Digital Library Technologies

Complex Objects, Annotation, Ontologies, Classification, Extraction, and Security

Synthesis Lectures on Information Concepts, Retrieval, and Services

Editor
Gary Marchionini, *University of North Carolina, Chapel Hill*

Digital Library Technologies: Complex Objects, Annotation, Ontologies, Classification, Extraction, and Security
Edward A. Fox, Ricardo da Silva Torres
2014

Digital Library Applications: CBIR, Education, Social Networks, eScience/Simulation, and GIS
Edward A. Fox, Jonathan P. Leidig
2014

Information and Human Values
Kenneth R. Fleischmann
November 2013

Multiculturalism and Information and Communication Technology
Pnina Fichman and Madelyn R. Sanfilippo
October 2013

The Future of Personal Information Management, Part II: Transforming Technologies to Manage Our Information
William Jones
September 2013

Information Retrieval Models: Foundations and Relationships
Thomas Roelleke
July 2013

Key Issues Regarding Digital Libraries: Evaluation and Integration
Rao Shen, Marcos Andre Goncalves, Edward A. Fox
February 2013

Visual Information Retrieval using Java and LIRE
Mathias Lux, Oge Marques
January 2013

On the Efficient Determination of Most Near Neighbors: Horseshoes, Hand Grenades, Web Search and Other Situations When Close is Close Enough
Mark S. Manasse
November 2012

The Answer Machine
Susan E. Feldman
September 2012

Theoretical Foundations for Digital Libraries: The 5S (Societies, Scenarios, Spaces, Structures, Streams) Approach
Edward A. Fox, Marcos André Gonçalves, Rao Shen
July 2012

The Future of Personal Information Management, Part I: Our Information, Always and Forever
William Jones
March 2012

Search User Interface Design
Max L. Wilson
November 2011

Information Retrieval Evaluation
Donna Harman
May 2011

Knowledge Management (KM) Processes in Organizations: Theoretical Foundations and Practice
Claire R. McInerney, Michael E. D. Koenig
January 2011

Search-Based Applications: At the Confluence of Search and Database Technologies
Gregory Grefenstette, Laura Wilber
2010

Information Concepts: From Books to Cyberspace Identities
Gary Marchionini
2010

Estimating the Query Difficulty for Information Retrieval
David Carmel, Elad Yom-Tov
2010

iRODS Primer: Integrated Rule-Oriented Data System
Arcot Rajasekar, Reagan Moore, Chien-Yi Hou, Christopher A. Lee, Richard Marciano, Antoine de Torcy, Michael Wan, Wayne Schroeder, Sheau-Yen Chen, Lucas Gilbert, Paul Tooby, Bing Zhu
2010

Collaborative Web Search: Who, What, Where, When, and Why
Meredith Ringel Morris, Jaime Teevan
2009

Multimedia Information Retrieval
Stefan Rüger
2009

Online Multiplayer Games
William Sims Bainbridge
2009

Information Architecture: The Design and Integration of Information Spaces
Wei Ding, Xia Lin
2009

Reading and Writing the Electronic Book
Catherine C. Marshall
2009

Hypermedia Genes: An Evolutionary Perspective on Concepts, Models, and Architectures
Nuno M. Guimarães, Luís M. Carrico
2009

Understanding User-Web Interactions via Web Analytics
Bernard J. (Jim) Jansen
2009

XML Retrieval
Mounia Lalmas
2009

Faceted Search
Daniel Tunkelang
2009

Introduction to Webometrics: Quantitative Web Research for the Social Sciences
Michael Thelwall
2009

Exploratory Search: Beyond the Query-Response Paradigm
Ryen W. White, Resa A. Roth
2009

New Concepts in Digital Reference
R. David Lankes
2009

Automated Metadata in Multimedia Information Systems: Creation, Refinement, Use in Surrogates, and Evaluation
Michael G. Christel
2009

Digital Library Technologies: Complex Objects, Annotation, Ontologies, Classification, Extraction, and Security

Edward A. Fox and Ricardo da Silva Torres

www.morganclaypool.com

ISBN: 9781627050302 print
ISBN: 9781627050319 ebook

DOI: 10.2200/S00566ED1V01Y201401ICR033

A Publication in the Morgan & Claypool Publishers series
SYNTHESIS LECTURES ON INFORMATION CONCEPTS, RETRIEVAL, AND SERVICES
Series ISSN: 1947-945X print 1947-9468 ebook

Lecture #33
Series Editor: Gary Marchionini, University of North Carolina, Chapel Hill

