an Application Programming Interface (API) that can identify and extract terms from a given input document programatically.

**TF** A term frequency (TF) is a number of times that a term appears in a document.

### Thesaurus

### Threshold

### Token

### Tokenizer

**Tool builder (user)**

**Topic** A noun phrase that expresses what is being talked about or what a sentence is about.

**Topic spotting** It is one of Natural Language Processing techniques that identifies a potential topic (or topics) from a given input textual data.

### Transaction

### Tree

### Trust

**Tweet** It is a post (also called a status update) of a short microblog writing, which can be maximum 140 characters long.

**User Interface**

**Usability**

**Vector**

**Vector space model** VSM represents documents as well as a given query as vectors, and finds their relevancy by calculating the cosine values between the vectors. Vectors are weighted using term frequency-inverse document frequency (tf-idf) so that the words frequently appear in a document are emphasized in the vector, and the effect of words that appear corpus-wide is attenuated.

**Video**

**View-in-context** View-in-context is a type of browsing that involves referencing and viewing a subdocument in situ. The view in context service enables a subdocument to be viewed in the original context of its containing base document.

**Visualize**

**Web**

**Web services**

**Weighting scheme**

**Wild card**

**WordNet** It is a database of English words such as nouns, verbs, adjectives and adverbs, and their cognitive synonyms called synsets. It was developed by Princeton University and freely and publicly available. It is widely used for Natural Language Processing studies.

**XML**

## Bibliography

- [1] Speed comparison of popular crypto algorithms, May 2011.
- [2] S. Abbasi, F. Mokhtarian, and J. Kittler. Enhancing CSS-based Shape Retrieval for Objects with Shallow Concavities. *Image and Vision Computing*, 18(3):199–211, February 2000.
- [3] S. Abiteboul, P. Buneman, and D. Suciu. *Data on the Web: From Relations to Semistructured Data and XML*. Morgan Kaufmann, San Francisco, CA, 1999.
- [4] S. Abiteboul, D. Quass, J. McHugh, J. Widom, and J. L. Wiener. The Lorel Query Language for Semistructured Data. *Int. Journal on Digital Libraries*, 1(1):5–19, April 1997.
- [5] A.Carkacioglu and F. Yarman-vural. Sasi: A new Texture Descriptor for Content-Based Image Retrieval. *IEEE International Conference on Image Processing*, 2:137–140, 2001.
- [6] N. Adam, V. Atluri, and I. Adiwijaya. Systems integration in digital libraries. *Commun. ACM*, 43(6):64–72, 2000.
- [7] N. Adam, V. Atluri, E. Bertino, and E. Ferrari. A content-based authorization model for digital libraries. *Knowledge and Data Engineering, IEEE Transactions on*, 14(2):296 –315, mar/apr 2002.
- [8] M. Addis, M. Boniface, S. Goodall, P. Grimwood, S. Kim, P. Lewis, K. Martinez, and A. Stevenson. Sculpteur: Towards a new paradigm for multimedia museum information handling. In *Proc. of Semantic Web ISWC 2870*, pages 582–596, 2003.
- [9] M. Adriani and M. L. Paramita. Identifying location in indonesian documents for geographic information retrieval. In *Proceedings of the 4th ACM workshop on Geographical information retrieval*, pages 19–24, Lisbon, Portugal, 2007. ACM.
- [10] M. Afzal, H. Maurer, W. Balke, and N. Kulathuramaiyer. Rule based Autonomous Citation Mining With TIERL. *Journal of Digital Information Management*, 8(3), 2010.
- [11] M. Agosti, H. Albrechtsen, N. Ferro, I. Frommholz, P. Hansen, N. Orio, E. Panizzi, A. M. Pejtersen, and U. Thiel. DiLAS: a digital library annotation service, 2005.

- [12] M. Agosti, S. Berretti, G. Brettlecker, A. del Bimbo, N. Ferro, N. Fuhr, D. Keim, C.-P. Klas, T. Lidy, M. Norrie, P. Ranaldi, A. Rauber, H.-J. Schek, T. Schreck, H. Schuldt, B. Signer, and M. Springmann. DelosDLMS – the Integrated DELOS Digital Library Management System. In *Proc. DELOS Conference on Digital Libraries*, pages 71–80, Pisa, Italy, 2007.
- [13] M. Agosti and N. Ferro. Annotations on digital contents, 2005.
- [14] M. Agosti and N. Ferro. A formal model of annotations of digital content. *Transactions on Information Systems (TOIS)*, 26(1):1–55, 2008.
- [15] M. Agosti, N. Ferro, and N. Orio. Annotating Illuminated Manuscripts: an Effective Tool for Research and Education. In *JCDL '05: Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*, pages 121–130, 2005.
- [16] D. Ahlers and S. Boll. Retrieving address-based locations from the web. In *Proceeding of the 2nd international workshop on Geographic information retrieval*, pages 27–34, Napa Valley, California, USA, 2008. ACM.
- [17] S. M. Ahmed, B. Beck, C. A. Maurana, and G. Newton. Overcoming Barriers to Effective Community-based Participatory Research in US Medical Schools. *Education for Health*, 17(2):141–151, 2004.
- [18] M. Akbar, W. Fan, C. A. Shaffer, Y. Chen, and E. A. Fox. Digital library 2.0 for educational resources. In *Proc. TPDL 2011, Berlin, GE, Sept. 2011*. Springer, Sept. 2011.
- [19] AlgoViz. The AlgoViz Portal. <http://www.algoviz.org/>, September 2011.
- [20] R. Amsler. Application of citation-based automatic classification. Technical report, The University of Texas at Austin, Linguistics Research Center, December 1972.
- [21] J. Andre, R. Furuta, and V. Quint. *Structured Documents*. University Press, Cambridge, 1989.
- [22] D. C. Andrews. Audience-specific Online Community Design. *Communications of the ACM*, 45(4):64–68, 2002.
- [23] ANSI. Information retrieval (z39.50): Application service definition and protocol specification: The z39.50 maintenance agency official text for z39.50-1995. Technical report, Library of Congress, 1995. 1995.
- [24] D. W. Archer, L. M. Delcambre, F. Corubolo, L. Cassel, S. Price, U. Murthy, D. Maier, E. A. Fox, S. Murthy, J. McCall, K. Kuchibhotla, and R. Suryavanshi. Superimposed

## 430 B. GLOSSARY

- information architecture for digital libraries. In *ECDL '08: Proceedings of the 12th European conference on Research and Advanced Technology for Digital Libraries*, pages 88–99, Berlin, Heidelberg, 2008. Springer-Verlag.
- [25] N. Arica and F. T. Y. Vural. BAS: A Perceptual Shape Descriptor Based on the Beam Angle Statistics. *Pattern Recognition Letters*, 24(9-10):1627–1639, June 2003.
  - [26] W. Arms, D. Hillmann, C. Lagoze, D. Krafft, R. Marisa, J. Saylor, and C. Terrizzi. A spectrum of interoperability: The site for science prototype for the nsdl. *D-Lib Magazine*, 8(1), 2002.
  - [27] W. Y. Arms. *Digital Libraries*. MIT Press, Cambridge, MA, 2000.
  - [28] Y. A. Aslandogan and C. T. Yu. Techniques and Systems for Image and Video Retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):56–63, January/February 1999.
  - [29] G. G. Aspiazu, S. C. Bauer, and M. Spillett. Improving the Academic Performance of Hispanic Youth: A Community Education Model. *Bilingual Research Journal*, 22(2):1–20, 1998.
  - [30] W. F. Atchison, S. D. Conte, J. W. Hamblen, T. E. Hull, T. A. Keenan, W. B. Kehl, E. J. McCluskey, S. O. Navarro, W. C. Rheinboldt, E. J. Schweppe, W. Viavant, and J. David M. Young. Curriculum 68: Recommendations for academic programs in computer science: a report of the acm curriculum committee on computer science. *CACM*, 11(3):151–197, 1968.
  - [31] S. Atnafu, R. Chbeir, D. Coquil, and L. Brunie. Integrating similarity-based queries in image dbmss. In *Proceedings of the 2004 ACM symposium on applied computing*, pages 735–739, 2004.
  - [32] C. Awre. User-author centered multimedia building blocks. *Managing compound objects within Fedora*, 2007. Managing compound objects within Fedora, Knowledge Exchange Group - Enhanced E-theses Project Deliverable 9, URL last accessed on 09/20/10.
  - [33] C. Awre. Legal issues of compound ETDs. In *Knowledge Exchange Group - Research paper - Enhanced E-theses Project Deliverable 9.1*, available at <http://igitur-archive.library.uu.nl/DARLIN/2010-0526-200238/UUindex.html> - last accessed on 05/05/11, 2009.
  - [34] E. Babbie. *The Practice of Social Research*. Wadsworth Publishing Company, Belmont, California, 6th edition, 1990.

- [35] J. Bacon, K. Moody, and W. Yao. Access control and trust in the use of widely distributed services. In *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg*, Middleware '01, pages 295–310, London, UK, 2001. Springer-Verlag.
- [36] R. Baeza-Yates and G. Navarro. Xql and proximal nodes. *J. Am. Soc. Inf. Sci. Technol.*, 53(6):504–514, 2002.
- [37] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, Harlow, England, 1999.
- [38] M. Q. W. Baldonado. A user-centered interface for information exploration in a heterogeneous digital library. *J. American Society for Information Science*, 51(3):297–210, 2000.
- [39] R. G. Baraniuk, C. S. Burrus, B. M. Hendricks, G. L. Henry, A. O. Hero, D. H. Johnson, D. L. Jones, J. Kusuma, R. D. Nowak, J. E. Odegard, L. C. Potter, K. Ramchandran, R. J. Reedstrom, P. Schniter, I. W. Selesnick, D. B. Williams, and W. L. Wilson. Connexions: DSP education for a networked world. In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, volume 4, pages IV–4144 –IV–4147, may 2002.
- [40] D. Bargerion, B. A. J. Brush, and A. Gupta. A common annotation framework. Technical report, Microsoft Corporation: MSR-TR-2001-108, 2001.
- [41] C. Barrett, K. Bisset, S. Eubank, X. Feng, and M. Marathe. Episimdemics: an efficient and scalable framework for simulating the spread of infectious disease on large social networks. *ACM/IEEE Conference on Supercomputing*, 2008.
- [42] M. J. Bass and M. Branschofsky. DSpace at MIT: meeting the challenges. In *Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries*, JCDL '01, pages 468–, New York, NY, USA, 2001. ACM.
- [43] D. S. Batista, M. J. Silva, F. M. Couto, and B. Behera. Geographic signatures for semantic retrieval. In *Proceedings of the 6th Workshop on Geographic Information Retrieval*, GIR '10, pages 19:1–19:8, New York, NY, USA, 2010. ACM.
- [44] K. D. Bayley. *Typologies and Taxonomies – An Introduction to Classification Techniques*. SAGE Publications, Thousand Oaks, California, 1994.
- [45] M. Bayraktar, C. Zhang, B. Vadapalli, N. Kipp, and E. A. Fox. A web art gallery. In *Proc. Digital Libraries '98, The Third ACM Conf. on Digital Libraries*, pages 277–278. ACM, Pittsburgh, PA, 1998.

## 432 B. GLOSSARY

- [46] B. B. Bederson. Photomesa: A Zoomable Image Browser Using Quantum Treemaps and Bubblemaps. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 71–80, Orlando, FL, USA, 2001.
- [47] G. Beenken, K. Ling, X. Wang, K. Chang, D. Frankowski, P. Resnick, and R. E. Kraut. Using Social Psychology to Motivate Contributions to Online Communities. In *Proceedings of the 2004 ACM conference on Computer Supported Cooperative Work*, pages 212–221, 2004.
- [48] C. Beeri. A formal approach to object-oriented databases. *IEEE DKE*, 5:353–382, December 1990.
- [49] N. Belkin. Anomalous states of knowledge as the basis for information retrieval. *Canadian Journal of Inf. Sci.*, 5:133–143, 1980.
- [50] N. Belkin, P. Marchetti, and C. Cool. Braque: Design of an interface to support user interaction in information retrieval. *Information Processing and Management*, 29(3):325–344, 1993.
- [51] N. J. Belkin, R. N. Oddy, and H. M. Brooks. ASK for information retrieval. *Journal of Documentation*, 33(2):61–71, 1982.
- [52] bepress. bepress repository technology, 2005.
- [53] L. D. Bergman, V. Castelli, and C.-S. Li. Progressive content-based retrieval from satellite image archives. *D-Lib Magazine*, 3(10), October 1997.
- [54] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5), May 2001.
- [55] S. Berretti, A. D. Bimbo, and E. Vicario. Spatial arrangement of color in retrieval by visual similarity. *Pattern Recognition*, 35(8):1661–1674, 2002.
- [56] A. P. Bishop, N. A. V. House, and B. P. Buttenfield. *Digital library use : social practice in design and evaluation*. MIT Press, Cambridge, Mass., 2003.
- [57] K. R. Bisset, J. Chen, X. Feng, V. A. Kumar, and M. V. Marathe. Epifast: a fast algorithm for large scale realistic epidemic simulations on distributed memory systems. In *Proceedings of the 23rd international conference on Supercomputing*, ICS '09, pages 430–439, New York, NY, USA, 2009. ACM.
- [58] D. C. Blair and M. E. Maron. An evaluation of retrieval effectiveness for a full-text document-retrieval system. *Communications of the ACM*, 28(3), March 1985.

- [59] A. D. Blaser and M. J. Egenhofer. A visual tool for querying geographic databases. In *Proceedings of the working conference on Advanced visual interfaces*, AVI '00, pages 211–216, New York, NY, USA, 2000. ACM.
- [60] D. Blei, L. Carin, and D. Dunson. Probabilistic topic models. *Signal Processing Magazine, IEEE*, 27(6):55 – 65, 2010.
- [61] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.
- [62] A. Blessing, R. Kuntz, and H. Schütze. Towards a context model driven german geotagging system. In *Proceedings of the 4th ACM workshop on Geographical information retrieval*, pages 25–30, Lisbon, Portugal, 2007. ACM.
- [63] M. Bober. MPEG-7 Visual Shape Descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):716–719, June 2001.
- [64] C. Bohm, S. Berchtold, and D. A. Keim. Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. *ACM Computing Surveys (CSUR)*, 33(3):322–373, 2001.
- [65] R. Boisvert. The architecture of an intelligent virtual mathematical software repository system. *Mathematics and Computers in Simulation*, 36:269–279, 1994.
- [66] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, SIGMOD '08, pages 1247–1250, New York, NY, USA, 2008. ACM.
- [67] G. Booch. UML in action. *Communications of the ACM*, 42(10):26–28, 1999.
- [68] K. A. V. Borges. *Uso de uma ontologia de lugar urbano para reconhecimento e extração de evidências geoespaciais na Web*. Doctoral thesis, UFMG - Universidade Federal de Minas Gerais, 2006.
- [69] K. A. V. Borges, A. H. F. Laender, C. B. Medeiros, and J. Clodoveu A. Davis. Discovering geographic locations in web pages using urban addresses. In *Proceedings of the 4th ACM workshop on Geographical information retrieval*, pages 31–36, Lisbon, Portugal, 2007. ACM.
- [70] C. Borgman. UCLA-NSF Social Aspects of Digital Libraries Workshop, Feb. 1996. <http://ucla.edu>.
- [71] C. L. Borgman. Social aspects of digital libraries. In *DL'96: Proceedings of the 1st ACM International Conference on Digital Libraries*, D-Lib Working Session 2A, pages 170–171, 1996.

## 434 B. GLOSSARY

- [72] C. L. Borgman. What are digital libraries? competing visions. *Information Processing and Management*, 35(3):227–243, 1999.
- [73] C. L. Borgman. *From Gutenberg to the global information infrastructure: Access to information in the networked world*. MIT Press, Cambridge, MA, 2003.
- [74] C. L. Borgman. *Scholarship in the Digital Age: Information, Infrastructure, and the Internet*. MIT Press, Cambridge, MA, Sept. 2010.
- [75] C. L. Borgman, J. C. Wallis, M. S. Mayernik, and A. Pepe. Drowning in data: digital library architecture to support scientific use of embedded sensor networks. In *Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital libraries*, JCDL '07, pages 269–277, New York, NY, USA, 2007. ACM.
- [76] P. Borlund and P. Ingwersen. The development of a method for the evaluation of interactive information retrieval systems. *Journal of Documentation*, 53(3):225–250, June 1997.
- [77] W. Borst. *Construction of Engineering Ontologies*. PhD thesis, University of Tweenty, Enschede, The Netherlands, 1997.
- [78] C. M. Bowman, P. B. Danzig, D. R. Hardy, U. Manber, and M. F. Schwartz. The harvest information discovery and access system. *Computer Networks and ISDN Systems*, 28(1):119–126, 1995.
- [79] C. M. Bowman, P. B. Danzig, D. R. Hardy, U. Manber, M. F. Schwartz, and D. P. Wessels. Harvest: A scalable, customizable discovery and access system. Technical Report CU-CS-732-94, Department of Computer Science, University of Colorado, Boulder, August 1994.
- [80] D. M. Boyd and N. B. Ellison. Social Network Sites: Definition, History, and Scholarship. *Journal of Computer-Mediated Commu.*, 13(1):210–230, 2008.
- [81] J. Brank, M. Grobelnik, and D. Mladenic. A survey of ontology evaluation techniques. In *Proceedings of the Conference on Data Mining and Data Warehouses (SiKDD 2005)*, pages 166–170. Citeseer, 2005.
- [82] D. F. Brauner, M. A. Casanova, and R. L. Milidiü. Towards gazetteer integration through an instance-based thesauri mapping approach. In *Advances in Geoinformatics, Part 4*, pages 235–245, Campos do Jordão, SP, Brazil, 2007. Springer. S6 - Distributed GIS / GIS and the Internet.
- [83] P. Brusilovsky, J. Eklund, and E. Schwarz. Web-based education for all: a tool for development adaptive courseware. *Computer Networks and ISDN Systems*, 30(1-7):291 – 300, 1998. Proceedings of the Seventh International World Wide Web Conference.