First Edition

10 9 8 7 6 5 4 3 2 1

Digital Library Technologies

Complex Objects, Annotation, Ontologies, Classification, Extraction, and Security

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

Ricardo da Silva Torres
Institute of Computing, University of Campinas, Campinas, SP, Brazil

Chapter Authors:
Pranav Angara, Lois M. Delcambre, Noha Elsherbiny, Nádia P. Kozievitch, Mohamed Magdy Gharib Farag, Uma Murthy, Sung Hee Park, Venkat Srinivasan, Ricardo da Silva Torres, and Seungwon Yang

SYNTHESIS LECTURES ON INFORMATION CONCEPTS, RETRIEVAL, AND SERVICES #33

MORGAN & CLAYPOOL PUBLISHERS

ABSTRACT

Digital libraries (DLs) have introduced new technologies, as well as leveraging, enhancing, and integrating related technologies, since the early 1990s. These efforts have been enriched through a formal approach, e.g., the 5S (Societies, Scenarios, Spaces, Structures, Streams) framework, which is discussed in two earlier volumes in this series. This volume should help advance work not only in DLs, but also in the WWW and other information systems.

Drawing upon four (Kozievitch, Murthy, Park, Yang) completed and three (Elsherbiny, Farag, Srinivasan) in-process dissertations, as well as the efforts of collaborating researchers and scores of related publications, presentations, tutorials, and reports, this book should advance the DL field with regard to at least six key technologies. By integrating surveys of the state-of-the-art, new research, connections with formalization, case studies, and exercises/projects, this book can serve as a computing or information science textbook. It can support studies in cyber-security, document management, hypertext/hypermedia, IR, knowledge management, LIS, multimedia, and machine learning.

Chapter 1, with a case study on fingerprint collections, focuses on complex (composite, compound) objects, connecting DL and related work on buckets, DCC, and OAI-ORE. Chapter 2, discussing annotations, as in hypertext/hypermedia, emphasizes parts of documents, including images as well as text, managing superimposed information. The SuperIDR system, and prototype efforts with Flickr, should motivate further development and standardization related to annotation, which would benefit all DL and WWW users. Chapter 3, on ontologies, explains how they help with browsing, query expansion, focused crawling, and classification. This chapter connects DLs with the Semantic Web, and uses CTRnet as an example. Chapter 4, on (hierarchical) classification, leverages LIS theory, as well as machine learning, and is important for DLs as well as the WWW. Chapter 5, on extraction from text, covers document segmentation, as well as how to construct a database from heterogeneous collections of references (from ETDs); i.e., converting strings to canonical forms. Chapter 6 surveys the security approaches used in information systems, and explains how those approaches can apply to digital libraries which are not fully open.

Given this rich content, those interested in DLs will be able to find solutions to key problems, using the right technologies and methods. We hope this book will help show how formal approaches can enhance the development of suitable technologies and how they can be better integrated with DLs and other information systems.

KEYWORDS

5S, annotation, CINET, classification, complex objects, Crisis/Tragedy/ Recovery network (CTRnet), digital libraries (DLs), ETDs, fingerprints, Flickr, formalization, network science, OAI-ORE, ontologies, security, subdocuments, SuperIDR, superimposed information, text extraction

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

Contents

List of Figures

List of Tables

Preface

Because of the importance of digital libraries, we integrated, organized, and condensed our related findings and publications into a single volume version of this book series, ultimately over 600 pages in length, that was successfully used in a semester-long class in 2011, as well as field tested at different universities. To make it easier for others to address their need for a digital library textbook, we have re-organized the original book into four parts, to cover: introduction and theoretical foundations, key issues, technologies/extensions, and applications. We are confident that this third book, and the others in the series, address digital library-related needs in many computer science, information science, and library science (e.g., LIS) courses, as well as the requirements of researchers, developers, and practitioners.

The main reason for our confidence is that our *5S* (Societies, Scenarios, Spaces, Structures, Streams) framework has broad descriptive power. 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., databases, metadata, ontologies, XML), and data Stream management systems.

The first book, *Theoretical Foundations for Digital Libraries,* the essential opening to the four book series, has three main parts. Chapter 1 is the key to 5S, providing a theoretical foundation for the field of digital libraries in a gentle, intuitive, and easy-to-apply manner. Chapter 2 explains how 5S can be applied to digital libraries in two ways. First, it covers the most important services of digital libraries: browsing, searching, discovery, and visualization. Second, it demonstrates how 5S helps with the design, implementation, and evaluation of an integrated digital library (ETANA-DL, for archaeology). The third part of book 1, made up of five appendices, demonstrates how 5S enables a formal treatment of digital libraries. It is freely accessible online, at https://sites.google.com/a/morganclaypool.com/dlibrary/.

Book 1 Appendix A gives a small set of definitions that cover the mathematical preliminaries underlying our work. Appendix B builds on that set to define each of the five Ss, and then uses them to define what we consider a minimal digital library. Thus, we allow people asking "Is X a digital library?" to answer that question definitively. Appendix C moves from a minimalist perspective to show how 5S can be used in a real, interesting, and complex application domain: archaeology. Appendix D builds upon all the definitions in Appendices A-C, to describe some key results of using 5S. This includes lemmas, proofs, and 5SSuite (software based on 5S). Finally, Appendix E,

the Glossary, explains key terminology. Concluding book 1 is an extensive bibliography and a helpful Index.

The second book in the series, *Key Issues Regarding Digital Libraries: Evaluation and Integration,* discusses key issues in the digital library field: evaluation and integration. It covers the Information Life Cycle, metrics, and software to help evaluate digital libraries. It uses both archaeology and electronic theses and dissertations to provide additional context, since addressing quality in highly distributed digital libraries is particularly challenging.

The following two books of this series are further elaborations of the 5S framework, as well as a comprehensive overview of related work on digital libraries.

This book, third in the series, describes six case studies of extensions beyond a minimal digital library. Its chapters cover: Complex Objects, Annotation, Ontologies, Classification, Text Extraction, and Security. *Regarding Complex Objects:* While many digital libraries focus on digital objects and/or metadata objects, with support for complex objects, they could easily be extended to 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. *Regarding Annotation:* This builds upon work on superimposed information, closely related to hypertext, hypermedia, and subdocuments. A case study covers the management of fish images. *Regarding Ontologies:* We address this key area of knowledge management, also integral to the Semantic Web. As a context, we consider our Crisis, Tragedy, and Recovery Network. That is quite broad, and involves interesting ontology development problems. *Regarding Classification:* We cover this core area of information retrieval and machine learning, as well as Library and Information Science (LIS). The context is electronic theses and dissertations (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. *Regarding Text Extraction:* Our coverage also is 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. *Regarding 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.

The final book, *Digital Library Applications: CBIR, Education, Social Networks, eScience/ Simulation, and GIS,* fourth in the series, focuses on digital library applications from a 5S perspective. *Regarding CBIR:* We move into the multimedia field, focusing on Content-based Image Retrieval (CBIR)—making use, for context, of the previously discussed work on fish images and CTRnet. *Regarding Education:* We describe systems for collecting, sharing, and providing access to educational resources, namely the AlgoViz and Ensemble systems. 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. *Regarding Social Networks:* We address very popular current issues, on the Societies side, namely Social Networks and Personalization. *Regarding e-Science/Simulation:* There has only been a limited adaptation and extension of digital libraries to this important domain. 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 support much broader theories and investigations, we cover (using the SimDL and CINET projects as context) 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. *Regarding Geospatial Information (GIS):* Many GIS-related technologies are now readily available in cell phones, cameras, and GPS systems. Our coverage (that uses the CTRnet project as context) connects that with metadata, images, and maps.

How can computer scientists connect with all this? Although some of the early curricular guidelines for computing 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. Fortunately, Virginia Tech has had graduate courses on information retrieval since the early 1970s and a senior course on "Multimedia, Hypertext, and Information Access" since the early 1990s. Now, there are offerings at many universities on multimedia, or with titles including keywords like "Web" or "search". Perhaps parts of this book series 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 book series may fit nicely into capstone courses in information science or information systems. Further, our handling of "information" goes well beyond the narrow view associated with electrical engineering or even computer science; we

connect content representations with context and application, across a range of human endeavors, and with semantics, pragmatics, and knowledge.

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, all four books should fit nicely into library science programs. While they could fit into theory or capstone courses, they also might serve well in introductory courses, if the more formal parts are skipped. On the other hand, they could be distributed across the program. Thus, the first book might work well early in a library school program, the second book could fit midway in the program, and the last two books might be covered in specialized courses that connect with technologies or applications. Further, those studying archival science might find the entire series to be of interest, though some topics like preservation are not covered in detail.

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, and deploy even better digital libraries. There should be less time "reinventing the wheel." Perhaps this will stimulate 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. Accordingly, we hope you can leverage this work to advance practices as well as 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 book series interesting. Given the rich content, we trust that those interested in digital libraries, or in related systems, will find this book to be intellectually satisfying, illuminating, and helpful. We hope the full series will help move digital libraries forward into a science as well as a practice. We hope too that this four book series will broadly address the needs of the next generation of digital librarians. Please share with us and others what ways you found these books to be useful and helpful!