- [84] G. Buchanan, J. Gow, A. Blandford, J. Rimmer, and C. Warwick. Representing aggregate works in the digital library. In *JCDL '07: Proceedings of the 2007 conference on Digital libraries*, pages 247–256, New York, NY, USA, 2007. ACM Press.
- [85] I. S. Burnett, F. Pereira, R. V. de Walle, , and R. Koenen. *The MPEG-21 Book*. John Wiley & Sons, 2006.
- [86] D. Buscaldi and P. Rosso. A conceptual density-based approach for the disambiguation of toponyms. *International Journal of Geographical Information Science*, 22(3):301, 2008.
- [87] V. Bush. As we may think. *Atlantic Monthly*, 1945.
- [88] A. J. C. Smythe. Reusable Asset Specification (RAS) - version 2.2, 2002.
- [89] A. J. C. Smythe. IMS Content Packaging Information Model, specification, IMS Global Learning Consortium, Inc., Oct, 2009.
- [90] R. Z. Cabada, M. L. B. Estrada, and C. A. R. Garcia. Educa: A web 2.0 authoring tool for developing adaptive and intelligent tutoring systems using a Kohonen network. *Expert Systems with Applications*, 38(8):9522 – 9529, 2011.
- [91] L. Cai and T. Hofmann. Hierarchical document categorization with support vector machines. In *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 78–87, New York, NY, USA, 2004. ACM.
- [92] G. Câmara, M. A. Casanova, A. S. Hemerly, G. C. Magalhães, and C. M. B. Medeiros. Anatomia de sistemas de informação geográfica. In *10a. Escola de Computação*, page 197, Campinas, 1996. Instituto de Computação - UNICAMP.
- [93] C. E. C. Campelo and C. d. S. Baptista. Geographic scope modeling for web documents. In *Proceeding of the 2nd international workshop on Geographic information retrieval*, pages 11–18, Napa Valley, California, USA, 2008. ACM.
- [94] F. Can, E. A. Fox, C. Snavely, and R. K. France. Incremental clustering for very large document databases: Initial marian experience. *Information Systems*, 84:101–114, 1995.
- [95] P. S. Canada. Canadian Disaster Database. <http://www.publicsafety.gc.ca/prg/em/cdd/srch-eng.aspx>, 2011. [Online; accessed 26-September-2011].
- [96] K. S. Candan, H. Liu, and R. Suvarna. Resource description framework: metadata and its applications. *ACM SIGKDD Explorations Newsletter*, 3:1, 2001.

## 436 B. GLOSSARY

- [97] L. Candela, D. Castelli, N. Ferro, Y. Ioannidis, G. Koutrika, C. Meghini, P. Pagano, S. Ross, D. Soergel, M. Agosti, M. Dobreva, V. Katifori, and H. Schuldt. The DELOS digital library reference model - foundations for digital libraries. version 0.98, 2008.
- [98] L. Candela, D. Castelli, Y. Ioannidis, G. Koutrika, P. Pagano, S. Ross, H. J. Schek, and H. Schuldt. Deliverable d1.4.2: A reference model for digital library management systems interim report, delos digital library, available at [http://www.delos.info/index.php?option=com\\_content&task=view&id=345](http://www.delos.info/index.php?option=com_content&task=view&id=345), , 2006.
- [99] L. Candela, D. Castelli, Y. Ioannidis, G. Koutrika, P. Pagano, S. Ross, H.-J. Schek, and H. Schuldt. The Digital Library Manifesto. In *DELOS, A Network of Excellence on Digital Libraries – IST-2002-2.3.1.12, Technology-enhanced Learning and Access to Cultural Heritage*. [http://146.48.87.122:8003/OLP/Repository/1.0/Disseminate/delos/2006\\_other\\_0081/content/pdf?version=1](http://146.48.87.122:8003/OLP/Repository/1.0/Disseminate/delos/2006_other_0081/content/pdf?version=1) [last visited 2007, March 23], September 2006.
- [100] N. Cardoso and M. J. Silva. Query expansion through geographical feature types. In *Proceedings of the 4th ACM workshop on Geographical information retrieval*, pages 55–60, Lisbon, Portugal, 2007.
- [101] M. Carey, L. Haas, V. Maganty, and J. Williams. Pesto: An integrated query/browser for object databases. In *Proceedings of VLDB*, pages 203–214, 1996.
- [102] F. Carmagnola, F. Cena, and C. Gena. User model interoperability: a survey. *User Modeling and User-Adapted Interaction*, 21:285–331, 2011. 10.1007/s11257-011-9097-5.
- [103] J. M. Carroll, editor. *Minimalism Beyond the Nurnberg Funnel*. MIT Press, Cambridge, MA, 1988.
- [104] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Image Segmentation Using Expectation-Maximization and its Application to Image Querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1026–1038, August 2002.
- [105] D. Castelli, Y. Ioannidis, S. Ross, H. J. Schek, and H. Schuldt. Delos network of excellence on digital libraries - reference model for dlms, 2006.
- [106] D. Castelli and P. Pagano. A Flexible Repository Service: the OpenDLib Solution. In *Proc. ELPUB*, pages 194–202, 2002.
- [107] D. Castelli and P. Pagano. Opendlib: A digital library service system. In *Research and Advanced Technology for Digital Libraries, Proceedings of the 6th European Conference, ECDL 2002, Rome, Italy, September 2002*, pages 292–308. ECDL, 2002.

- [108] CC2001. Computing curricula 2001: Computer science (ieee computer society and association for computing machinery joint task force on computing curricula). *Journal on Educational Resources in Computing (JERIC)*, 1(3es), 2001.
- [109] CCSDS. Reference model for an open archival information system (oais) : Recommendation for space data system standards : Cesds 650.0-b-1. Technical report, Consultative Committee for Space Data Systems, January 2002.
- [110] M. Ceci and D. Malerba. Classifying web documents in a hierarchy of categories: a comprehensive study. *Journal of Intelligent Information Systems*, 28(1):37–78, February 2007.
- [111] J. Chang and D. M. Blei. Relational topic models for document networks. *Artificial Intelligence*, 9:81–88, 2009.
- [112] S. Chaudhuri and L. Gravano. Optimizing Queries over Multimedia Repositories. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, pages 91–102, Montreal, Quebec, 1996.
- [113] D. N. Chen and Y. C. Chiang. A Document Recommendation System Based on Collaborative Filtering and Personal Ontology. In *11th International Conference on Informatics and Semiotics in Organisations*, pages 255–262, April 2009.
- [114] Y.-Y. Chen, T. Suel, and A. Markowetz. Efficient query processing in geographic web search engines. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 277–288, Chicago, IL, USA, 2006.
- [115] K. Cheung, A. Lashtabeg, and D. J. SCOPE: A Scientific Compound Object Publishing and Editing System. *The International Journal of Digital Curation*, 2(2):4–18, 2008.
- [116] N. Ching, V. Jones, and M. Winslett. Authorization in the digital library: secure access to services across enterprise boundaries. In *Research and Technology Advances in Digital Libraries, 1996. ADL '96., Proceedings of the Third Forum on*, pages 110 –119, may 1996.
- [117] H. Cho, S. Sra, I. Dhillon, and Y. Guan. Minimum Sum-Squared Residue Co-Clustering of Gene Expression Data. In *Proceedings of the Fourth SIAM International Conference on Data Mining*, 2004.
- [118] G. Chowdhury and S. Chowdhury. *Introduction to Digital Libraries*. Facet Publishing, 2003.
- [119] P. Ciaccia, M. Patella, and P. Zezula. M-tree: An Efficient Access Method for Similarity Search in Metric Spaces. In *Proceedings of 23rd International Conference on Very Large Data Bases*, pages 426–435, Athens, Greece, 1997.

## 438 B. GLOSSARY

- [120] CITIDEL. Computing and Information Technology Interactive Digital Educational Library, [www.citidel.org](http://www.citidel.org), 2004.
- [121] C. L. Clarke, G. V. Cormack, and F. J. Burkowski. An algebra for structured text search and a framework for its implementation. *The Computer Journal*, 38:43–56, 1995.
- [122] A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Physical Review E*, 70(6):1–6, 2004.
- [123] E. Clementini, P. D. Felice, and P. van Oosterom. A small set of formal topological relationships suitable for end-user interaction. In *Proceedings of the Third International Symposium on Advances in Spatial Databases*, pages 277–295. Springer-Verlag, 1993.
- [124] J. H. Coombs, A. H. Renear, and S. J. DeRose. Markup systems and the future of scholarly text processing. *Communications of the ACM*, 30(11):933–947, 1988.
- [125] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press, 1990.
- [126] E. Cortez, A. da Silva, M. Gonçalves, F. Mesquita, and E. de Moura. FLUX-CIM: flexible unsupervised extraction of citation metadata. In *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, page 224. ACM, 2007.
- [127] F. Corubolo, P. B. Watry, and J. Harrison. Location and format independent distributed annotations for collaborative research. In *Proc. ECDL 2007, Budapest*. ECDL, 2007.
- [128] E. Cosijn and P. Ingwersen. Dimensions of relevance. *Inf. Process. Manage.*, 36(4):533–550, 2000.
- [129] L. Costa and R. C. Jr. *Shape Analysis and Classification: Theory and Practice*. CRC Press, Boca Raton, FL, USA, 2001.
- [130] I. Councill, C. Giles, and M. Kan. ParsCit: An open-source CRF reference string parsing package. In *Proceedings of LREC*, volume 2008. Citeseer, 2008.
- [131] T. Couto, M. Cristo, M. A. Gonçalves, P. Calado, N. Ziviani, E. Moura, and B. Ribeiro-Neto. A comparative study of citations and links in document classification. In *Proceedings of the 6th ACM/IEEE-CS joint conference on digital libraries*, pages 75–84, New York, NY, USA, 2006. ACM Press.
- [132] A. Crabtree, M. B. Twidale, J. O’Brien, and D. M. Nichols. Talking in the library: implications for the design of digital libraries. In *Proc. of the 2nd ACM Int. Conf. on Digital Libraries*, pages 221–229, New York, July 1997.
- [133] A. Crespo and H. Garcia-Molina. Archival storage for digital libraries. In *DL’98: Proceedings of the 3rd ACM International Conference on Digital Libraries*, pages 69–78, 1998.

- [134] F. Crestani, M. Lalmas, C. J. v. Rijsbergen, and I. Campbell. “Is this document relevant? probably”: A survey of probabilistic models in information retrieval. *ACM Computing Surveys*, 30(4):528–552, 1998.
- [135] D. Crockford. Json: The fat-free alternative to XML. Presented at XML 2006, Boston, MA, December 2006.
- [136] W. Croft and R. Thompson. A New Approach to the Design of Document Retrieval Systems. *JASIS*, 38(6):389–404, 1987.
- [137] F. Curbera and al. Unraveling the Web services web: An introduction to SOAP, WSDL, and UDDI. *IEEE Distributed Systems Online*, 3(4), 2002.
- [138] R. da S. Torres and A. X. Falcão. Contour Salience Descriptors for Effective Image Retrieval and Analysis. *Image and Vision Computing*, 2006. To appear.
- [139] R. da S. Torres, A. X. Falcão, and L. da F. Costa. A Graph-based Approach for Multiscale Shape Analysis. *Pattern Recognition*, 37(6):1163–1174, June 2004.
- [140] F. A. Das Neves and E. A. Fox. A study of user behavior in an immersive virtual environment for digital libraries. In *Proceedings of the Fifth ACM Conference on Digital Libraries: DL '00, June 2-7, 2000, San Antonio, TX*, pages 103–111. ACM Press, New York, 2000.
- [141] D. Davies and D. Bouldin. A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2):224–227, 1979.
- [142] D. Davis. Fedora repository 3 documentation, available at <http://www.fedora-commons.org/confluence/display/fcr30>, 2009.
- [143] M. Davis, J. Spohrer, and P. Maglio. Guest editorial: How technology is changing the design and delivery of services. *Operations Management Research*, 4:1–5, 2011. 10.1007/s12063-011-0046-6.
- [144] M. D. Davis, R. Sigal, and E. J. Weyuker. *Computation, Complexity, and Languages (second edition)*. Academic Press, 1994.
- [145] M. Day, R. Tsai, C. Sung, C. Hsieh, C. Lee, S. Wu, K. Wu, C. Ong, and W. Hsu. Reference metadata extraction using a hierarchical knowledge representation framework. *Decision Support Systems*, 43(1):152–167, 2007.
- [146] R. O. de Alencar, C. A. Davis,Jr., and M. A. Gonçalves. Geographical classification of documents using evidence from wikipedia. In *Proceedings of the 6th Workshop on Geographic Information Retrieval, GIR '10*, pages 12:1–12:8, New York, NY, USA, 2010. ACM.
- [147] E. S. de Moura, G. Navarro, N. Ziviani, and R. A. Baeza-Yates. Fast and flexible word searching on compressed text. *ACM Trans. Inf. Syst.*, 18(2):113–139, 2000.

## 440 B. GLOSSARY

- [148] H. V. de Sompel and C. Lagoze. The Santa Fe Convention of the Open Archives Initiative. *D-Lib Magazine*, 6(2), Feb. 15, 2000.
- [149] C. S. de Souza and J. Preece. A Framework for Analyzing and Understanding Online Communities. *Interacting with Computers*, 16(3):579–610, 2004.
- [150] E. Deelman, J. Blythe, Y. Gil, C. Kesselman, S. Koranda, A. Lazzarini, G. Mehta, M. A. Papa, and K. Vahi. Pegasus and the pulsar search: From metadata to execution on the grid. In *Applications Grid Workshop at the Fifth International Conference on Parallel Processing and Applied Mathematics*, pages 821–830, 2003.
- [151] A. del Bimbo. *Visual Information Retrieval*. Morgan Kaufmann Publishers, San Francisco, CA, USA, 1999.
- [152] L. Delcambre, D. Maier, S. Bowers, M. Weaver, L. Deng, P. Gorman, J. Ash, M. Lavelle, and J. A. Lyman. Bundles in captivity: An application of superimposed information. In *Proceedings of the 17th International Conference on Data Engineering (ICDE), Heidelberg, Germany*, pages 111–120. ICDE, 2001.
- [153] L. Delcambre, D. Maier, and R. Reddy. Structured maps: Modeling explicit semantics over a universe of information. *International Journal of Digital Libraries*, 1(1):20–35, 1997.
- [154] L. M. Delcambre, D. Archer, S. Price, U. Murthy, E. A. Fox, and L. Cassel. Superimposing a strand map over a database lecture, 2008. Presented at the 39th SIGCSE technical symposium on Computer science education.
- [155] L. M. L. Delcambre and D. Maier. Models for superimposed information. In *ER '99: Proceedings of the Workshops on Evolution and Change in Data Management, Reverse Engineering in Information Systems, and the World Wide Web and Conceptual Modeling*, pages 264–280, London, UK, 1999. Springer-Verlag.
- [156] DELOS. Sixth delos workshop: Preservation of digital information, june 17-19, 1998.
- [157] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '01, pages 269–274, 2001.
- [158] Y. Ding, G. Chowdhury, and S. Foo. Template mining for the extraction of citation from digital documents. In *Proceedings of the Second Asian Digital Library Conference, Taiwan*, pages 47–62. Citeseer, 1999.
- [159] DLIB. D-lib magazine: The magazine of digital library research, doi:10.1045/dlib.magazine — issn:1082-9873, 1995.

- [160] P. Dourado, P. Ferreira, and A. Santanchè. Representação unificada de objetos digitais complexos: Confrontando o ras com o ims cp. In *III Workshop de Bibliotecas Digitais.*, 2006.
- [161] Drupal. Drupal.
- [162] DSpace. DSpace, 2003.
- [163] S. A. Dudani, K. J. Breeding, and R. B. McGhee. Aircraft Identification by Moment Invariants. *IEEE Transactions on Computers*, 26(1):39–45, January 1977.
- [164] S. Dumais and H. Chen. Hierarchical classification of web content. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 256–263, New York, NY, USA, 2000. ACM.
- [165] R. Dunlap, L. Mark, S. Rugaber, V. Balaji, J. Chastang, L. Cinquini, C. Deluca, D. Middleton, and S. Murphy. Earth system curator: metadata infrastructure for climate modeling. *Earth Science Informatics*, 1(3):131–149, Nov. 2008.
- [166] J. Dunn. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Journal of Cybernetics*, 3(3):32–57, 1974.
- [167] L. Duranti. The long-term preservation of accurate and authentic digital data: theINTER-PARES project. *Data Science Journal*, 4:106–118, 2005.
- [168] DuraSpace. DuraSpace.org, 2011.
- [169] N. Dushay, J. C. French, and C. Lagoze. Using query mediators for distributed searching in federated digital libraries. In *Proceedings of the Fourth ACM Conference on Digital Libraries (DL '99, August 11-14, 1999)*, pages 171–178. ACM, Berkeley, CA, 1999. Aug.
- [170] P. Eades and M. L. Huang. Navigating Clustered Graphs using Force-Directed Methods. *Journal of Graph Algorithms and Applications*, 4:157–181, 2000.
- [171] D. Egan, J. Remde, T. Landauer, C. Lochbaum, and L. Gomez. Behavioral evaluation and analysis of a hypertext browser. In *Proceedings of CHI*, pages 205–210, 1989.
- [172] D. E. Egan, J. R. Remde, L. M. Gomez, T. K. Landauer, J. Eberhardt, and C. C. Lochbaum. Formative design evaluation of superbook. *ACM Trans. Inf. Syst.*, 7(7):30–57, 1989. Jan.
- [173] M. J. Egenhofer. Query processing in spatial-query-by-sketch. *Journal of Visual Languages & Computing*, 8:403–424, Aug. 1997.
- [174] D. Ellis. The physical and cognitive paradigms in information retrieval research. *Journal of Documentation*, 48:45–64, 1992.