Edward A. Fox, Editor
Blacksburg, Virginia
February 2014

Acknowledgments

As lead in this effort, 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 families, as well as my father and many other relatives. Similarly, on behalf of my co-editor and each of the chapter co-authors, I thank all of their families.

Since this book is the third in a series of four books, and draws some definitions and other elements from content that either was presented in the first or second book, or will appear in the fourth book, it is important to acknowledge the contributions of all of the other co-authors from the full series: Monika Akbar, Pranav Angara, Yinlin Chen, Lois M. Delcambre, Noha Elsherbiny, Alexandre X. Falcão, Eric Fouh, Nádia P. Kozievitch, Spencer Lee, Jonathan Leidig, Lin Tzy Li, Mohamed Magdy Gharib Farag, Uma Murthy, Sung Hee Park, Venkat Srinivasan, Ricardo da Silva Torres, and Seungwon Yang. Special thanks go to Uma Murthy for helping with the bibliography and to Monika Akbar, Pranav Angara, and Shashwat Dave for assistance with technical aspects of book production. Further, Shashwat Dave assisted with the glossary, found in the first book of the series as well as online; it is useful in this book, too.

Teachers and mentors deserve a special note of thanks. My interest in research was stimulated and guided by J.C.R. Licklider, my undergraduate advisor, author of *Libraries of the Future*,[1] who, when at ARPA, funded the start of the Internet. Michael Kessler, who introduced the concept of bibliographic coupling, was my B.S. 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 students who have collaborated in these matters, including: Scott Britell, Pavel Calado, Yuxin Chen, Kiran Chitturi, Fernando Das Neves, Shahrooz Feizabadi, Robert France, S.M.Shamimul Hasan, Nithiwat Kampanya, Rohit Kelapure, S.H. Kim, Neill Kipp, Aaron Krowne, Sunshin Lee, Bing Liu, Ming Luo, Paul Mather, Sai Tulasi Neppali, Unni. Ravindranathan, W. Ryan Richardson, Nathan Short, Ohm Sornil, Hussein Suleman, Wensi Xi, Baoping Zhang, and Qinwei Zhu.

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

Further, we thank faculty and staff, at a variety of universities and other institutions, who have collaborated, including: A. Lynn Abbott, Felipe Andrade, Robert Beck, Keith Bisset, Paul Bogen II, Peter Brusilovsky, Lillian Cassel, Donatella Castelli, Vinod Chachra, Hsinchun Chen, Debra Dudley, Roger Ehrich, Hicham Elmongui, Joanne Eustis, Tiago Falcão, Weiguo Fan, James Flanagan, James French, Richard Furuta, Dan Garcia, C. Lee Giles, Martin Halbert, Kevin Hall, Eric Hallerman, Riham Hassan, Eberhard Hilf, Gregory Hislop, Michael Hsiao, Haowei Hsieh, John Impagliazzo, Filip Jagodzinski, Andrea Kavanaugh, Douglas Knight, Deborah Knox, Alberto Laender, Carl Lagoze, Madhav Marathe, Gary Marchionini, Susan Marion, Gail McMillan, Claudia Medeiros, Barbara Moreira, Henning Mortveit, Sanghee Oh, Donald Orth, Jeffrey Pomerantz, Manuel Perez Quinones, Naren Ramakrishnan, Evandro Ramos, Mohammed Samaka, Steven Sheetz, Frank Shipman, Donald Shoemaker, Layne Watson, and Barbara Wildemuth.

Clearly, with regard to this volume, my special thanks go to my co-author, Ricardo da Silva Torres. He played a key role in the unfolding of the theory, practice, systems, and usability of what is described herein. Regarding earlier work on 5S, Marcos André Gonçalves helped launch our formal framework, and Rao Shen extended that effort, as can be seen in the first two books of the series.

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 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-9752190, DUE-9752408, DUE-0121679, DUE-0121741, DUE-0136690, DUE-0333531, DUE-0333601, DUE-0435059, DUE-0532825, DUE-0840719, DUE-1141209, IIS-9905026, IIS-9986089, IIS-0002935, IIS-0080748, IIS-0086227, IIS-0090153, IIS-0122201, IIS-0307867, IIS-0325579, IIS-0535057, IIS-0736055, IIS-0910183, IIS-0916733, IIS-1319578, ITR-0325579, OCI-0904844, OCI-1032677, and 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. Students in our VT-MENA program in Egypt have been supported through that program.

We thank corporate and institutional sponsors, including Adobe, AOL, CNI, Google, IBM, Microsoft, NASA, NCR, OCLC, SOLINET, SUN, SURA, UNESCO, U.S. Department of Education (FIPSE), and 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.

Finally, we acknowledge the support of the Qatar National Research Fund for Project No. NPRP 4-029-1-007, running 2012-2015.

CHAPTER 1

Complex Objects

Nádia P. Kozievitch and Ricardo da Silva Torres

Abstract: In order to reuse, integrate, and unify different resources from a common perspective, Complex Objects (COs) have emerged to support digital library (DL) initiatives from both theoretical and practical perspectives. From the theoretical perspective, the use of COs facilitates aggregation abstraction. From the implementation point of view, the use of COs helps developers to manage heterogeneous resources and their components. On the other hand, DL applications still lack support for mechanisms to process and manage COs in services such as reference creation, annotation, content-based searches, harvesting, and component organization. This chapter extends the discussions in the previous books of this series regarding: (i) the formalization of complex objects based on the 5S framework; (ii) the study of three widely used technologies for managing COs; and (iii) a case study discussion on how to handle complex image objects in DL applications. The concepts addressed in this chapter can be used to classify, compare, and highlight the differences among CO-related components, technologies, and applications, impacting DL researchers, designers, and developers.

1.1 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 data collections.

Users involved in the creation and management of, 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 (DLs).

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 elements combined together into a unique logical object [114, 154 155]. Among the several advantages of structuring together individual components, we can cite their reuse in

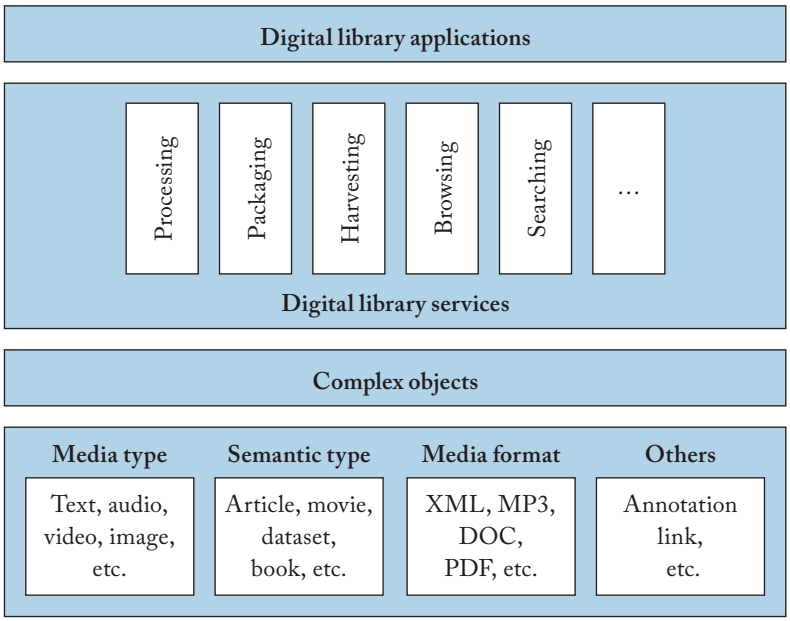

FIGURE 1.1: Architecture for a CO-based digital library. (Adapted from [105])

multiple representations with flexibility, or the exploration of complex inter-object relationships (e.g., semantic linkages) [65].

Figure 1.1 shows the architecture of 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 DL applications. Yet, these applications have faced some challenging issues [11, 187]: (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 their components.

Most of the existing solutions dealing with these issues have focused only on textual data. With the creation of large image and video collections motivated by novel technologies for data acquisition and sharing, new challenges have emerged. In particular, if we consider image and video data, significant research efforts have been spent in the development of appropriate systems to efficiently manage multimedia collections [219]. In many cases, however, those initiatives are not enough to deal with COs that integrate both textual and visual components.

In fact, in spite of all the advances, there is a lack of consensus on the precise formalization involved in reusing, integrating, unifying, managing, and supporting CO-related tasks in diverse

application domains. 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 [72]. On the other hand, formalized concepts can be used to classify, compare, and highlight the differences among components, technologies, and applications, thus aiding DL researchers, designers, and developers.

In this chapter, we address the formal definitions and descriptions for COs by exploiting concepts of the 5S framework. 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 in digital libraries.

1.2 COMPLEX OBJECTS

This section introduces the definition of a CO and compares widely used technologies for implementing CO-related services.