## 442 B. GLOSSARY

- [175] D. Embley, D. Campbell, Y. Jiang, S. Liddle, D. Lonsdale, Y. Ng, and R. Smith. Conceptual-model-based data extraction from multiple-record Web pages. *Data & Knowledge Engineering*, 31(3):227–251, 1999.
- [176] Ensemble. Ensemble Computing Portal. <http://www.computingportal.org/>, September 2011.
- [177] I. Esslimani, A. Brun, and A. Boyer. Densifying a Behavioral Recommender System by Social Networks Link Prediction Methods. *Social Network Analysis and Mining*, 1(3):159–172, 2011.
- [178] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165(1):91 – 134, 2005.
- [179] European Commission. Commission Recommendation of 24 August 2006 on the digitisation and online accessibility of cultural material and digital preservation. *Official Journal of the European Union, OJ L 236, 31.8.2006*, 49:28–30, August 2006.
- [180] A. X. Falcão, J. Stolfi, and R. A. Lotufo. The Image Foresting Transform: Theory, Algorithms, and Applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):19–29, Jan 2004.
- [181] W. Fan, P. Pathak, and L. Wallace. Nonlinear ranking function representations in genetic programming-based ranking discovery for personalized search. *Decision Support Systems*, 42(3):1338 – 1349, 2006.
- [182] D. L. Federation. Metadata encoding and transmission standard (mets), 2009.
- [183] D. Fernandes and A. C. Salgado. Geovisual interface - a visual query interface for geographic information systems. In *SBBD'00*, pages 7–19, 2000.
- [184] N. Ferran, E. Mor, and J. Minguillón. Towards personalization in digital libraries through ontologies. *Library management*, 26(4/5):206–217, 2005.
- [185] E. Ferrari, N. Adam, V. Atluri, E. Bertino, and U. Capuozzo. An authorization system for digital libraries. *The VLDB Journal*, 11:58–67, August 2002.
- [186] M. Fetscherin and M. Schmid. Comparing the usage of digital rights management systems in the music, film, and print industry. In *Proceedings of the 5th international conference on Electronic commerce*, ICEC '03, pages 316–325, New York, NY, USA, 2003. ACM.
- [187] S. Finkelstein, D. Ussishkin, and B. Halpern. Monograph Series of the Institute of Archaeology, 2000.

- [188] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by Image and Video Content: The QBIC System. *IEEE Computer*, 28(9):23–32, 1995.
- [189] F. Fonseca, M. Egenhofer, C. Davis, and G. Câmara. Semantic granularity in ontology-driven geographic information systems. *Annals of Mathematics and Artificial Intelligence*, 36(1-2):121–151, 2002.
- [190] C. for Research on the Epidemiology of Disasters CRED. EM-DAT: The International Disaster Database. <http://www.emdat.be/>, 2011. [Online; accessed 26-September-2011].
- [191] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75 – 174, 2010.
- [192] S. Fortunato and C. Castellano. Community Structure in Graphs. *Networks*, 814(2005):42, 2007.
- [193] D. J. Foskett. A note on the concept of relevance. *Information Storage and Retrieval*, 8(2):77–78, 1972.
- [194] D. J. Foskett. Thesaurus. In A. Kent, H. Lancour, and J. Daily, editors, *Encyclopedia of Library and Information Science - Volume 30*, pages 416–462. Marcel Dekker, New York, 1980.
- [195] E. Fox. The digital libraries initiative: Update and discussion: Guest editor's introduction to special section. *Bulletin of the American Society of Information Science*, 26(1):7–11, 1999.
- [196] E. Fox and R. France. Architecture of an expert system for composite document analysis, representation and retrieval. *International Journal of Approximate Reasoning*, 1(2):151–175, 1987.
- [197] E. Fox, R. France, E. Sahle, A. Daoud, and B. Cline. Development of a Modern OPAC: From REVTOLC to MARIAN. In *Proc. 16th Annual Intern'l ACM SIGIR Conf. on R & D in Information Retrieval, SIGIR '93, Pittsburgh, PA, June 27 - July 1*, pages 248–259, 1993.
- [198] E. Fox, R. Moore, R. Larsen, S. Myaeng, and S. Kim. Toward a global digital library: Generalizing us-korea collaboration on digital libraries. *D-Lib Magazine*, 8(10), October 2002.
- [199] E. A. Fox. Development of the coder system: A testbed for artificial intelligence methods in information retrieval. *Information Processing and Management*, 23(4):341–366, 1987.
- [200] E. A. Fox. Digital library source book. Technical Report TR-93-35, Virginia Tech Dept. of Computer Science, Blacksburg, VA, 1993.

## 444 B. GLOSSARY

- [201] E. A. Fox. Ir curriculum: Information engineering to digital libraries. In *Information Retrieval 2000 — Workplace Needs and Curricular Implications, Drexel University hosted Workshop/Symposium sponsored by the W.K. Kellogg Foundation*, Marriott Hotel, Philadelphia PA, 1996. invited presentation.
- [202] E. A. Fox. The 5S framework for digital libraries and two case studies: NDLTD and CSTC. In *Proceedings NIT'99, The 11th International Conf. on New Information Technology*. NIT, Taipei, Taiwan, August 1999.
- [203] E. A. Fox. Digital library research laboratory (dlrl home page), 2011.
- [204] E. A. Fox, R. Akscyn, R. Furuta, and J. Leggett. Guest editors' introduction to digital libraries. *Communications of the ACM*, 38(4):22–28, 1995. 88, April 1995.
- [205] E. A. Fox, Y. Chen, M. Akbar, C. A. Shaffer, S. H. Edwards, P. Brusilovsky, D. D. Garcia, L. M. Delcambre, F. Decker, D. W. Archer, R. Furuta, F. Shipman, S. Carpenter, and L. Cassel. Ensemble pdp-8: Eight principles for distributed portals. In *Proc. JCDL/ICADL 2010, June 21-25, Gold Coast, Australia*, pages 341–344. ACM, 2010.
- [206] E. A. Fox and K. Garach. CITIDEL collection building. Technical Report TR-03-14, Computer Science, Virginia Tech, 2003.
- [207] E. A. Fox, L. S. Heath, and D. Hix. Project Envision Final Report: A User-Centered Database from the Computer Science Literature. Technical report, Virginia Tech Dept. of Computer Science, Blacksburg, VA, 1995. <http://ei.cs.vt.edu/papers/ENVreport/final.html>.
- [208] E. A. Fox, R. S. Heller, A. Long, and D. Watkins. Crim: Curricular resources in interactive multimedia. In *Proceedings ACM Multimedia '99*. ACM, Orlando, 1999. Oct. 30 - Nov. 5, 1999.
- [209] E. A. Fox, D. Hix, L. Nowell, D. Brueni, W. Wake, L. Heath, and D. Rao. Users, user interfaces, and objects: Envision, a digital library. *J. American Society Information Science*, 44(8):480–491, 1993. 98, Sept. 1993.
- [210] E. A. Fox and L. Kieffer. Multimedia curricula, courses and knowledge modules. *ACM Computing Surveys*, 27(4):549–551, 1995. 90.
- [211] E. A. Fox, D. Knox, L. Cassel, J. A. N. Lee, M. Pérez-Quiñones, J. Impagliazzo, and C. L. Giles. Citidel: Computing and information technology interactive digital educational library, 2002.
- [212] E. A. Fox and G. Marchionini. Toward a worldwide digital library. *Communications of the ACM*, 41(4):22–28, 1998.

- [213] E. A. Fox, F. Neves, X. Yu, R. Shen, S. Kim, and W. Fan. Exploring the computing literature with visualization and stepping stones and pathways. *Commun. ACM*, 49(4):52–58, 2006.
- [214] E. A. Fox, S. Yang, and S. Kim. ETDs, NDLTD, and open access: a 5S perspective. *Ciencia da Informacao*, 35:75 – 90, 08 2006.
- [215] Fox, E. A. et al. Curriculum on digital libraries.
- [216] Fox, E. A. et al. Digital libraries curriculum development.
- [217] R. K. France, , M. A. Gonçalves, and E. A. Fox. MARIAN digital library system. <http://www.dlib.vt.edu/products/marian.html>, 2002.
- [218] E. Frank and G. W. Paynter. Predicting library of congress classifications from library of congress subject headings. *Journal of the American Society for Information Science and Technology*, 55(3):214–227, 2004.
- [219] M. Freeston. The alexandria digital library and the alexandria digital earth prototype. In *Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, JCDL '04, pages 410–410, New York, NY, USA, 2004. ACM.
- [220] R. Freitas and R. S. Torres. OntoSAIA: Um Ambiente Baseado em Ontologias para Recuperação e Anotação Semi-Automática de Imagens. In *I Workshop in Digital Libraries, Proc. XX Brazilian Symposium on Databases - SBBD 2005*, pages 60–79, Uberlândia, Brasil, 2005. (In Portuguese).
- [221] J. C. French, A. C. Chapin, and W. N. Martin. An application of multiple viewpoints to content-based image retrieval. In *Proceedings of the 3rd ACM/IEEE-CS joint conference on digital libraries*, pages 128–130, Washington, DC, USA, 2003. IEEE Computer Society.
- [222] J. C. French, A. L. Powell, C. L. Viles, T. Emmitt, and K. Prey. Evaluating database selection techniques: A testbed and experiment. In *Proc. 21st International Conf. on R&D in Information Retrieval (ACM SIGIR'98)*, pages 121–129. ACM, Melbourne Australia, 1998. Aug.
- [223] J. C. French and C. L. Viles. Ensuring retrieval effectiveness in distributed digital libraries. *J. Visual Communication and Image Representation*, 7(1):61–73, 1996. March.
- [224] J. Frew, M. Freeston, N. Freitas, L. Hill, G. Janée, K. Lovette, R. Nideffer, T. Smith, and Q. Zheng. The alexandria digital library architecture. *International Journal on Digital Libraries*, 2:259–268, May 2000.
- [225] T. L. Friedman. *The World Is Flat 3.0: A Brief History of the Twenty-first Century*. Picador, New York, 2007.

## 446 B. GLOSSARY

- [226] G. Fu, C. B. Jones, and A. I. Abdelmoty. Ontology-based spatial query expansion in information retrieval. In *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE*, Lecture Notes in Computer Science, pages 1466–1482. Springer Berlin / Heidelberg, 2005.
- [227] N. Fuhr, , and K. Grobjohann. XIRQL - an XML query language based on information retrieval concepts. *ACM Transactions on Information Systems*, 22(2):313 – 356, Apr. 2004.
- [228] N. Fuhr. XIRQL - An Extension of XQL for Information Retrieval. In *Proc. of the ACM SIGIR 2000 – Workshop on XML and Information Retrieval*, Athens, Greece, 2000.
- [229] N. Fuhr, P. Hansen, M. Mabe, A. Micsik, and I. Solvberg. Digital libraries: A generic classification and evaluation scheme. *Lecture Notes in Computer Science*, 2163:187, 2001.
- [230] N. Fuhr, C.-P. Klas, A. Schaefer, and P. Mutschke. Daffodil: An integrated desktop for supporting high-level search activities in federated digital libraries. In *ECDL 2002*, pages 597–612, City, 2002. Springer-Verlag.
- [231] N. Fuhr, G. Tsakonas, T. Aalberg, M. Agosti, P. Hansen, S. Kapidakis, C.-P. Klas, L. Kovas, M. Landoni, A. Micsik, C. Papatheodorou, C. Peters, and I. Solvberg. Evaluation of digital libraries. *International Journal of Digital Libraries*, 8(1):21–38, 2007.
- [232] K. Fullerton, J. Greenberg, M. McClure, E. Rasmussen, and D. Stewart. A digital library for education: the pen-dor project. *Electronic Library*, 17(2):75–82, 1999. Article 189EN English Times Cited:1 Cited References Count:19.
- [233] G. Furnas and S. Rauch. Considerations for information environments and the navique workspace. In *Proceedings of Digital Libraries*, pages 79–88, 1998.
- [234] R. Furuta. Defining and using structure in digital documents. In J. L. Schnase, J. J. Leggett, R. K. Furuta, and T. Metcalfe, editors, *Proceedings of Digital Libraries'94: The First Annual Conference on the Theory and Practice of Digital Libraries*, pages 139–145, College Station, TX, 1994.
- [235] V. Gaede and O. Gunther. Multidimensional Access Methods. *ACM Computing Surveys*, 30(2):170–231, 1998.
- [236] A. Gallagher, D. Joshi, J. Yu, and J. Luo. Geo-location inference from image content and user tags. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009*, pages 55–62. IEEE, June 2009.
- [237] E. Garfield. From 1950s documentalists to 20th century information scientists - and beyond. *Bulletin of the American Society for Information Science*, 26(2), 2000. December / January.

- [238] G. Geisler and G. Marchionini. The open video project: A research-oriented digital video repository. In *Proceedings of the Fifth ACM Conference on Digital Libraries: DL '00, June 2-7, 2000, San Antonio, TX*, pages 258–259. ACM Press, New York, 2000. test collection, multimedia retrieval, open source, metadata, contributed by Informed Project and Internet Archive <http://www.archive.org>.
- [239] M. R. Genesereth and R. E. Fikes. Knowledge Interchange Format, Version 3.0 Reference Manual. Technical Report Logic-92-1, Stanford University, Stanford, CA, USA, 1992.
- [240] A. Gerber and J. Hunter. Authoring, editing and visualizing compound objects for literary scholarship. *Journal of Digital Information (JoDI)*, 1(1), 2010.
- [241] P. Ginsparg. arxiv.org e-print archive, 2000.
- [242] A. Girgensohn and A. Lee. Making Web Sites be Places for Social Interaction. In *Proceedings of the 2002 ACM conference on Computer Supported Cooperative Work*, pages 136–145, 2002.
- [243] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.
- [244] H. M. Gladney. Access control for large collections. *ACM Trans. Inf. Syst.*, 15:154–194, April 1997.
- [245] H. M. Gladney and A. Cantu. Authorization management for digital libraries. *Communications of the ACM*, 44(5):63–65, 2001.
- [246] R. Godement. *Algebra*. Kershaw Publ. Co. Ltd, London, 1969.
- [247] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using Collaborative Filtering to Weave an Information Tapestry. *Communications of the ACM*, 35(12):61–70, 1992.
- [248] C. F. Goldfarb and P. Prescod. *The XML Handbook*. Prentice-Hall PTR, Upper Saddle River, NJ 07458, USA, 1998.
- [249] G. Golovchinsky. Queries? links? is there a difference? In *Proc. CHI'97*, pages 407–417, 1997.
- [250] M. A. Goncalves. *Streams, Structures, Spaces, Scenarios, and Societies (5S): A Formal Digital Library Framework and Its Applications*. PhD thesis, Virginia Tech, Blacksburg, VA, 2004.
- [251] M. A. Gonçalves and E. A. Fox. 5sl: a language for declarative specification and generation of digital libraries. In *JCDL '02: Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pages 263–272, New York, NY, USA, 2002. ACM Press.

## 448 B. GLOSSARY

- [252] M. A. Gonçalves, E. A. Fox, A. Krowne, P. Calado, A. H. F. Laender, A. S. da Silva, and B. Ribeiro-Neto. The Effectiveness of Automatically Structured Queries in Digital Libraries. In *Proc. of the 4th Joint Conf. on Digital Libraries (JCDL'2004)*, pages 98–107, Tucson, Arizona, June 7-11, 2004.
- [253] M. A. Gonçalves, E. A. Fox, L. T. Watson, and N. A. Kipp. Streams, structures, spaces, scenarios, societies (5S): A formal model for digital libraries. Technical Report TR-03-04, Computer Science, Virginia Tech, Blacksburg, VA, 2003.
- [254] M. A. Gonçalves, E. A. Fox, L. T. Watson, and N. A. Kipp. Streams, structures, spaces, scenarios, societies (5S): A formal model for digital libraries. *ACM Transactions on Information Systems*, 22(2):270–312, 2004.
- [255] M. A. Gonçalves, R. K. France, and E. A. Fox. MARIAN: Flexible interoperability for federated digital libraries. *Lecture Notes in Computer Science*, 2163:173–186, 2001.
- [256] M. A. Gonçalves, R. K. France, E. A. Fox, and T. E. Doszkocs. Marian: Searching and querying across heterogeneous federated digital libraries. In *Proceedings of the First DELOS Network of Excellence Workshop on Information Seeking, Searching and Querying in Digital Libraries, Dec. 11-12, 2000*. DELOS, Zurich, Switzerland, 2000.
- [257] M. A. Gonçalves, M. Luo, R. Shen, M. F. Ali, and E. A. Fox. An xml log standard and tool for digital library logging analysis. In *Research and Advanced Technology for Digital Libraries, 6th European Conference, ECDL 2002, Rome, Italy, September 16-18, 2002, Proceedings*, eds. Maristella Agosti and Constantino Thanos, LNCS 2458, Springer, pages 129–143, 2002.
- [258] M. A. Gonçalves, P. Mather, J. Wang, Y. Zhou, M. Luo, R. Richardson, R. Shen, L. Xu, and E. A. Fox. Java MARIAN: From an OPAC to a modern digital library system. In *Proc. of SPIRE'02*, pages 194–209, Lisbon, Portugal, September 11-13 2002.
- [259] M. A. Gonçalves, B. L. Moreira, E. A. Fox, and L. T. Watson. What is a good digital library? - defining a quality model for digital libraries. *Information Processing & Management*, 43(5):1416–1437, 2007.
- [260] M. A. Gonçalves, G. Panchanathan, U. Ravindranathan, A. Krowne, E. A. Fox, F. Jagodzinski, and L. Cassel. The xml log standard for digital libraries: Analysis, evolution, and deployment. In *Proc. JCDL'2003, Third ACM / IEEE-CS Joint Conference on Digital Libraries, May 27-31, Houston, TX*, pages 312–314. ACM, 2003.
- [261] M. Gordon and P. Pathak. Finding information on the world wide web: the retrieval effectiveness of search engines. *Inf. Process. Manage.*, 35(2):141–180, 1999.

- [262] D. Gorton. Practical digital library generation into dspace with the 5s framework. Master's thesis, Virginia Tech, 2007. Committee Chairman E. A. Fox.
- [263] Greenstone. Greenstone digital library software homepage, 2011.
- [264] H. Greisdorf. Relevance thresholds: a multi-stage predictive model of how users evaluate information. *Information Processing and Management*, 39(3):403–423, 2003.
- [265] S. Griffin. Digital libraries initiative, 1999.
- [266] R. Grishman and B. Sundheim. Message understanding conference-6: A brief history. In *Proceedings of COLING*, volume 96, pages 466–471, 1996.
- [267] T. R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.
- [268] V. A. Gruzman and V. I. Senichkin. Hypermedia models. *Autom. Remote Control*, 62(5):677–694, 2001.
- [269] F. Guerroudji-Meddah, H. Belbachir, and R. Laurini. A visual language for gis querying. In *Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on*, pages 518–521, aug. 2009.
- [270] Z. Guo, L. Ma, and H. Sun. A cooperative service model for digital library alliances based on grid. In *Proceedings of the 2010 International Conference on Machine Vision and Human-machine Interface, MVHI '10*, pages 483–486, Washington, DC, USA, 2010. IEEE Computer Society.
- [271] D. Gurzick and W. G. Lutters. Towards a Design Theory for Online Communities. In *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology*, pages 11:1–11:20, 2009.
- [272] R. Guttman. R-Tree: A Dynamic Index to Structure for Spatial Searching. In *SIGMOD Conf. Ann. Meeting*, pages 47–57, Boston, 1984.
- [273] J. Hadidjojo and S. A. Cheong. Equal graph partitioning on estimated infection network as an effective epidemic mitigation measure. *PLoS ONE*, 6(7):e22124, 2011.
- [274] F. Halasz and M. Schwartz. The Dexter Hypertext Reference Model. *Communications of the ACM*, 37(2):30, 1994.
- [275] H. Han, H. Zha, and L. Giles. A model-based k-means algorithm for name disambiguation. In *Second International Semantic Web Conference (ISWC-03) Workshop on Semantic Web Technologies for Searching and Retrieving Scientific Data.*, Workshop on Semantic Web Technologies for Searching and Retrieving Scientific Data, 2003.

## 450 B. GLOSSARY

- [276] J. V. Hansen. Audit considerations in distributed processing systems. *Communications of the ACM*, 26(8):562–569, 1983.
- [277] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural Features for Image Classification. *IEEE Transactions on Systems, Man and Cybernetics*, 3(6):610–621, 1973.
- [278] L. Hardman, D. C. A. Bulterman, and G. van Rossum. The Amsterdam hypermedia model: adding time and context to the Dexter model. *Commun. ACM*, 37(2):50–62, 1994.
- [279] S. Harnad. Integrating and navigating eprint archives through citation-linking, 1999. NSF / JISC - eLib Collaborative Project: International Digital Libraries Research Programme.
- [280] S. Harnad. CogPrints archive, 2000.
- [281] S. Harnad. The self-archiving initiative - freeing the refereed research literature online. *Nature*, 411(6837):522, 2001. May 31.
- [282] S. Harum. Digital Library Initiative (DLI). <http://dli.grainger.uiuc.edu/national.htm>.
- [283] B. Haslhofer, R. Simon, R. Sanderson, and H. van de Sompel. The open annotation collaboration (oac) model. Technical Report arXiv:1106.5178v1 [cs.DL], arXiv CoRR, June 2011.
- [284] W. Hasselbring. Information System Integration: Introduction. *Commun. ACM*, 43(6):32–38, 2000.
- [285] J. Hays and A. A. Efros. im2gps: estimating geographic information from a single image. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008.
- [286] D. He. A study of self-organizing map in interactive relevance feedback. In *ITNG '06: Proceedings of the Third International Conference on Information Technology: New Generations*, pages 394–401, Washington, DC, USA, 2006. IEEE Computer Society.
- [287] M. Hedstrom. It's about time: Research challenges in digital archiving and long-term preservation: Final report. Technical report, NSF and Library of Congress, August 2003.
- [288] B. M. Hemminger. Etd dspace implementers group, 2005.
- [289] B. Hendrickson and R. Leland. An improved spectral graph partitioning algorithm for mapping parallel computations. *SIAM J. Sci. Comput.*, 16(2):452–469, 1995.
- [290] A. Henrich and V. Luedcke. Characteristics of geographic information needs. In *Proceedings of the 4th ACM workshop on Geographical information retrieval*, pages 1–6, Lisbon, Portugal, 2007.

- [291] E. Hetzner. A simple method for citation metadata extraction using hidden markov models. In *Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*, pages 280–284. ACM, 2008.
- [292] T. Hey, S. Tansley, and K. Tolle, editors. *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Microsoft Research, 2009.
- [293] C. Hong, J. Gozali, and M. Kan. FireCite: Lightweight real-time reference string extraction from webpages. In *Proceedings of the 2009 Workshop on Text and Citation Analysis for Scholarly Digital Libraries*, pages 71–79. Association for Computational Linguistics, 2009.
- [294] J. S. Hong, H. Chen, and J. Hsiang. A Digital Museum of Taiwanese Butterflies. In *Proceedings of the Fifth ACM Conference on digital Libraries*, pages 260–261, San Antonio, Texas, United States, 2000.
- [295] P. Hsia, J. Samuel, J. Gao, D. Kung, Y. Toyoshima, and C. Chen. Formal approach to scenario analysis. *IEEE Software*, 11(2):33–41, Mar. 1994.
- [296] M. K. Hu. Visual Pattern Recognition by Moment Invariants. *IRE Transactions on Information Theory*, 8(2):179–187, 1962.
- [297] C.-Y. Huang, L.-m. Liu, and D. D. Hung. Fingerprint analysis and singular point detection. *Pattern Recogn. Lett.*, 28:1937–1945, November 2007.
- [298] J. Huang, S. Kumar, M. Mitra, W. Zhu, and R. Zabih. Image Indexing Using Color Correlograms. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 762–768, Puerto Rico, June 1997.
- [299] M.-x. Huang, C.-x. Xing, and J.-j. Yang. A cooperative framework of service chain for digital library. In *Proceedings of the 2009 33rd Annual IEEE International Computer Software and Applications Conference - Volume 02*, COMPSAC '09, pages 359–364, Washington, DC, USA, 2009. IEEE Computer Society.
- [300] J. Hunter and S. Choudhury. A semi-automated digital preservation system based on semantic web services. In *Proceedings of the Fourth ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 269–278, Tucson, Arizona, 2004.
- [301] T. K. Huwe. Exploiting Synergies Among Digital Repositories, Special Collections, and Online Community. *Online*, 33(2):14–19, 2009.
- [302] J. Impagliazzo. Using citidel as a portal for cs education. In *CCSCNE Conference*, 2002. Panel Presentation and Chair (with L. Cassel and D. Knox).
- [303] P. Ingwersen, K. van Rijsbergen, and N. Belkin. Context in Information Retrieval. [http://ir.dcs.gla.ac.uk/context/IRinContext\\_WorkshopNotes\\_SIGIR2004.pdf](http://ir.dcs.gla.ac.uk/context/IRinContext_WorkshopNotes_SIGIR2004.pdf), 2004.

## 452 B. GLOSSARY

- [304] H. Ino, M. Kudo, and A. Nakamura. Partitioning of web graphs by community topology. In *Proceedings of the 14th international conference on World Wide Web, WWW '05*, pages 661–669. ACM, 2005.
- [305] U. Institute for Information Technologies in Education: Analytical Survey. Digital Libraries in Education, Science and Culture. <http://iite.unesco.org/pics/publications/en/files/3214660.pdf>, Moscow, 2007.
- [306] InternetArchive. Internet archive, 2000.
- [307] Y. Ioannidis. User working group: Towards user interoperability. In *First DL.org Workshop at European Conference on Research and Advanced Technology for Digital Libraries*, 2009.
- [308] Y. Ioannidis, D. Maier, S. Abiteboul, P. Buneman, S. Davidson, E. A. Fox, A. Halevy, C. Knoblock, F. Rabitti, H.-J. Schek, and G. Weikum. Digital library information-technology infrastructures. *International Journal on Digital Libraries*, 5(4):266–274, 2005.
- [309] P. G. Ipeirotis and L. Gravano. Distributed search over the hidden Web: hierarchical database sampling and selection. In *Proceedings of the 28th Int. Conference on Very Large Data Bases, Hong Kong SAR, China, 20–23 August 2002*, pages 394–405, 2002.
- [310] P. H. J. Bekaert and H. V. de Sompel. Using mpeg-21 didl to represent complex digital objects in the los alamos national laboratory digital library, available at <http://www.dlib.org/dlib/november03/bekaert/11bekaert.html>. *D-Lib Magazine*, 9(11), November 2003.
- [311] S. C. Jane Hunter. Implementing Preservation Strategies for Complex Multimedia Objects. In *Proc. 7th European Conf. Research and Advanced Technology for Digital Libraries, ECDL 2003*, pages 473–486, Trondheim, Norway, August 17-22, 2003.
- [312] G. Janée and J. Frew. The ADEPT digital library architecture. In *Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pages 342–350, Portland, Oregon, USA, 2002. ACM.
- [313] R. E. Jenkins and N. M. Burkhead. *Freshwater Fishes of Virginia*. American Fisheries Society, Bethesda, Maryland, 1993.
- [314] F. Jiao, S. Wang, C. Lee, R. Greiner, and D. Schuurmans. Semi-supervised conditional random fields for improved sequence segmentation and labeling. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 209–216. Association for Computational Linguistics, 2006.
- [315] T. Joachims, T. Hofmann, Y. Yue, and C. Yu. Predicting structured objects with support vector machines. *Communications of the ACM*, 52(11):97–104, 2009.

- [316] N. Johnson and S. Jajodia. Exploring steganography: Seeing the unseen. *Computer*, 31(2):26–34, feb. 1998.
- [317] R. K. Johnson. Institutional repositories: Partnering with faculty to enhance scholarly communication. *D-Lib Magazine*, 8(11), 2002.
- [318] C. Jones. Geographic information retrieval, Nov. 2006. Presentation.
- [319] C. B. Jones, A. I. Abdelmoty, D. Finch, G. Fu, and S. Vaid. The SPIRIT spatial search engine: Architecture, ontologies and spatial indexing. In *Geographic Information Science, Lecture Notes in Computer Science*, pages 125–139. Springer, 2004.
- [320] C. B. Jones and R. S. Purves. Geographical information retrieval. *International Journal of Geographical Information Science*, 22(3):219, 2008.
- [321] R. Jones, A. Hassan, and F. Diaz. Geographic features in web search retrieval. In *Proceeding of the 2nd international workshop on Geographic information retrieval*, pages 57–58, Napa Valley, California, USA, 2008. ACM.
- [322] W. Jones and J. Teevan, editors. *Personal Information Management*. University of Washington Press, 2007.
- [323] M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural Comput.*, 6(2):181–214, 1994.
- [324] T. H. Jordan. Scce 2009 annual report. *Southern California Earthquake Center*, 2009.
- [325] J. Kahan and M.-R. Koivunen. Annotea: an open RDF infrastructure for shared web annotations. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 623–632, New York, NY, USA, 2001. ACM.
- [326] B. Kahin and H. R. Varian. *Internet Publishing and Beyond: The Economics of Digital Information and Intellectual Property*. MIT Press, Cambridge, Massachusetts, 2000.
- [327] R. Kahn and R. Wilensky. A framework for distributed digital object services. Technical report cnri.dlib/tn95-01, CNRI, Reston, VA, May 1995.
- [328] N. Kampanya, R. Shen, S. Kim, C. North, and E. A. Fox. Citiviz: A visual user interface to the citidel system. In *Proc. European Conference on Digital Libraries (ECDL) 2004, September 12-17, University of Bath, UK*. Springer, 2004.
- [329] V. Karavirta and P. Ihantola. Initial set of services for algorithm visualization. In *Proceedings of the Sixth Program Visualization Workshop*, pages 67–71, Darmstadt, Germany, 2011.

## 454 B. GLOSSARY

- [330] J. F. Karpovich, A. S. Grimshaw, and J. C. French. Extensible file system (elfs): an object-oriented approach to high performance file i/o. *ACM SIGPLAN Notices*, 29(10):191–204, 1994.
- [331] H. Kautz, B. Selman, and M. Shah. Referral Web: Combining social networks and collaborative filtering. *Communications of the ACM*, 40(3):63–65, 1997.
- [332] R. Kelapure. Scenario-based generation of digital library services. Master’s thesis, Virginia Tech, 2003.
- [333] P. Kelm, S. Schmiedeke, and T. Sikora. Multi-modal, Multi-resource Methods for Placing Flickr Videos on the Map. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, ICMR ’11, 2011.
- [334] D. A. Kemp. Relevance, pertinence, and information system development. *Information Storage and Retrieval*, 10(2):37–47, 1974.
- [335] A. Kerne. recombinant information workshop [interface ecology lab], 2005.
- [336] A. Kerne, E. Koh, B. Dworaczyk, M. J. Mistrot, H. Choi, S. M. Smith, R. Graeber, D. Caruso, A. Webb, R. Hill, and J. Albea. combinformation: a mixed-initiative system for representing collections as compositions of image and text surrogates. In *JCDL ’06: Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*, pages 11–20, New York, NY, USA, 2006. ACM.
- [337] A. Kerne, E. Koh, S. M. Smith, A. Webb, and B. Dworaczyk. combinformation: Mixed-initiative composition of image and text surrogates promotes information discovery. *ACM Trans. Inf. Syst.*, 27(1):1–45, 2008.
- [338] M. M. Kessler. Bibliographic coupling between scientific papers. *American Documentation*, 14(1):10–25, 1963. A short description of a simple method for finding related and grouping technical or scientific papers by the citations (one citation!) they share. An example is shown on 36 volumes of articles from Physical Review.
- [339] S. Kethers, X. Shen, A. E. Treloar, and R. G. Wilkinson. Discovering australia’s research data. In *Proceedings of the 10th ACM/IEEE Joint Conference on Digital libraries*, pages 345–348, 2010.
- [340] S. A. Khirni, B. Yang, R. Purves, and M. Kopczynski. Query interface design. Project Report W4 D74101, University of Zurich, Zurich, Switzerland, Zurich, Switzerland, 2003.
- [341] M. Kimpton and S. Payette. Duraspaces. In *Proceedings of the 4th International Conference on Open Repositories*, OR09, Atlanta, GA, USA, 2009. Georgia Institute of Technology.

- [342] C.-P. Klas, N. Fuhr, S. Kriewel, H. Albrechtsen, G. Tsakonas, S. Kapidakis, C. Papatheodorou, P. Hansen, L. Kovacs, A. Micsik, and E. Jacob. An experimental framework for comparative digital library evaluation: the logging scheme. In *Proceedings of the 6th ACM/IEEE-CS joint conference on digital libraries*, pages 308–309, New York, NY, USA, 2006. ACM Press.
- [343] J. Koh, Y. Kim, B. Butler, and G. Bock. Encouraging Participation in Virtual Communities. *Commun. ACM*, 50:68–73, February 2007.
- [344] U. Kohl, J. Lotspiech, and S. Nusser. Security for the digital library-protecting documents rather than channels. In *Database and Expert Systems Applications, 1998. Proceedings. Ninth International Workshop on*, pages 316 –321, aug 1998.
- [345] D. Koller and M. Sahami. Hierarchically classifying documents using very few words. In *ICML '97: Proceedings of the Fourteenth International Conference on Machine Learning*, pages 170–178, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.
- [346] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl. GroupLens: Applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3):77–87, 1997. Konstan, Joseph/Applying Collaborative Filtering to Usenet News.pdf.
- [347] I. Konstas, V. Stathopoulos, and J. M. Jose. On social networks and collaborative recommendation, 2009.
- [348] N. P. Kozievitch. Complex Objects in Digital Libraries. In *JCDL '09: Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries, Doctoral Consortium*, available <http://www.ieee-tcdl.org/Bulletin/v5n3/Kozievitch/kozievitch.html> - last accessed on 05/05/11, 2009.
- [349] N. P. Kozievitch. Reusing a Compound-Based Infrastructure for Searching Video Stories. In *12th IEEE International Conference on Information Reuse and Integration (IRI)*. IEEE, 2011.
- [350] N. P. Kozievitch, S. Codio, J. A. Francois, E. Fox, and R. da S. Torres. Exploring CBIR concepts in the CTRnet Project. Technical Report IC-10-32, Institute of Computing, University of Campinas, November 2010. In English, 20 pages.
- [351] N. P. Kozievitch and R. da S. Torres. Describing oai-ore from the 5s framework perspective. In *Proceedings of the role of digital libraries in a time of global change, and 12th international conference on Asia-Pacific digital libraries*, ICADL'10, pages 260–261, Berlin, Heidelberg, 2010. Springer-Verlag.
- [352] N. P. Kozievitch, R. da S. Torres, S. H. Park, E. A. Fox, N. Short, A. L. Abbott, S. Misra, and M. Hsiao. Database for fingerprint experiments. *Poster for CESCA (Center for Embedded Systems for Critical Applications) Day, Virginia Tech, Blacksburg, VA, USA*, 5 2010.