1.2.1 DEFINITIONS

Some authors name the integration of resources into a single digital object as *Aggregation* [226], a *Component-Based Object* [195, 196], a *Complex Object* [154], or a *Compound Object* [10]. We adopt the same definition of structuring digital objects present in [10]: 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) in a preferred format. The compound 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. [108], COs are single entities that are composed of multiple digital objects, each of which is an entity in and of itself. Cheung et al. [31] 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. 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 [114]. Santanchè, on the other hand, used the idea of COs in the field of software reuse and exchange [195, 196]. Like the script concept [198] or the frame concept [135], the components in a CO are supposed to have the same behavior, respect the same rules, or represent the same concept.

1.2.2 TECHNOLOGIES FOR HANDLING COMPLEX OBJECTS

Several complex object (CO) formats arise from different communities [107, 124, 154, 155] and can be used under different domains [99]. In scientific computing, standards arise, such as Network Common Data Form (NetCDF) [160], Hierarchical Data Format (HDF) [82], and Extensible File System (ELFS) [92]. 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) [22] or Metadata Encoding and Transmission Standard (METS) [45]. One example for including digital object formats is the Moving Picture Experts Group—21 Digital Item Declaration Language (MPEG-21 DIDL) [15].

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 [49]; see their comparative study of the IMS Content Package (IMS CP) [205] and Reusable Asset Specification (RAS) [204].

Newer standards have emerged, like SQL Multimedia and Application Packages (SQL/MM) [132]. 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) [29] is an International Organization for Standardization (ISO) reference model, with a particular focus on digital information, 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) [111] is a framework for archives (e.g., institutional repositories) containing digital content (i.e., a type of DL). The OAI technical infrastructure, specified in the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) [167, 212], 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 [119] 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 DL object, where file elements may be grouped within fileGrp elements, to provide for subdividing the files by object version. A $\langle fileGrp \rangle$ structure is used to comprise a single electronic version of the DL object. $\langle FContent \rangle$ 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 DL objects,
- recording the names and locations of the files that comprise those objects, and
- recording associated metadata.

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

SCORM [2] 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 manifest (*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 webpage can use (e.g., Flash, JavaScript, frames, and images).

MPEG-21 [22] aims to define an open framework for multimedia applications, to support, for example, declaration (and identification), digital rights management, and adaptation. 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 1.1 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.

1.2.3 COMPARISON OF CO-RELATED TECHNOLOGIES (DCC, BUCKETS, OAI-ORE)

Each of the CO technologies were created to address different problems, so DL developers will have to judge which technology best addresses existing requirements. From the several CO technologies available, three different approaches were chosen for a comparison. DCC was chosen for comparison

TABLE 1.1: 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

because it can be implemented in several languages, it supports the encapsulation of software, and it allows the reuse/composition of components. OAI-ORE is a widely used protocol for representing and describing aggregations for future reuse and exchange (metadata harvesting). Several applications have been developed lately taking advantage of the OAI-ORE standard. Finally, Buckets are used in the DL community, as an aggregation construct (allowing links to remote packages, networks, or databases) which can be archived and manipulated as a single object. For instance, OAI-ORE is a metadata harvesting approach and focuses on data integration, while Buckets and DCC have an operational orientation focusing on the repository level.

Digital Content Component (DCC)

Digital Content Component (DCC) [49, 175, 194, 195, 196] was proposed in 2006, as a generalization format for representing complex objects. The approach derives from an analysis and comparison of content packages, and Open Complex Digital Object (OCDO) and reuse standards [194].

A DCC is composed of four distinct subdivisions (see Figure 1.2):

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—a WSDL and OWL-S (semantics); and

metadata. metadata to describe version, functionality, applicability, and use restrictions—using OWL.

Buckets

Buckets [156, 157, 158] 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 with arbitrary network services. Buckets are based on standard World Wide Web (WWW)