## 456 B. GLOSSARY

- [353] N. P. Kozievitch, R. da S. Torres, S. H. Park, E. A. Fox, N. Short, A. L. Abbott, S. Misra, and M. Hsiao. Rethinking Fingerprint Evidence Through Integration of Very Large Digital Libraries. *VLDL Workshop at 14th European Conference on Research and Advanced Technology for Digital Libraries (ECDL2010)*, pages 23–30, 07 2010.
- [354] N. P. Kozievitch, R. da Silva Torres, F. Andrade, U. Murthy, E. A. Fox, and E. Hallerman. A teaching tool for parasitology: Enhancing learning with annotation and image retrieval. In *Proc. ECDL 2010*, pages 466–469, 2010.
- [355] N. P. Kozievitch, E. Fox, and R. da S. Torres. Analyzing Compound Object Technologies from the 5S Perspective. Technical Report IC-11-01, Institute of Computing, University of Campinas, 2011.
- [356] D. B. Krafft, A. Birkland, and E. J. Cramer. Ncore: architecture and implementation of a flexible, collaborative digital library. In *JCDL '08: Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*, pages 313–322, New York, NY, USA, 2008. ACM.
- [357] S. Kumar, S. K. Das, and R. Biswas. Graph partitioning for parallel applications in heterogeneous grid environments. In *Parallel and Distributed Processing Symposium - (IPDPS'02)*, pages 66–72, 2002.
- [358] M. Kying. Creating contexts for design. In *Scenario-Based Design: Envisioning Work and Technology in System Development*. John Wiley & Sons, New York, NY, USA, 1995.
- [359] M. Kyriolidou and S. Giersch. Developing the digual protocol for digital library evaluation. In *JCDL '05: Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*, pages 172–173, New York, NY, USA, 2005. ACM Press.
- [360] A. Laender, B. Ribeiro-Neto, A. da Silva, and J. Teixeira. A brief survey of web data extraction tools. *ACM Sigmod Record*, 31(2):84–93, 2002.
- [361] A. H. F. Laender, M. A. Gonçalves, and P. A. Roberto. BDBComp: building a digital library for the Brazilian computer science community. In *Proceedings of the 4th ACM/IEEE-CS joint conference on digital libraries*, pages 23–24, New York, NY, USA, 2004. ACM Press.
- [362] B. Lagoero, M. A. Gonçalves, and E. A. Fox. 5squal: A quality tool for digital libraries. In *Proceedings of the 7th ACM/IEEE Joint Conference on Digital Libraries*, page (demonstration accepted), New York, NY, USA, 2007. ACM Press.
- [363] C. Lagoze. A secure repository design for digital libraries. *D-Lib Magazine*, 1(12), December 1995. December, ISOS.
- [364] C. Lagoze, W. Arms, S. Gan, D. Hillmann, C. Ingram, D. Krafft, R. Marisa, J. Phipps, J. Saylor, C. Terrizzi, W. Hoehn, D. Millman, J. Allan, S. Guzman-Lara, and Kalt. Core

- services in the architecture of the national science digital library (nsdl). In *JCDL'02: Proceedings of the 2nd ACM/IEEE-CS Joint Conference on Digital Libraries, Houston, TX*, pages 201–209, 2002.
- [365] C. Lagoze and H. V. de Sompel. The Open Archives Initiative: building a low-barrier interoperability framework. In *JCDL '01: Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries*, pages 54–62, New York, NY, USA, 2001. ACM Press.
  - [366] C. Lagoze, D. B. Krafft, S. Payette, and S. Jesuropgai. What is a digital library anyway? beyond search and access in the NSDL. *D-Lib magazine*, 11(11), 2005.
  - [367] C. Lagoze and H. V. Sompel. Compound information objects: the oai-ore perspective. *Open Archives Initiative Object Reuse and Exchange, White Paper*, <http://www.openarchives.org/ore/documents>, 2007.
  - [368] A. V. Lamsweerde and L. Willemet. Inferring declarative requirements specifications from operational scenarios. *IEEE Trans. on Soft. Engineering*, 24(12):1089–1114, December 1998.
  - [369] R. L. Larsen and H. D. Wactlar. *Knowledge Lost in Information: Report of the NSF Workshop on Research Directions for Digital Libraries, June 15-17, 2003, Chatham, MA*. University of Pittsburgh, Pittsburgh, 2004.
  - [370] M. Larson, M. Soleymani, P. Serdyukov, S. Rudinac, C. Wartena, V. Murdock, G. Friedland, R. Ordelman, and G. J. F. Jones. Automatic tagging and geotagging in video collections and communities. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval, ICMR '11*, pages 51:1–51:8, New York, NY, USA, 2011. ACM.
  - [371] R. R. Larson. Geographic information retrieval and spatial browsing. In L. C. Smith and M. Gluck, editors, *Geographic information systems and libraries: patrons, maps, and spatial information: papers presented at the 1995 Clinic on Library Applications of Data Processing, April 10-12, 1995*, pages 81–124. Graduate School of Library and Information Science, University of Illinois at Urbana-Champaign, University of Illinois, Urbana-Champaign, Apr. 1995.
  - [372] R. R. Larson. Geographic information retrieval and digital libraries. In *Proceeding of 13th European Conference, ECDL 2009*, volume Volume 5714/2009, pages 461–464. Springer Berlin / Heidelberg, Sept 2009. DOI 10.1007/978-3-642-04346-8.
  - [373] O. Lassila and R. R. Swick. Resource description framework (RDF). model and syntax specification. Technical report, W3C, 2 1999.
  - [374] A. D. Learning. Sharable content object reference model (scorm), 2004.

## 458 B. GLOSSARY

- [375] W. G. LeFurgy. Pdf/a: Developing a file format for long-term preservation. *RLG DigiNews*, 7(6), 2003.
- [376] J. Leidig, E. Fox, M. Marathe, and H. Mortveit. Epidemiology experiment and simulation management through schema-based digital libraries. In *Proceedings of the 2nd DL.org Workshop at ECDL, Making Digital Libraries Interoperable: Challenges and Approaches*, pages 57–66, 2010.
- [377] J. Leidig, E. A. Fox, K. Hall, M. Marathe, and H. Mortveit. Simdl: a model ontology driven digital library for simulation systems. In *Proceeding of the 11th annual international ACM/IEEE joint conference on Digital libraries, JCDL '11*, pages 81–84, New York, NY, USA, 2011. ACM.
- [378] J. M. Leimeister, P. Sidiras, and H. Krcmar. Success Factors of Virtual Communities from the Perspective of Members and Operators: An Empirical Study. In *Proceedings of the 37th Annual Hawaii International Conference on System Sciences, HICSS '04*, pages 1530–1605, 2004.
- [379] P. P. Leonardo Candela, Donatella Castelli. D4science: an e-infrastructure for supporting virtual research. In *5th Italian Research Conference on Digital Libraries (IRCDL)*, pages 166–169, 2009.
- [380] M. Lesk. Perspectives on DLI-2 - growing the field. *D-Lib Magazine*, 5(7/8), 1999.
- [381] M. Lesk. *Understanding Digital Libraries*, 2nd ed. San Francisco: Morgan Kaufmann, 2004.
- [382] J. Leveling and S. Hartrumpf. On metonymy recognition for geographic information retrieval. *International Journal of Geographical Information Science*, 22(3):289, 2008.
- [383] D. M. Levy. Heroic measures: reflections on the possibility and purpose of digital preservation. In *DL '98: Proceedings of the 3rd ACM International Conference on Digital Libraries*, pages 152–161, Pittsburgh, PA, 1998.
- [384] D. M. Levy and C. C. Marshall. Going Digital: A Look at Assumptions Underlying Digital Libraries. *Communications of the ACM*, 38:77–84, April 1995.
- [385] M. Leyton. Symmetry-Curvature Duality. *Computer Vision, Graphics, and Image Processing*, 38(3):327–341, 1987.
- [386] L. T. Li and R. d. S. Torres. Coping with geographical relationships in web searches. Technical Report IC-10-04, Institute of Computing, University of Campinas, Jan. 2010.
- [387] N. Li, L. Zhu, P. Mitra, K. Mueller, E. Poweleit, and C. L. Giles. oreChem ChemXSeer: a semantic digital library for chemistry. In *Proceedings of the 10th ACM/IEEE Joint Conference on Digital libraries*, 2010.

- [388] X. Li, Y. Wang, and A. Acero. Extracting structured information from user queries with semi-supervised conditional random fields. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 572–579. ACM, 2009.
- [389] LibraryOfCongress. METS - Metadata Encoding and Transmission Standard. <http://www.loc.gov/standards/mets/>, 2003.
- [390] J. C. R. Licklider. *Libraries of the Future*. MIT Press, Cambridge, Massachusetts, 1965.
- [391] H. Lin. Determinants of Successful Virtual Communities: Contributions from System Characteristics and Social Factors. *Information and Management*, 45(8):522–527, 2008.
- [392] T. Y. Liu, Y. Yang, H. Wan, H. J. Zeng, Z. Chen, and W. Y. Ma. Support vector machines classification with a very large-scale taxonomy. *SIGKDD Explor. Newsl.*, 7(1):36–43, 2005.
- [393] LOMcommittee. *IEEE Standard for Learning Object Metadata*. IEEE, 2002. IEEE 1484.12.1-2002.
- [394] S. Loncaric. A Survey of Shape Analysis Techniques. *Pattern Recognition*, 31(8):983–1190, Aug 1998.
- [395] P. Lopez. Automatic extraction and resolution of bibliographical references in patent documents. *Advances in Multidisciplinary Retrieval*, pages 120–135, 2010.
- [396] C. Lopez-Pujalte, V. P. G. Bote, and F. de Moya Anegon. Order-based fitness functions for genetic algorithms applied to relevance feedback. *Journal of the American Society for Information Science and Technology*, 54(2):152–160, 2003.
- [397] R. A. Lorie. Long term preservation of digital information. In *JCDL 2001*, pages 346–352, Roanoke, VA, 2001. ACM.
- [398] R. A. Lorie. A methodology and system for preserving digital data. In *JCDL 2002*, pages 312–319, Portland, Oregon, 2002. ACM.
- [399] B. Ludäscher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. A. Lee, J. Tao, and Y. Zhao. Scientific workflow management and the kepler system. *Concurrency and Computation: Practice and Experience*, 18(10):1039–1065, 2006.
- [400] P. J. Ludford, D. Cosley, D. Frankowski, and L. Terveen. Think Different: Increasing Online Community Participation using Uniqueness and Group Dissimilarity. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 631–638, 2004.
- [401] J. Luo, D. Joshi, J. Yu, and A. Gallagher. Geotagging in multimedia and computer vision—a survey. *Multimedia Tools Appl.*, 51:187–211, January 2011.

## 460 B. GLOSSARY

- [402] C. Lynch, S. Parastatidis, N. Jacobs, H. Van de Sompel, and C. Lagoze. The oai-ore effort: progress, challenges, synergies. In *JCDL '07: Proceedings of the 2007 conference on Digital libraries*, page 80. ACM, 2007.
- [403] C. A. Lynch. The z39.50 information retrieval standard part i: A strategic view of its past, present and future. *D-Lib Magazine*, 3(4), 1997. April.
- [404] C. A. Lynch. Institutional repositories: Essential infrastructure for scholarship in the digital age. *ARL Bimonthly Report*, 226, February 2003 2003.
- [405] W. Y. Ma and B. S. Manjunath. Netra: A Toolbox for Navigating Large Image Databases. In *IEEE International Conference on Image Processing*, pages 256–268, 1997.
- [406] S. D. MacArthur, C. E. Brodley, A. C. Kak, and L. S. Broderick. Interactive Content-Based Image Retrieval Using Relevance Feedback. *Computer Vision and Image Understanding*, 88(2):55–75, 2002.
- [407] W. E. Mackay and M. Beaudouin-Lafon. DIVA exploratory data analysis with multimedia streams. In *Proc. of CHI-98*, pages 416–423, Los Angeles, CA, USA, Apr. 18-23, 1998.
- [408] D. Maier and L. M. L. Delcambre. Superimposed information for the internet. In *WebDB (Informal Proceedings)*, pages 1–9, 1999.
- [409] S. Majithia, M. S. Shields, I. J. Taylor, and I. Wang. Triana: A graphical web service composition and execution toolkit. In *Proceedings of the IEEE International Conference on Web Services*, 2004.
- [410] T. Malloy and G. Hanley. Merlot: A faculty-focused web site of educational resources. *Behavior Research Methods*, 33:274–276, 2001. 10.3758/BF03195376.
- [411] L. Malmi, V. Karavirta, A. Korhonen, J. Nikander, O. Seppälä, and P. Silvasti. Visual algorithm simulation exercise system with automatic assessment: Trakla2. *Informatics in Education*, 3(2):267–288, 2004.
- [412] D. Maltoni and R. Cappelli. Advances in fingerprint modeling. *Image Vision Comput.*, 27:258–268, February 2009.
- [413] U. Manber, M. Smith, and B. Gopal. Webglimpse: Combining browsing and searching. In *Proceedings of Usenix Technical Conference*, pages 195–206, 1997.
- [414] P. Manghi, M. Mikulicic, L. Candela, D. Castelli, and P. Pagano. Realizing and Maintaining Aggregative Digital Library Systems: D-NET Software Toolkit and OAster System. *D-Lib Magazine*, available at <http://www.dlib.org/dlib/march10/manghi/03manghi.html>, 16(3/4), 2010.

- [415] B. S. Manjunath and W. Y. Ma. Texture Features for Browsing and Retrieval of Image Data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–842, August 1996.
- [416] B. S. Manjunath, J. R. Ohm, V. V. Vasudevan, and A. Yamada. Color and Texture Descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):703–715, June 2001.
- [417] G. Mann and A. McCallum. Simple, robust, scalable semi-supervised learning via expectation regularization. In *Proceedings of the 24th international conference on Machine learning*, pages 593–600. ACM, 2007.
- [418] G. Marchionini. *Information Seeking in Electronic Environments*. Cambridge University Press, Cambridge, 1995.
- [419] M. Markland. Technology and People: Some Challenges when Integrating Digital Library Systems into Online Learning Environments. *The New Review of Information and Library Research*, 9(1):85–96, 2003.
- [420] C. C. Marshall. Annotation: from Paper Books to the Digital Library. In *DL'97*, pages 233–240, 1997.
- [421] B. Martins, M. J. Silva, and L. Andrade. Indexing and ranking in Geo-IR systems. In *GIR '05: Proceedings of the 2005 workshop on Geographic information retrieval*, pages 31–34, New York, NY, USA, 2005. ACM.
- [422] A. Maslov, J. Creel, A. Mikeal, S. Phillips, J. Leggett, and M. McFarland. Adding OAI-ORE Support to Repository Platforms. *Journal of Digital Information (JoDI)*, 11(1), 2010.
- [423] R. McGreal. A typology of learning object repositories. In H. H. Adelsberger, P. Kinshuk, J. M. Pawlowski, and D. G. Sampson, editors, *Handbook on Information Technologies for Education and Training*, International Handbooks on Information Systems, pages 5–28. Springer Berlin Heidelberg, 2008.
- [424] B. M. Mehtre, M. S. Kankanhalli, and W. F. Lee. Shape Measures for Content Based Image Retrieval: A Comparison. *Information Processing and Management*, 33(3):319–337, 1997.
- [425] J. Melton and A. Eisenberg. Sql multimedia and application packages (sql/mm). *SIGMOD Rec.*, 30(4):97–102, 2001.
- [426] S. E. Middleton, D. D. Roure, and N. R. Shadbolt. Ontology-based recommender systems. In S. Staab and D. Rudi Studer, editors, *Handbook on Ontologies*, International Handbooks on Information Systems, pages 779–796. Springer Berlin Heidelberg, 2009. 10.1007/978-3-540-92673-3\_35.

## 462 B. GLOSSARY

- [427] D. R. Millen and J. F. Patterson. Stimulating Social Engagement in a Community Network. In *Proceedings of the 2002 ACM conference on Computer supported cooperative work*, pages 306–313, 2002.
- [428] G. A. Miller. Wordnet: a lexical database for english. *Commun. ACM*, 38:39–41, November 1995.
- [429] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. Wordnet: An on-line lexical database. *International Journal of Lexicography*, 3:235–244, 1990.
- [430] R. Miller, O. Tsatalos, and J. Williams. DataWeb: Customizable Database Publishing for the Web. *Multimedia, IEEE*, 4(4):14–21, 1997.
- [431] M. Minsky. A framework for representing knowledge. In J. Haugeland, editor, *Mind Design*. MIT Press, Cambridge, Massachusetts, 1981.
- [432] F. Mintzer, J. Lotspiech, and N. Morimoto. Safeguarding digital library content and users. *D-Lib Magazine*, 3, 1997. <http://www.dlib.org/dlib/december97/ibm/12lotspiech.html>.
- [433] P. Miranda, R. da S. Torres, and A. X. Falcão. TSD: A Shape Descriptor Based on a Distribution of Tensor Scale Local Orientation. In *XVIII Brazilian Symposium on Computer Graphics and Image Processing*, pages 139–146, Natal, RN, Brazil, October 2005.
- [434] R. Mirandola and D. Hollinger. A New Approach to Performance Modelling of Client/Server Distributed DataBase Architectures. *Performance Evaluation*, Elsevier Science, 29(4):255–272, 1997.
- [435] S. Mizzaro. A cognitive analysis of information retrieval. In *Information Science: Integration in Perspective – Proceedings of CoLIS2*, pages 233–250, Copenhagen, Denmark, 1996.
- [436] S. Mizzaro. How many relevances in information retrieval? *Interacting With Computers*, 10(3):305–322, 1998.
- [437] F. Mokhtarian and S. Abbasi. Shape Similarity Retrieval Under Affine Transforms. *Pattern Recognition*, 35(1):31–41, January 2002.
- [438] C. Monroy, R. Furuta, and F. Castro. Using an ontology and a multilingual glossary for enhancing the nautical archaeology digital library. In *Proceedings of the 10th ACM/IEEE Joint Conference on Digital libraries*, pages 259–262, 2010.
- [439] C. S. Mooers. Coding information retrieval and the rapid selector. *American Documentation*, 1(4):225–29, 1950.
- [440] R. W. Moore, A. Rajasekar, M. Wan, Y. Katsis, D. Zhou, A. Deutsch, and Y. Papakonstantinou. Constraint-based knowledge systems for grids, digital libraries, and persistent archives: Yearly report. Technical report, SDSC, University of California, San Diego, 2005.

- [441] L. Moreau, B. Clifford, J. Freire, Y. Gil, P. Groth, J. Futrelle, N. Kwasnikowska, S. Miles, P. Missier, J. Myers, Y. Simmhan, E. Stephan, and J. V. den Bussche. The open provenance model — core specification (v1.1). In *JCDL '09: Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries, Doctoral Consortium, available <http://www.ieee-tcdl.org/Bulletin/v5n3/Kozievitch/kozievitch.html> - last accessed on 05/05/11*. Elseview, 2009.
- [442] P. Mukhopadhyay and Y. Papakonstantinou. Mixing querying and navigation in mix. In *Proceedings of ICDE*, pages 245–254, 2002.
- [443] H. Muller, N. Michoux, D. Bandon, and A. Geissbuhler. A Review of Content-Based Image Retrieval Systems in Medical Applications – Clinical Benefits and Future Directions. *International Journal of Medical Informatics*, 73(1):1–23, Feb 2004.
- [444] K. Munroe and Y. Papakonstantinou. Bbq: A visual interface for integrated browsing and querying of xml. In *Proceedings of VDB*, pages 277–296, 2002.
- [445] S. Murthy, L. Delcambre, and D. Maier. Explicitly representing superimposed information in a conceptual model. In *Conceptual Modeling - ER 2006*, pages 126–139, Berlin, Heidelberg, 2006. Springer-Verlag.
- [446] S. Murthy, D. Maier, and L. Delcambre. Querying bi-level information. In *Proceedings of WebDB workshop*, 2004.
- [447] S. Murthy, D. Maier, L. Delcambre, and S. Bowers. Putting integrated information into context: Superimposing conceptual models with sparce. In *Proceedings of the First Asia-Pacific Conference of Conceptual Modeling*, pages 71–80, Denedin, New Zealand, 2004.
- [448] S. Murthy, D. Maier, L. Delcambre, and S. Bowers. Superimposed applications using SPARCE. In *Proceedings of the 20th International Conference on Data Engineering*, Los Alamitos, CA, USA, 2004. IEEE Computer Society.
- [449] U. Murthy. A superimposed information-supported digital library, 2007. Presented at the Doctoral Consortium - held in conjunction with JCDL'07.
- [450] U. Murthy. *Digital Libraries with Superimposed Information: Supporting Scholarly Tasks that Involve Fine Grain Information*. Ph.d. dissertation, Virginia Tech, 2011.
- [451] U. Murthy, K. Ahuja, S. Murthy, and E. A. Fox. SIMPEL: a superimposed multimedia presentation editor and player. In *JCDL '06: Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries*, page 377, 2006.
- [452] U. Murthy, E. A. Fox, Y. Chen, E. Hallerman, R. Torres, E. Ramos, and T. Falcao. Superimposed image description and retrieval for fish species identification. *13th European Conference on Research and Advanced Technology for Digital Libraries*, 2009.

## 464 B. GLOSSARY

- [453] U. Murthy, E. A. Fox, Y. Chen, E. Hallerman, R. Torres, E. J. Ramos, and T. R. Falcao. Species identification: fish images with cbir and annotations. In *JCDL '09: Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries*, pages 435–436, New York, NY, USA, 2009. ACM.
- [454] U. Murthy, D. Gorton, R. Torres, M. Gonçalves, E. Fox, and L. Delcambre. Extending the 5S digital library (DL) framework: From a minimal DL towards a DL reference model., 2007. Presented at the First Workshop on Digital Library Foundations - held in conjunction with JCDL'07.
- [455] U. Murthy, R. Richardson, E. A. Fox, and L. Delcambre. Enhancing concept mapping tools below and above to facilitate the use of superimposed information. In A. J. Cañas and J. D. Novak, editors, *the Second International Conference on Concept Mapping*, San Jose, Costa Rica, 2006.
- [456] A. Nadeem and M. Javed. A performance comparison of data encryption algorithms. In *Information and Communication Technologies, 2005. ICICT 2005. First International Conference on*, pages 84 – 89, aug. 2005.
- [457] S. Nagaraj. Access control in distributed object systems: problems with access control lists. In *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2001. WET ICE 2001. Proceedings. Tenth IEEE International Workshops on*, pages 163 –164, 2001.
- [458] G. Navarro and R. Baeza-Yates. Proximal nodes: A model to query document databases by content and structure. *ACM Transactions on Information Systems*, 15(4):400–435, 1997.
- [459] R. Neches, R. Fikes, T. Finin, T. Gruber, R. Patil, T. Senator, and W. Swartout. Enabling technology for knowledge sharing. *AI Magazine*, 12(3):36–56, 1991.
- [460] M. L. Nelson, B. Argue, M. Efron, S. Denn, and M. C. Pattuelli. A survey of complex object technologies for digital libraries. Technical Report TM-2001-211426, NASA, 2001.
- [461] M. L. Nelson and H. V. de Sompel. IJDL special issue on complex digital objects: Guest editors' introduction. *International Journal of Digital Libraries*, 6(2):113–114, 2006.
- [462] M. L. Nelson and K. Maly. Buckets: Smart objects for digital libraries. *Communications of the ACM*, 44(5):60–61, 2001.
- [463] M. L. Nelson, K. Maly, M. Zubair, and S. N. T. Shen. Buckets: Aggregative, intelligent agents for publishing. *Webnet Journal*, 1(1):58–66, 1998.
- [464] M. L. Nelson, G. Marchionini, G. Geisler, and M. Yang. A bucket architecture for the open video project. In *Proceedings of the 1st ACM/IEEE-CS Joint Conference on Digital libraries*, JCDL '01, pages 310–311. ACM, 2001.

- [465] T. H. Nelson. Xanalogical structure, needed now more than ever: parallel documents, deep links to content, deep versioning, and deep re-use. *ACM Comput. Surv.*, 31(4es), 1999.
- [466] S. Nestorov, S. Abiteboul, and R. Motwani. Inferring structure in semistructured data. *SIGMOD Record*, 26(4):39–43, 1997.
- [467] B. Neuman and T. Ts'o. Kerberos: an authentication service for computer networks. *Communications Magazine, IEEE*, 32(9):33 –38, sep 1994.
- [468] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69:026113, Feb 2004.
- [469] N. Nguyen and Y. Guo. Comparisons of sequence labeling algorithms and extensions. In *Proceedings of the 24th international conference on Machine learning*, pages 681–688. ACM, 2007.
- [470] NISO. Information retrieval (z39.50): Application service definition and protocol specification (ansi/niso z39.501995). Technical report, NISO (National Information Standards Organization) Press, 1995.
- [471] P. Noerr. *The Digital Library Toolkit*. Sun Microsystems, Inc., Palo Alto, CA, 2nd edition, March 2000.
- [472] P. Norris. *Digital Divide: Civic Engagement, Information Poverty, and the Internet Worldwide*. Cambridge University Press, Cambridge, UK, 2001.
- [473] O. Nov, M. Naaman, and C. Ye. Analysis of Participation in an Online Photo-sharing Community: A Multidimensional Perspective. *Journal of the American Society for Information Science and Technology*, 61(3):555–566, 2010.
- [474] OAC. Open annotation collaboration.
- [475] OAI. Oai-pmh - open archives initiative protocol for metadata harvesting - v.2.0. <http://www.openarchives.org/OAI/openarchivesprotocol.html>, 2001.
- [476] OAI. The Open Archives Initiative Protocol for Metadata Harvesting – Version2.0. <http://www.openarchives.org/OAI/openarchivesprotocol.html> [last visited 2007, March 23], October 2004.
- [477] OAI. Open Archive Initiative, <http://www.openarchives.org/>, 2005.
- [478] D. Oard, C. Peters, M. Ruiz, R. Frederking, J. Klavans, and P. Sheridan. Multilingual information discovery and access (midas): A joint acm dl'99 / acm sigir'99 workshop. *D-Lib Magazine*, 5(10), Oct. 1999.

## 466 B. GLOSSARY

- [479] A. Oberweis and P. Sander. Information system behavior specification by high level Petri nets. *ACM Transactions on Information Systems*, 14(4):380–420, 1996.
- [480] OCLC. Xcat ndltd union catalog, 2004.
- [481] OCLC. SRW/U, 2010.
- [482] C. of Civil Informatics. Ontology Evaluation. <http://i2c.engineering.utoronto.ca/home/>, 2004. [Online; accessed 26-September-2011].
- [483] L. of Congress. MARC Home Page (Library of Congress), March 1998. <http://lcweb.loc.gov/marc/marc.html>.
- [484] U. of Richmond. The Disaster Database Project. <http://learning.richmond.edu/disaster/index.cfm>, 2011. [Online; accessed 26-September-2011].
- [485] R. Ogawa, H. Harada, and A. Kaneko. Scenario-based hypermedia: A model and a system. In *Proc. of the ECHT'90 European Conf. on Hypertext*, pages 38–51, 1990.
- [486] C. K. Ogden and I. Richards. *The meaning of meaning*. Trubner & Co, London, 1923.
- [487] V. E. Ogle and M. Stonebraker. Chabot: Retrieval from Relational Database of Images. *IEEE Computer*, 28(9):40–48, Sep 1995.
- [488] T. Oinn, M. Greenwood, M. Addis, N. Alpdemir, J. Ferris, K. Glover, C. Goble, A. Goderis, D. Hull, D. Marvin, P. Li, P. Lord, M. Pocock, M. Senger, R. Stevens, A. Wipat, and C. Wroe. Taverna: lessons in creating a workflow environment for the life sciences. *Concurrency and Computation: Practice and Experience*, 18(10):1067–1100, 2006.
- [489] M. A. Oliveira and N. J. Leite. A multiscale directional operator and morphological tools for reconnecting broken ridges in fingerprint images. *Pattern Recogn.*, 41:367–377, January 2008.
- [490] C. Olston and E. Chi. ScentTrails: Integrating browsing and searching on the Web. *ACM Trans. Comput.-Hum. Interact.*, 10(3):177–197, 2003.
- [491] T. O'Reilly. What Is Web 2.0. Design Patterns and Business Models for the Next Generation of Software. <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>, September 2005.
- [492] C. OSSO. DesInventar: Inventory system of the effects of disasters. <http://www.desinventar.org/>, 2011. [Online; accessed 26-September-2011].
- [493] S. Overell and S. Rüger. Using co-occurrence models for placename disambiguation. *International Journal of Geographical Information Science*, 22(3):265, 2008.

- [494] A. Paepcke, R. Brandriff, G. Janee, R. Larson, B. Ludaescher, S. Melnik, and S. Raghavan. Search middleware and the simple digital library interoperability protocol. *D-Lib Magazine*, 6(3), 2000. March.
- [495] A. Paepcke, C.-C. K. Chang, T. Winograd, and H. Garcia-Molina. Interoperability for Digital Libraries Worldwide. *Communications of the ACM*, 41(4):33–42, 1998.
- [496] G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.
- [497] J. Park and S. Ram. Information systems interoperability: What lies beneath? *Trans. ACM*, 22(4):595–632, 2004.
- [498] S. Park and E. Fox. Enriching the VT ETD-db System with References. In *Proceeding of 14th International Symposium on Electronic Theses and Dissertations*. NDLTD, 2011.
- [499] S. H. Park, N. Lynberg, J. Racer, P. McElmurray, and E. A. Fox. HTML5 ETDs. *13th International Symposium on Electronic Thesis and Dissertations (ETD 2010)*, UT Austin Libraries, Austin, 06 2010.
- [500] R. C. Pasley, P. D. Clough, and M. Sanderson. Geo-tagging for imprecise regions of different sizes. In *Proceedings of the 4th ACM workshop on Geographical information retrieval*, pages 77–82, Lisbon, Portugal, 2007.
- [501] G. Pass, R. Zabih, and J. Miller. Comparing Images Using Color Coherence Vectors. In *Proceedings of the fourth ACM international conference on Multimedia*, pages 65–73, 1996.
- [502] G. Z. Pastorello, Jr, R. D. A. Senra, and C. B. Medeiros. Bridging the gap between geospatial resource providers and model developers. In *GIS '08: Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*, pages 1–4, New York, NY, USA, 2008. ACM.
- [503] M. Patel, T. Koch, M. Doerr, and C. Tsinaraki. *Semantic Interoperability in Digital Library Systems: Report of DELOS2 Network of Excellence in Digital Libraries*. DELOS, 2005.
- [504] A. M. B. Pavani. Digital reference center - the maxwell project. In *Proceedings of the 1999 International Conference on Engineering Education*. INEER, 1999.
- [505] S. Payette, C. Blanchi, C. Lagoze, and E. A. Overly. Interoperability for digital objects and repositories: The cornell/cnri experiments. *D-Lib Magazine*, 5(5), 1999. May.
- [506] S. Payette and C. Lagoze. Flexible and extensible digital object and repository architecture (fedora). In *ECDL '98: Proceedings of the Second European Conference on Research and Advanced Technology for Digital Libraries*, pages 41–59, London, UK, 1998. Springer-Verlag.

## 468 B. GLOSSARY

- [507] PBS. PBS teachers. <http://www.pbs.org/teachers/>, March 2011.
- [508] A. Pease and I. Niles. IEEE standard upper ontology: a progress report. *The Knowledge Engineering Review*, 17(01):65–70, 2002.
- [509] O. A. Penatti and R. d. S. Torres. Eva: an evaluation tool for comparing descriptors in content-based image retrieval tasks. In *Proceedings of the international conference on Multimedia information retrieval*, MIR ’10, pages 413–416, New York, NY, USA, 2010. ACM.
- [510] F. Peng and A. McCallum. Information extraction from research papers using conditional random fields. *Information Processing & Management*, 42(4):963–979, 2006.
- [511] A. Pentland, R. Picard, and S. Sclaroff. Photobook: Content-based Manipulation of Image Databases. In *SPIE Storage and Retrieval for Image and Video Databases II*, pages 34–47, San Jose, CA, 1994.
- [512] E. Persoon and K. Fu. Shape Discrimination Using Fourier Descriptors. *IEEE Transactions on Systems, Man, and Cybernetics*, 7(3):170–178, 1977.
- [513] S. Perugini, K. McDevitt, R. Richardson, M. Perez-Quinones, R. Shen, N. Ramakrishnan, C. Williams, and E. A. Fox. Enhancing usability in citidel: Multimodal, multilingual, and interactive visualization interfaces. In *Proceedings Fourth ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL2004), Tucson, AZ, June 7-11*, pages 315–324. IEEE-CS, 2004.
- [514] T. A. Phelps and R. Wilensky. Multivalent documents. *Communications of the ACM*, 43(6):82–90, 2000.
- [515] T. A. Phelps and R. Wilensky. Robust Intra-document Locations. *Computer Networks: The International Journal of Computer and Telecommunications Networking*. Elsevier North-Holland, Inc., New York, USA, 33(1-6):105–118, 2000.
- [516] I. J. PicHunter, M. L. Miller, T. P. Minka, T. V. Papathomas, and P. N. Yianilos. The Bayesian Image Retrieval System, PicHunter: Theory, Implementation, and Psychophysical Experiments. *IEEE Transactions on Image Processing*, 9(1):20–37, January 2000.
- [517] L. L. Pipino, Y. W. Lee, and R. Y. Wang. Data quality assessment. *Communications of the ACM*, 45(4):211–218, 2002.
- [518] J. Pomerantz, S. Oh, B. Wildemuth, S. Yang, and E. A. Fox. Digital library education in computer science programs. In *Proc. 7th ACM/IEEE-CS Joint Conference on Digital Libraries, Vancouver, British Columbia, Canada, June 18-23*. ACM, 2007.
- [519] J. Pomerantz, S. Oh, S. Yang, E. A. Fox, and B. Wildemuth. The core: Digital library education in library and information science programs. *D-Lib Magazine*, 12(11), 2006.

- [520] J. Pomerantz, B. Wildemuth, E. A. Fox, and S. Yang. Curriculum development for digital libraries. In *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 175–184. ACM, New York, 2006.
- [521] A. Popescu, G. Grefenstette, and P. A. Moëllic. Gazetiki: automatic creation of a geographical gazetteer. In *Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*, pages 85–93, Pittsburgh PA, PA, USA, 2008. ACM.
- [522] H. J. Porck and R. Teygeler. *Preservation Science Survey: An Overview of Recent Developments in Research on the Conservation of Selected Analog Library and Archival Materials*. CLIR, Washington, D.C., 2000.
- [523] J. Preece, B. Nonnemeke, and D. Andrews. The Top Five Reasons for Lurking: Improving Community Experiences for Everyone. *Computers in Human Behavior*, 20(2):201–223, 2004.
- [524] R. Prince, J. Su, H. Tang, and Y. Zhao. The design of an interactive online help desk in the Alexandria Digital Library. In *Proc. of the Int. Joint Conf. on Work Activities and Collaboration: WACC '99*, pages 217–226, San Francisco, CA, 1999.
- [525] A. Project. ARROW: Australian Research Repositories Online to the World. <http://www.arrow.edu.au/>, October 2011.
- [526] R. S. Purves, P. Clough, C. B. Jones, A. Arampatzis, B. Bucher, D. Finch, G. Fu, H. Joho, A. K. Syed, S. Vaid, and B. Yang. The design and implementation of SPIRIT: a spatially aware search engine for information retrieval on the internet. *International Journal of Geographical Information Science*, 21(7):717–745, 2007.
- [527] L. Rabiner and B. Juang. An introduction to hidden markov models. *ASSP Magazine, IEEE*, 3(1):4–16, 1986.
- [528] A. Rae, V. Murdock, P. Serdyukov, and P. Kelm. Working notes for the placing task at MediaEval 2011. In *Working Notes Proceedings of the MediaEval 2011 Workshop, Santa Croce in Fossabanda, Pisa, Italy, September 1-2, 2011.*, volume Vol-807 of *CEUR Workshop Proceedings (CEUR-WS)*, Pisa, Italy, Sept. 2011. CEUR-WS.org.
- [529] B. G. Raggad. *Information Security Management*. CRC Press, 2010.
- [530] A. Raghavan, D. Rangarajan, R. Shen, M. Goncalves, N. Vemuri, W. Fan, and E. A. Fox. Schema mapper: A visualization tool for dl integration. In *JCDL '05: Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 414–414, 2005.
- [531] A. Raghavan, N. S. Vemuri, R. Shen, M. A. Gonçalves, W. Fan, and E. A. Fox. Incremental, semi-automatic, mapping-based integration of heterogeneous collections into archaeological digital libraries: Megiddo case study. In *ECDL '05: Proceedings of the 5th European Conference on Digital Libraries*, pages 139–150, 2005.

## 470 B. GLOSSARY

- [532] S. Ram, J. Park, and D. Lee. Digital libraries for the next millennium: Challenges and research directions. *Information Systems Frontiers*, 1(1):75–94, 1999.
- [533] D. Ramage, S. Dumais, and D. Liebling. Characterizing Microblogs with Topic Models. In *ICWSM*, 2010.
- [534] N. Ramakrishnan. Pipe: Web personalization by partial evaluation. *IEEE Internet Computing*, 4(6):21–31, 2000.
- [535] S. R. Ranganathan. *A Descriptive Account of Colon Classification*. Bangalore: Sarada Ranganathan Endowment for Library Science, 1965.
- [536] A. Rauber. DELOS and the Future of Digital Libraries. *D-Lib Magazine*, 10(10). <http://www.dlib.org/dlib/october04/10contents.html> [last visited 2007, March 23], October 2004.
- [537] U. Ravindranathan. Prototyping digital libraries handling heterogeneous data sources - an ETANA-DL case study. Master's thesis, Virginia Tech CS Department, April 2004.
- [538] U. Ravindranathan, R. Shen, M. A. Gonçalves, W. Fan, E. A. Fox, and F. Flanagan. Prototyping digital libraries handling heterogeneous data sources - the ETANA-DL case study. In *Proc. 8th European Conf. Research and Advanced Technology for Digital Libraries, ECDL*, number 3232 in LNCS, pages 186–197, Bath, UK, Sept. 2004. Springer-Verlag.
- [539] U. Ravindranathan, R. Shen, M. A. Gonçalves, W. Fan, E. A. Fox, and J. W. Flanagan. ETANA-DL: managing complex information applications – an archaeology digital library. In *JCDL '04: Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 414–414, New York, NY, USA, 2004. ACM Press.
- [540] M. Recker and B. Palmer. Using resources across educational digital libraries. In *JCDL '06: Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 240–241, New York, NY, USA, 2006. ACM.
- [541] R. Reddy and I. Wladawsky-Berger. Digital Libraries: Universal Access to Human Knowledge - A Report to the President. President's Information Technology Advisory Committee (PITAC), Panel on Digital Libraries. <http://www.itrd.gov/pubs/pitac/pitac-dl-9feb01.pdf>, 2001.
- [542] T. C. Redman. *Data Quality – Management and Technology*. Bantam Books, New York, 1992.
- [543] U. o. O. Refugee Studies Centre. Forced Migration Online. <http://www.forcedmigration.org/>, October 2011.