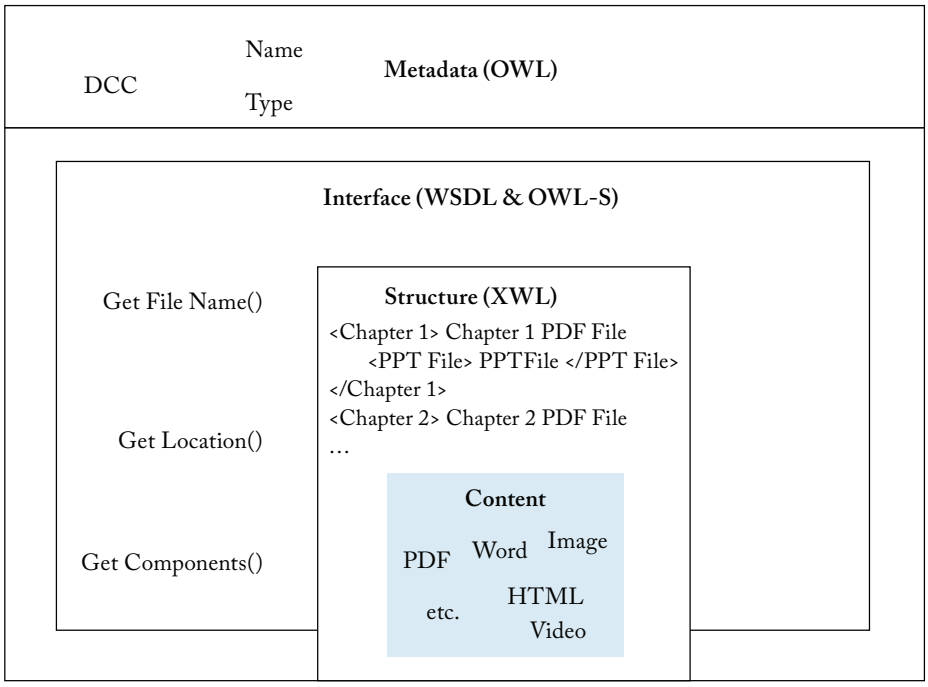

FIGURE 1.2: Digital Content Component (DCC) representation.

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.

Open Archives Initiative Protocol—Object Reuse and Exchange (OAI-ORE)

OAI-ORE [114, 124] aims to develop, identify, and profile extensible standards and protocols to allow repositories, agents, and services to interoperate in the context of use and reuse of COs. 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. Figure 1.3 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 defines an abstract data model [112] conformant with the architecture of the Web, essentially consisting of:

URIs. (Uniform Resource Identifiers) for identifying objects;

resources. which are items of interest;

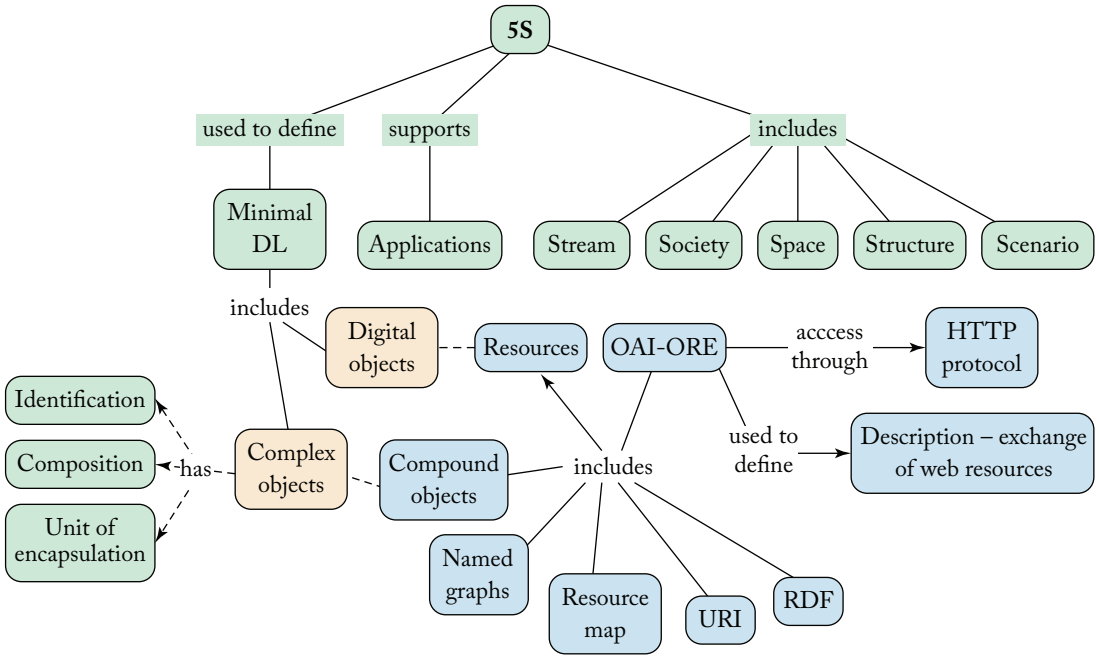

FIGURE 1.3: Matching the main concepts of the 5S framework and OAI-ORE. (Adapted from [102])

standard protocols. such as HTTP, that enable access to the data;

named graphs. for encapsulating information into a CO. The encoded description (serialization) of the named graph is called a resource map; and

proxy. a virtual resource acting as a proxy for an aggregated resource in the context of a certain aggregation. Its use is optional.