- [544] D. Rehberger, M. Fegan, and M. Kornbluh. Reevaluating access and preservation through secondary repositories: Needs, promises, and challenges. In *Research and Advanced Technology for Digital Libraries, LNCS 4172*, pages 39–50. Springer, 2006.
- [545] S. E. Robertson. The probability ranking principle in IR. *J. Documentation*, 33:294–304, 1977.
- [546] K. Rodden, W. Basalaj, D. Sinclair, and K. Wood. Does Organization by Similarity Assist Image Browsing? In *ACM Conference on Human Factors in Computing Systems*, volume 3, pages 190–197, 2001.
- [547] P. Rödig, U. M. Borghoff, J. Scheffczyk, and L. Schmitz. Preservation of digital publications: an oais extension and implementation. In *Proceedings of the 1st ACM Symposium on Document Engineering*, pages 131–139, Grenoble, France, 2003.
- [548] M. Romanello, F. Boschetti, and G. Crane. Citations in the digital library of classics: extracting canonical references by using conditional random fields. In *Proceedings of the 2009 Workshop on Text and Citation Analysis for Scholarly Digital Libraries*, pages 80–87. Association for Computational Linguistics, 2009.
- [549] M. B. Rosson. Integrating development of task and object models. *Communications of the ACM*, 42(1):49–56, 1999.
- [550] M. B. Rosson and J. M. Carroll. Object-oriented design from user scenarios. In *Proc. of ACM CHI 96 Conf. on Human Factors in Computing Systems*, pages 342–343, 1996.
- [551] J. Rothenberg. *Avoiding Technological Quicksand: Finding a Viable Technical Foundation for Digital Preservation*. CLIR, Washington, D.C., 1999.
- [552] J. Rothenberg. *Using Emulation to Preserve Digital Documents*. Koninklijke Bibliotheek, The Netherlands, 2000.
- [553] M. Rowan, P. Gregor, D. Sloan, and P. Booth. Evaluating web resources for disability access. In *Fourth Annual ACM Conference on Assistive Technologies*, pages 80–84, Arlington, Virginia, 2000. ACM.
- [554] S. Roy, Y. Wan, and A. Saberi. A flexible algorithm for sensor network partitioning and self-partitioning problems. In *Algorithmic Aspects of Wireless Sensor Networks*, volume 4240 of *Lecture Notes in Computer Science*, pages 152–163. Springer Berlin / Heidelberg, 2006.
- [555] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra. A Power Tool in Interactive Content-Based Image Retrieval. *IEEE Tran. on Circuits and Systems for Video Technology*, 8(5):644–655, 1998.

## 472 B. GLOSSARY

- [556] P. Ruijgrov and M. Slabbertje. Requirements for Management & Storage to support complex objects & ORE in DSpace. In *Knowledge Exchange Group - Research paper - Enhanced E-theses Project Deliverable 9.1*, available at <http://igitur-archive.library.uu.nl/DARLIN/2010-0526-200239/UUindex.html>, last accessed on 05/05/11, 2009.
- [557] M. E. Ruiz and P. Srinivasan. Hierarchical text categorization using neural networks. *Information Retrieval*, 5(1):87–118, January 2002.
- [558] S. Rumsey and B. O’Steen. Oai-ore, preserv2 and digital preservation, ariadne, 6, available at <http://www.ariadne.ac.uk/issue57/rumsey-osteen/>, 2008.
- [559] N. Ryan. Managing complexity: Archaeological information systems past, present and future. In *Proc. British Association Annual Festival of Science, University of Birmingham, 8-13 Sept.*, 1996.
- [560] T. S. S. Chang and A. Puri. Overview of the mpeg-7 standard. *Circuits and Systems for Video Technology, IEEE Transactions*, 11(6):688–695, 2001.
- [561] M. Safar, C. Shahabi, and X. Sun. Image Retrieval by Shape: A Comparative Study. In *IEEE International Conference on Multimedia and Expo (I)*, pages 141–144, 2000.
- [562] N. Sager. *Natural Language Information Processing: A computer grammar of English and its applications*. Addison-Wesley, 1981.
- [563] P. Saha. Tensor Scale: A Local Morphometric Parameter With Applications to Computer Vision and Image Processing. Technical Report 306, Medical Image Processing Group, Department of Radiology, University of Pennsylvania, September 2003.
- [564] K. Saidis and A. Delis. Integrating Multi-dimensional Information Spaces. *Second Workshop on Very Large Digital Libraries, in conjunction with the 13th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2009)*, 07 2009.
- [565] G. Salton. *Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer*. Addison-Wesley, Boston, Massachusetts, USA, 1989.
- [566] G. Salton, E. A. Fox, and H. Wu. Extended Boolean information retrieval. *Communications of the ACM*, 26(11):1022–1036, November 1983.
- [567] G. Salton and M. E. Lesk. The smart automatic document retrieval system - an illustration. *Communications of the ACM*, 8(6):391–398, 1965.
- [568] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

- [569] M. Sanderson and Y. Han. Search words and geography. In *Proceedings of the 4th ACM workshop on Geographical information retrieval*, pages 13–14, Lisbon, Portugal, 2007.
- [570] R. Sandhu and P. Samarati. Access control: principle and practice. *Communications Magazine, IEEE*, 32(9):40 –48, sep 1994.
- [571] S. Sanett. The Cost to Preserve Authentic Electronic Records in Perpetuity: Comparing Costs across Cost Models and Cost Frameworks. *RLG DigiNews*, 7(4), August 2003. [http://www.rlg.org/preserv/diginews/v7\\_n4\\_feature2.html](http://www.rlg.org/preserv/diginews/v7_n4_feature2.html).
- [572] A. Santanchè and C. B. Medeiros. Fluid web and digital content components: from a document-centric perspective to a content-centric perspective. In *Proceedings of the XX Brazilian Symposium on Databases*, pages 10–24, 2005.
- [573] A. Santanchè and C. B. Medeiros. A Component Model and Infrastructure for a Fluid Web. *IEEE Transactions on Knowledge and Data Engineering*, 19(2):324–341, February 2007.
- [574] A. Santanchè, C. B. Medeiros, and G. Z. Pastorello Jr. User-author centered multimedia building blocks. *Multimedia Systems*, 12(4):403–421, March 2007.
- [575] S. Santini, A. Gupta, and R. Jain. Emergent Semantics through Interaction in Image Databases. *IEEE Transactions on Knowledge and Data Engineering*, 13(3):337–351, May/June 2001.
- [576] D. Santos and M. S. Chaves. The place of place in geographical ir. In *Proceedings of the 3rd ACM Workshop On Geographic Information*, pages 5–8, Seattle, Aug. 2006. Department of Geography, University of Zurich.
- [577] T. Saracevic. Relevance: a review and a framework for thinking on the notion in information science. *Journal of the American Society for Information Science*, 26:321–343, 1975.
- [578] T. Saracevic. Relevance reconsidered. In *Information science: Integration in perspectives. Proceedings of the Second Conference on Conceptions of Library and Information Science. Copenhagen (Denmark)*, pages 201–218, 1996.
- [579] T. Saracevic. Digital library evaluation: Toward evolution of concepts. *Library Trends*, 49(2):350–369, 2000.
- [580] T. Saracevic. Evaluation of digital libraries: an overview. Technical report, Rutgers University, 2004.
- [581] T. Saracevic and L. Covi. Challenges for digital library evaluation. In *Proceedings of the 63rd Annual Meeting of the American Society for Information Science*, volume 37, pages 341–350, 2000.

## 474 B. GLOSSARY

- [582] S. Sarkar and A. Dong. Community detection in graphs using singular value decomposition. *Phys. Rev. E*, 83:046114, Apr 2011.
- [583] K. Sayood. *Introduction to Data Compression*. Morgan Kaufmann Publishers, 2929 Campus Drive, Suite 260, San Mateo, CA 94403, USA, 1996.
- [584] R. C. Schank. *Tell Me a Story: Narrative and Intelligence*. Northwestern University Press, 1995. Introduction-Morson, Gary Saul.
- [585] J. Schloen. Archaeological Data Models and Web Publication Using XML. *Computers and the Humanities*, 35(2):123–152, 2001.
- [586] S. Schockaert, M. De Cock, and E. E. Kerre. Location approximation for local search services using natural language hints. *International Journal of Geographical Information Science*, 22(3):315, 2008.
- [587] D. Schonberg and D. Kirovski. Fingerprinting and forensic analysis of multimedia. In *Proceedings of the 12th annual ACM international conference on Multimedia*, MULTIMEDIA '04, pages 788–795, New York, NY, USA, 2004. ACM.
- [588] M. F. Schwartz and D. C. M. Wood. Discovering shared interests using graph analysis. *Communications of the ACM*, 36(8):78–89, Aug. 1993.
- [589] E. Sciascio, F. Donini, and M. Mongiello. Spatial layout representation for query-by-sketch content-based image retrieval. *Pattern Recognition*, 23(13):1599–1612, 2002.
- [590] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys (CSUR)*, 34(1):1–47, 2002.
- [591] P. Serdyukov, V. Murdock, and R. van Zwol. Placing flickr photos on a map. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, pages 484–491, New York, NY, USA, 2009. ACM.
- [592] D. E. Shackelford, J. B. Smith, and F. D. Smith. The architecture and implementation of a distributed hypermedia storage system. In *Proc. of the 5th Conf. on Hypertext*, pages 1–13, Seattle, Washington, Nov. 1993.
- [593] D. Shah and T. Zaman. Community detection in networks: The leader-follower algorithm. *CoRR*, 2010.
- [594] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, July, Oct. 1948.
- [595] R. Shen. *Applying the 5S Framework to Integrating Digital Libraries*. Ph.d. dissertation, Virginia Tech CS Department, Blacksburg, Virginia, 2006. URL - <http://scholar.lib.vt.edu/theses/available/etd-04212006-135018/>.

- [596] R. Shen, M. A. Gonçalves, W. Fan, and E. A. Fox. Requirements gathering and modeling of domain-specific digital libraries with the 5s framework: An archaeological case study with etana. In *Proc. European Conference on Digital Libraries, ECDL 2005, Vienna, Sept. 18-23*. Springer, 2005.
- [597] R. Shen, N. S. Vemuri, W. Fan, R. da S. Torres, and E. A. Fox. Exploring digital libraries: integrating browsing, searching, and visualization. In *JCDL '06: Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*, pages 1–10, New York, NY, USA, 2006. ACM.
- [598] R. Shen, N. S. Vemuri, W. Fan, and E. A. Fox. Integration of complex archeology digital libraries: An etana-dl experience. *Information Systems*, 33(7-8):699–723, 2008.
- [599] R. Shen, N. S. Vemuri, V. Vijayaraghavan, W. Fan, and E. A. Fox. Etanaviz: A visual user interface to archaeological digital libraries. Technical Report TR-05-14, Computer Science, Virginia Tech, October 2005.
- [600] S. B. Shum, E. Motta, and J. Domingue. Scholonto: An ontology-based digital library server for research documents and discourse. *International Journal on Digital Libraries*, 3(3):237–248, 2000. Sept.
- [601] M. Singhal and N. Shivaratri. *Advanced Concepts in Operating Systems: Distributed, Database, and Multiprocessor Operating Systems*. McGraw-Hill, New York, 1994.
- [602] H. G. Small. Co-citation in the scientific literature: A new measure of the relationship between two documents. *Journal of the American Society for Information Science*, 24(4):265–269, July-Aug. 1973.
- [603] J. R. Smith and S. F. Chang. VisualSEEk: A fully automated content-based image query system. In *Proceedings of the ACM Multimedia*, pages 87–98, Boston, MA, November 1996.
- [604] T. Smith, G. F. Killeen, N. Maire, A. Ross, L. Molineaux, F. Tediosi, G. Hutton, J. Utzinger, K. Dietz, and M. Tanner. Mathematical modeling of the impact of malaria vaccines on the clinical epidemiology and natural history of plasmodium falciparum malaria: Overview. In *Am. J. Trop. Med. Hyg.*, volume 75, pages 1–10, 2006.
- [605] V. Srinivasan, M. Magdy, and E. Fox. Enhanced Browsing System for Electronic Theses and Dissertations. In *Proceeding of 14th International Symposium on Electronic Theses and Dissertations*. NDLTD, 2011.
- [606] W. Stallings. *Cryptography and Network Security*. Pearson Prentice Hall, 4 edition, 2006.
- [607] D. Stan and I. K. Sethi. eID: a System for Exploration of Image Databases. *Information Processing and Management*, 39(3):335–365, 2003.

## 476 B. GLOSSARY

- [608] T. Staples and R. Wayland. Virginia dons fedora: A prototype for a digital object repository. *D-Lib Magazine*, 6(7/8), 2000. July/August.
- [609] T. Staples, R. Wayland, and S. Payette. The fedora project - an open-source digital object repository management system. *D-Lib Magazine*, 9(4), Apr. 2003.
- [610] R. Stehling, M. Nascimento, and A. Falcão. A Compact and Efficient Image Retrieval Approach Based on Border/Interior Pixel Classification. In *Proceedings of the 11th ACM International Conference on Information and Knowledge Management*, pages 102–109, McLean, Virginia, USA, November 2002.
- [611] M. A. Stricker and M. Orengo. Similarity of Color Images. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 381–392, 1995.
- [612] R. Studer, R. R. Benjamins, and D. Fensel. Knowledge Engineering: Principles and Methods. *Data Knowledge Engineering*, 25(1-2):161–197, 1998.
- [613] G. Stumme and A. Maedche. Ontology merging for federated ontologies for the semantic web. In *Proc. Intl. Workshop on Foundations of Models for Information Integration, Viterbo, Italy, Sept. 16-18, 2001*. LNAI, Springer 2001, Sept. 2001.
- [614] H. Suleman. *Open Digital Libraries*. PhD thesis, Virginia Tech CS Department, Blacksburg, Virginia, 2002. <http://scholar.lib.vt.edu/theses/available/etd-11222002-155624/>.
- [615] H. Suleman and E. A. Fox. A Framework for Building Open Digital Libraries. *D-Lib Magazine*, 7(12), 2001.
- [616] H. Suleman and E. A. Fox. Designing protocols in support of digital library componentization. In *Proceedings of the 6th European Conference on Research and Advanced Technology for Digital Libraries (ECDL2002)*, LNCS 2458, pages 75–84. Springer, Rome, Italy, 2002.
- [617] H. Suleman, E. A. Fox, and M. Abrams. Building quality into a digital library. In *Proceedings of the Fifth ACM Conference on Digital Libraries: DL '00, June 2-7, 2000, San Antonio, TX*. ACM Press, New York, 2000. June 4-7, 2000.
- [618] T. Sumner and M. Marlino. Digital libraries and educational practice: a case for new models. In *Digital Libraries, 2004. Proceedings of the 2004 Joint ACM/IEEE Conference on*, pages 170 – 178, june 2004.
- [619] A. Sutcliffe. A technique combination approach to requirements engineering. In *Proc. of the 3rd Int. Symp. on Requirements Engineering*, pages 65–77, Annapolis, 1997. IEEE.
- [620] A. G. Sutcliffe, N. A. M. Maiden, S. Minocha, and D. Manuel. Supporting scenario-based requirements engineering. *IEEE Trans. on Soft. Engineering*, 24(12):1072–1088, 1998.

- [621] M. Swain and D. Ballard. Color Indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.
- [622] H. Tamura, S. Mori, and T. Yamawaki. Textural Features Corresponding to Visual Perceptron. *IEEE Transactions on Systems, Man and Cybernetics*, 8(6):460–473, 1978.
- [623] R. Tansley, M. Bass, M. Branschofsky, G. Carpenter, G. McClellan, and D. Stuve. Dspace system documentation, available at <http://www.dspace.org/index.php?/architecture/technology/system-docs/index.html>, 2009.
- [624] R. Tansley, M. Bass, D. Stuve, M. Branschofsky, D. Chudnov, G. McClellan, and M. Smith. DSpace: An institutional digital repository system. In *Proc. of the 3rd Joint Conference on Digital Libraries*, pages 87–97, Houston, Texas, 2003.
- [625] R. Tansley, M. Bass, D. Stuve, M. Branschofsky, D. Chudnov, G. McClellan, and M. Smith. The dspace institutional digital repository system: current functionality. In *JCDL '03: Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries*, pages 87–97, Washington, DC, USA, 2003. IEEE Computer Society.
- [626] R. Tansley and S. Harnad. Eprints.org software for creating institutional and individual open archives. *D-Lib Magazine*, 6(10), October 2000.
- [627] M. Taube and et al. Storage and retrieval of information by means of the association of ideas. *American Documentation*, 6(1):1–18, 1955.
- [628] R. S. Taylor. Question-negotiation and information seeking in libraries. *College and Research Libraries*, 29:178–194, 1968.
- [629] TEI. Text encoding initiative: Tei: Yesterday’s information tomorrow, 2005.
- [630] H. S. Thompson and D. Connolly. W3c xml pointer, xml base and xml linking, 1997-2003. Description of work done by the W3C XML Linking Working Group.
- [631] Q. Tian, B. Moghaddam, and T. S. Huang. Display Optimization for Image Browsing. In *Proceedings of the 2nd International Workshop on Multimedia Databases and Image Communications*, Amalfi, Italy, Sep 2001.
- [632] H. R. Tibbo. Archival perspectives on the emerging digital library. *CACM*, 44(5), 2001.
- [633] W. Tolone, G.-J. Ahn, T. Pai, and S.-P. Hong. Access control in collaborative systems. *ACM Comput. Surv.*, 37:29–41, March 2005.
- [634] S. Tong and E. Y. Chang. Support vector machine active learning for image retrieval. In *Proceedings of 9th ACM international conference on multimedia*, pages 107–118, NYC, 2001. ACM.

## 478 B. GLOSSARY

- [635] S. Toral, M. Martinez-Torres, F. Barrero, and F. Cortes. An Empirical Study of the Driving Forces behind Online Communities. *Internet Research*, 19(4):378–392, 2009.
- [636] R. Torres, C. Medeiros, M. Gonçalves, and E. Fox. A digital library framework for biodiversity information systems. *International Journal on Digital Libraries*, 6(1):3–17, 2006.
- [637] R. S. Torres and A. X. F. ao. Content-Based Image Retrieval: Theory and Applications. *Revista de Informática Teórica e Aplicada*, 13(2):161–185, 2006.
- [638] R. S. Torres, C. G. Silva, C. B. Medeiros, and H. V. Rocha. Visual Structures for Image Browsing. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management*, pages 167–174, New Orleans, LA, USA, November 2003.
- [639] C. Traina, J. M. Figueiredo, and A. J. M. Traina. Image domain formalization for content-based image retrieval. In *Proceedings of the 2005 ACM symposium on applied computing*, pages 604–609, 2005.
- [640] C. Traina, B. Seeger, C. Faloutsos, and A. Traina. Fast Indexing and Visualization of Metric Datasets Using Slim-Trees. *IEEE Transactions on Knowledge and Data Engineering*, 14(2):244–60, March/April 2002.
- [641] G. Tsakonas, S. Kapidakis, and C. Papatheodorou. Evaluation of user interaction in digital libraries. *DELOS Workshop on the Evaluation of Digital Libraries*, 2004.
- [642] F. A. Twaroch, P. D. Smart, and C. B. Jones. Mining the web to detect place names. In *Proceeding of the 2nd international workshop on Geographic information retrieval*, pages 43–44, Napa Valley, California, USA, 2008. ACM.
- [643] P. Tyrvainen. concepts and a design for fair use and privacy in drm. *D-Lib Magazine*, 11, 2005. <http://www.dlib.org/dlib/february05/tyrvainen/02tyrvainen.html>.
- [644] S. Urs and E. A. Fox. Indo-us workshop on open digital libraries and interoperability, 23-25 june 2003, arlington, usa, June 2003.
- [645] U.S. National Archives and Records Administration. National archives.
- [646] O. Van Laere, S. Schockaert, and B. Dhoedt. Finding locations of flickr resources using language models and similarity search. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, ICMR '11, pages 48:1–48:8, New York, NY, USA, 2011. ACM.
- [647] R. C. Veltkamp and M. Tanase. Content-based image retrieval systems: A survey. Technical Report TR UU-CS-2000-34, Department of Computing Science, Utrecht University, October 2002.

- [648] Ø. Vestavik. Geographic information retrieval: An overview. In *Internal Doctoral Conference*, page 7, IDI, NTNU, Norway, 2003.
- [649] B. C. Vickery. Faceted classification schemes. In *Rutgers Series for the Intellectual Organization of Information – Volume 5*. Rutgers University Press, New Brunswick, NJ, USA, 1965.
- [650] S. Vitaladevuni and R. Basri. Co-clustering of image segments using convex optimization applied to em neuronal reconstruction. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2203–2210, 2010.
- [651] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17:395–416, 2007.
- [652] E. M. Voorhees. Evaluation by highly relevant documents. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 74–82, 2001.
- [653] K. Vu, K. A. Hua, and W. Tavanapong. Image Retrieval Based on Regions of Interest. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):1045–1049, July/August 2003.
- [654] W3C. *Resource Description Framework (RDF) Model and Syntax Specification*, 1998. <http://www.w3.org/TR/WD-rdf-syntax/>.
- [655] w3schools. Semantic Web Tutorial. <http://www.w3schools.com/semweb/default.asp/>, 2011. [Online; accessed 26-September-2011].
- [656] Y. Wand and R. Y. Wang. Anchoring data quality dimensions in ontological foundations. *Communications of the ACM*, 39(11):86–95, Nov. 1996.
- [657] F. Wang, C. Rabsch, P. Kling, P. Liu, and J. Pearson. Web-based collaborative information integration for scientific research. In *IEEE 23rd International Conference on Data Engineering*, pages 1232–1241, 2007.
- [658] J. Z. Wang and Y. Du. Scalable integrated region-based image retrieval using irm and statistical clustering. In *Proceedings of the 1st ACM/IEEE-CS joint conference on digital libraries*, pages 268–277, 2001.
- [659] W. Wang and R. Rada. Structured hypertext with domain semantics. *ACM Transactions on Information Systems*, 16(4):372–412, October 1998.
- [660] Y. Wang, F. Makedon, J. Ford, L. Shen, and D. Goldin. Generating fuzzy semantic metadata describing spatial relations from images using the r-histogram. In *Proceedings of the 4th ACM/IEEE-CS joint conference on digital libraries*, pages 202–211, 2004.

## 480 B. GLOSSARY

- [661] S. Warner. Exposing and harvesting metadata using the oai metadata harvesting protocol: A tutorial. *HEP Libraries Webzine*, (4), June 2001 2001.
- [662] D. Waters and J. Garrett. *Preserving Digital Information: Report of the Task Force on Archiving of Digital Information*. CLIR, Washington, D.C., 1996.
- [663] D. J. Waters. What are digital libraries? *CLIR Issues*, (4), July/August 1998.
- [664] A. Waugh, R. Wilkinson, B. Hills, and J. Dell'oro. Preserving digital information forever. In *DL'00: Proceedings of the 5th ACM International Conference on Digital Libraries*, pages 175–184, San Antonio, Texas, 2000.
- [665] S. Weibel. The state of the dublin core metadata initiative: April 1999. *D-Lib Magazine*, 5(4), 1999.
- [666] S. L. Weibel and T. Koch. The dublin core metadata initiative: Mission, current activities, and future directions, available at <http://www.dlib.org/dlib/december00/weibel/12weibel.html>. *D-Lib Magazine*, 6(12), December 2000.
- [667] H. G. Wells. World Brain: The Idea of a Permanent World Encyclopaedia. In *Encyclopédie Française*. Anatole de Monzie and Lucien Febvre, 1937.
- [668] D. West, M. Finnegan, R. Lane, and D. Kysar. *Analysis of Faunal Remains Recovered from Tell Nimrin, Dead Sea Valley, Jordan, final report*. Case Western Reserve University, 1996.
- [669] G. Wiederhold. Digital libraries, value, and productivity. *Communications of the ACM*, 38(4):85–96, 1995.
- [670] D. A. Wiley. Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. *Learning Technology*, 2830(435):1–35, 2000.
- [671] R. Wilkison and M. Fuller. Integration of information retrieval and hypertext via structure. In *Information Retrieval and Hypertext*, pages 257–271. Kluwer Academic Publishers, 1996.
- [672] K. Williams and H. Suleman. A Survey of Digital Library Aggregation Services. In *Scholarship at Penn Libraries*, available at: [http://works.bepress.com/martha\\_brogan/10/](http://works.bepress.com/martha_brogan/10/), 2003.
- [673] M. Williams. What makes RABBIT run? *International Journal of Man-Machine Studies*, 21(4):333–352, 1984.
- [674] G. Winskel. *The Formal Semantics of Programming Languages: An Introduction*. Foundations of Computing series. MIT Press, Cambridge, MA, USA, Feb. 1993.

- [675] M. Winslett, N. Ching, V. Jones, and I. Slepchin. Assuring security and privacy for digital library transactions on the web: client and server security policies. In *Research and Technology Advances in Digital Libraries, 1997. ADL '97. Proceedings., IEEE International Forum on*, pages 140–151, may 1997.
- [676] I. Witten, D. Bainbridge, and S. J. Boddie. Greenstone Open-Source Digital Library Software. *D-Lib Magazine*, 7(10), 2000.
- [677] I. H. Witten and D. Bainbridge. *How to Build a Digital Library*. Morgan Kaufmann Publishers, San Francisco (CA), USA, 2003.
- [678] I. H. Witten and D. Bainbridge. A retrospective look at Greenstone: Lessons from the first decade, Research Commons at The University of Waikato. In *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries, Vancouver, BC, Canada June 18 - 23, 2007*, pages 147–156. ACM, 2007.
- [679] I. H. Witten, D. Bainbridge, and S. Boddie. Greenstone: Open-source DL software. *Communications of the ACM*, 44(5):47, 2001.
- [680] I. H. Witten, D. Bainbridge, and D. M. Nichols. *How to Build a Digital Library, 2nd ed.* Morgan Kaufmann Publishers, San Francisco (CA), USA, 2009.
- [681] I. H. Witten, D. Bainbridge, G. Paynter, and S. Boddie. The Greenstone plugin architecture. In *JCDL '02: Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pages 285–286, New York, NY, USA, 2002. ACM.
- [682] I. H. Witten, D. Bainbridge, R. Tansley, C. Huang, and K. Don. StoneD: A bridge between Greenstone and DSpace. *D-Lib Magazine*, 11(9), September 2005.
- [683] I. H. Witten, R. J. McNab, S. J. Boddie, and D. Bainbridge. Greenstone: A comprehensive open-source digital library software system. In *Proc. of the 5th ACM Int. Conf. on Digital Libraries*, pages 113–121, San Antonio, TX, June 2-7, 2000.
- [684] I. H. Witten, A. Moffat, and T. C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann Publishers, 2nd edition, 1999.
- [685] K. Wittenburg and E. Sigman. Integration of browsing, searching, and filtering in an applet for web information access. In *Proceedings of CHI*, pages 293–294, 1997.
- [686] Z. Xu, X. Xu, and V. Tresp. A hybrid relevancefeedback approach to text retrieval. In *In Proceedings of the 25th European Conference on Information Retrieval Research, Lecture Notes in Computer Science*, pages 281–293. Springer-Verlag, 2003.

## 482 B. GLOSSARY

- [687] G. R. Xue, D. Xing, Q. Yang, and Y. Yu. Deep classification in large-scale text hierarchies. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 619–626, New York, NY, USA, 2008. ACM.
- [688] S. Yang, A. L. Kavanaugh, N. P. Kozievitch, L. T. Li, V. Srinivasan, S. D. Sheetz, T. Whalen, D. Shoemaker, R. da Silva Torres, and E. A. Fox. CTRnet DL for disaster information services. In *JCDL*, pages 437–438, 2011.
- [689] Y. Yang, J. Zhang, and B. Kisiel. A scalability analysis of classifiers in text categorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '03, pages 96–103, New York, NY, USA, 2003. ACM.
- [690] B. Yu and G. Cai. A query-aware document ranking method for geographic information retrieval. In *Proceedings of the 4th ACM workshop on Geographical information retrieval*, pages 49–54, Lisbon, Portugal, 2007. ACM.
- [691] J. Yu and X. Fan. Metadata Extraction from Chinese Research Papers Based on Conditional Random Fields. In *Fuzzy Systems and Knowledge Discovery, 2007. FSKD 2007. Fourth International Conference on*, volume 1, pages 497–501. IEEE, 2007.
- [692] B. Zhang, M. A. Gonçalves, and E. A. Fox. An OAI-Based Filtering Service for CITIDEL from NDLTD. In *Proc. of the 6th International Conference on Asian Digital Libraries, ICADL 2003*, pages 590–601, Kuala Lumpur, Malaysia, December 8-12, 2003.
- [693] D. Zhang and G. Lu. Review of Shape Representation and Description. *Pattern Recognition*, 37(1):1–19, Jan 2004.
- [694] J. Zhao. *Making Digital Libraries Flexible, Scalable, and Reliable: Reengineering the MARIAN System in JAVA*. Master of science thesis, Virginia Tech, 1999.
- [695] J. Zhao, M.-Y. Kan, and Y. L. Theng. Math information retrieval: user requirements and prototype implementation. In *Proceedings of the 8th ACM/IEEE-CS Joint Conference on Digital libraries*, pages 187–196. ACM, 2008.
- [696] X. S. Zhou and T. S. Huang. Relevance feedback in image retrieval: A comprehensive review. *Multimedia Systems*, 8:536–544, 2003.
- [697] B. Zhu, M. Ramsey, and H. Chen. Creating a Large-Scale Content-Based Airphoto Image Digital Library. *IEEE Transactions on Image Processing*, 9(1):163–167, January 2000.
- [698] Q. Zhu. *5SGraph: A Modeling Tool for Digital Libraries*. Masters thesis, Virginia Tech, 2002.

- [699] Q. Zhu, M. A. Gonçalves, R. Shen, L. Cassel, and E. A. Fox. Visual semantic modeling of digital libraries. In *Proc. 7th European Conference on Digital Libraries (ECDL 2003), 17-22 August, Trondheim, Norway, Springer LNCS 2769*, pages 325–337. Springer, 2004.
- [700] N. Ziviani, E. S. de Moura, G. Navarro, and R. Baeza-Yates. Compression: A key for next-generation text retrieval systems. *IEEE Computer*, 33(11):37–44, Nov. 2000.
- [701] J. Zou, D. Le, and G. Thoma. Locating and parsing bibliographic references in HTML medical articles. *International journal on document analysis and recognition*, 13(2):107–119, 2010.
- [702] A. Zubizarreta, P. d. l. Fuente, J. M. Cantera, M. Arias, J. Cabrero, G. García, C. Llamas, and J. Vegas. Extracting geographic context from the web: GeoReferencing in MyMoSe. In *ECIR '09: Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, volume 5478/2009 of *LNCS*, pages 554–561, Toulouse, France, Apr. 2009.

## Editor's Biography

### EDWARD A. FOX

**Edward A. Fox** grew up on Long Island, New York. He attended the Massachusetts Institute of Technology (MIT), receiving a B.S. in 1972 in Electrical Engineering, through the Computer Science option. His undergraduate adviser was JCR Licklider. His thesis adviser was Michael Kessler. At MIT he founded the ACM Student Chapter and the Student Information Processing Board, receiving the William Stewart Award.

From 1971–1972 he worked as Data Processing Instructor at the Florence Darlington Technical College. From 1972–1978 he was Data Processing Manager at Vulcraft, a Division of NUCOR Corporation, also in Florence, SC. In the fall of 1978 he began his graduate studies at Cornell University in Ithaca, NY. His adviser was Gerard Salton. He received an M.S. in Computer Science in 1981 and a Ph.D. in 1983. From the summer of 1982 through the spring of 1983 he served as Manager of Information Systems at the International Institute of Tropical Agriculture, Ibadan, Nigeria. From the fall of 1983 through the present he has been on the faculty of the Department of Computer Science at Virginia Tech (also called VPI&SU or Virginia Polytechnic Institute and State University). In 1988 he was given tenure and promoted to the rank of Associate Professor. In 1995 he was promoted to Professor.

Dr. Fox has been a member of ACM since 1968. He was vice chairman of ACM SIGIR 1987–1991. Then he was chair 1991–1995. During that period, he helped launch the new ACM SIG on Multimedia. He served as a member of the ACM Publications Board 1988–1992 and as Editor-in-chief of ACM Press Database and Electronic Products 1988–1991, during which time he helped conceive and launch the ACM Digital Library. He served 2000–2006 as a founder and Co-editor-in-chief of the ACM Journal of Education Resources In Computing (JERIC), which led to the ACM Transactions on Education. Over the period 2004–2008 he served as Chairman of the IEEE-CS Technical Committee on Digital Libraries, and continues to serve on its Executive Committee. Dr. Fox served 1995–2008 as Editor of the Morgan Kaufmann Publishers, Inc. Series on Multimedia Information and Systems. Dr. Fox has been a member of Sigma Xi since the 1970s and a member of Upsilon Pi Epsilon since 1998.

In 1987 Dr. Fox began to explore the idea of all students shifting to electronic theses and dissertations (ETDs), and has worked in this area ever since. He led the establishment of the Networked Digital Library of Theses and Dissertations (operating informally starting

## **EDITOR'S BIOGRAPHY 485**

in 1995, incorporated in May 2003). He serves as founder and Executive Director of NDLTD. He won its 1st Annual NDLTD Leadership Award in May 2004.

Dr. Fox has been involved in a wide variety of professional service activities. He has chaired scores of conferences or workshops, and served on hundreds of program or conference committees. At present he serves on ten editorial boards, and is a member of the board of directors of the Computing Research Association (CRA; he is co-chair of its membership committee, as well as a member of CRA-E, its education committee). He also chairs the steering committee of the ACM/IEEE-CS Joint Conference on Digital Libraries.

Dr. Fox has been (co)PI on over 110 research and development projects. In addition to his courses at Virginia Tech, Dr. Fox has taught over 77 tutorials in more than 28 countries. His publications and presentations include: 13 books, 97 journal/magazine articles, 48 book chapters, 168 refereed (+40 other) conference/workshop papers, 55 posters, 66 keynote/banquet/international invited/distinguished speaker presentations, 38 demonstrations, and over 300 additional presentations. His research and teaching has been on digital libraries, information storage and retrieval, hypertext/hypermedia/multimedia, computing education, computational linguistics, and sub-areas of artificial intelligence.

# Index

- 5S, ii, 3, 8  
access, 1, 4, 5, 11  
archive, 27  
bioinformatics, 321  
bucket, 141  
catalog, 9, 27  
CC2001, 2  
collection, 27  
completeness, 104  
complex object, 137  
compression, 137  
conceptual representation, 58  
conformance, 104  
content-based image retrieval (CBIR), 261  
curriculum, 2  
DCC, 141  
DELOS Reference Model, 28  
digital libraries, ii  
digital object, 9, 89  
e-science, 4, 321  
education, 305  
Envision, 15  
evaluation, 88  
event, 9  
exploration, 54  
extraction, 395  
facets, 5  
federation, 28  
geospatial, 348  
GIR, 353  
harvest, 165  
hybrid, 11  
index, 1, 9  
indicator, 89  
information life cycle, 1  
institutional repository, 27  
integration, 160  
Kessler, v  
Licklider, v  
measure, 89  
Memex, 14  
metadata, 3, 4, 9, 11  
NSF, vi  
OAI-ORE, 141  
online community, 284  
ontology, 218  
Open Archives Initiative, 11  
personalization, 284  
pertinence, 93  
preservation, 7  
preserve, 1  
question bank, 315

- relevance, 89
- repository, 9, 27, 165
- Salton, v
- scenario, 9
- security, 3, 373
- segmentation, 403
- service, 9, 28
- simulation, 321
- social network, 284
- subdocument, 183
- system, 28
- union, 161
- Vannevar Bush, 14
- video, 9