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

For highlighting their differences even more, we selected other parameters related to the identification, component organization, structure, boundary, and manipulation of COs (see Table 1.2):

TABLE 1.2: Basic CO concepts from DCC, buckets, and OAI-ORE perspectives

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
CO Organization	Parts accessed through relative URI, other DCCs	Packages and nested Buckets	Resource map and aggregations
Advantages	Ontology, interface, encapsulate executable	Pointer to remote package, network or database, log	Move repository, used as standard between content different systems
How to 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 compressed for archival or transport	Description allows easy transport and reuse

(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 issues.

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

and processes in DCCs; metadata and content in Buckets; and descriptions of aggregations in OAI-ORE.

In DCC, the internal CO format is divided into content, structure, interface, and metadata. In Buckets, the internal CO format is divided into 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 component reuse. They present different advantages, but all include characteristics of digital preservation (e.g., encapsulate content and software, allow directories to be compressed for archiving, and include description facilities for exchange and reuse).

1.3 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 [68] specified, for example, an ontology for the encapsulation of digital resources and bibliographic records;

Granularity. Fonseca et al. in [60] cited vertical navigation, where accessing a class immediately above or below implies a change of level of detail;

Standards for aggregations. In the context of the DELOS project, a DL Manifesto [26] has been proposed, in which Candela 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 [26], also the priority explored was 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 [174] 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 [126] suggested different access roles for the different parts, as suggested in the authentication and authorization service;

Reuse and preservation. Rehberger et al. in [185] examined the role that secondary repositories can play in the preservation and access of digital historical and cultural heritage materials;

Others. Tracking of provenance [138], timelines [77], etc.

COs also have been used in preservation and harvesting [131, 189], to combine current objects to create new ones [194], to combine services [105], or even for grouping information with respect to the same permissions or operations. Depending on the aggregation, different layers can be exposed, using different information granularity, or type of media, for example. Within applications for CO, we can mention LORE [68] and Escape [215].

1.4 FORMALIZATION

Formalizing complex objects facilitates the development, comparison, and evaluation of solutions based on distinct information resources; makes clear to users what a solution means; indicates how components are related; and helps users to evaluate the applicability of a solution. Furthermore, it allows us to leverage special-purpose techniques for combining, aggregating, and understanding the integration process. In this section, we introduce, having the 5S formal framework as foundation, concepts related to the minimum CO and a novel type of CO, named CIO (CIO), defined to encapsulate images.

Notation: Let DL_1 be a DL; let $\{do_1, do_2, \ldots, do_n\}$ 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.

1.4.1 COMPLEX OBJECT

Recall the 5S definition of a digital object (Def. MI B.18 of Appendix B, first book of this series). 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 (unique identifiers);
2. $SM = \{sm_1, sm_2, \ldots, sm_n\}$ is a set of streams;
3. $ST = \{st_1, st_2, \ldots, st_m\}$ is a set of structural metadata specifications;
4. $StructuredStreams = \{stsm_1, stsm_2, \ldots, stsm_p\}$ 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). *StructuredStreams* define the mapping of a structure to streams (for example, how chapters, sections, etc. are organized in a book).

Definition 1.1 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 $DO = \{do_1, do_2, \ldots, do_n\}$, and do_i is a digital object or another complex object; and $SM = \{sm_a, sm_b, \ldots, sm_z\}$ is a set of streams;
3. S is a structure that composes the complex object cdo into its parts in $SCDO$.

Note that the 5S definitions consider the object's metadata in a separate catalog. 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 CO is not specified. It can be seen as 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 [10], where lists represent multiple single files that were packed together, and graphs represent files that are related, creating networks of digital objects. If we consider files arranged in HTML5 [174], the S structure encodes 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.

Another type of structuring resource comprises the concept of a collection. The main difference is that a collection is a simple set of objects (Def. MI B.19 in Book 1), while the elements in a CO represent parts of a single concept and might have specific relations connecting them. In particular, consider the issue of Compound Scholarly Publications, explored through several organizations and projects (Europeana [48], SURF Foundation [214], and Eco4r [52]).

The definitions presented in this section can be used to formally describe aggregations and their several aspects available in the 5S framework. These definitions also could be used to construct and initialize applications, similar to initiatives presented in [197, 202, 233], or to devise novel concepts when looking for interoperability and compatibility among different technologies.

Definition 1.2 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 $SM = \{sm_a, sm_b, \ldots, sm_z\}$ 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 (Def. MI B.18 in Book 1) can be applied.

Definition 1.3 In particular, the *complex image object* (CIO) [101] is a CO with the following components: the digital image object, feature vector, and similarity scores (presented in Figure 1.4). If we consider the CO definition, the CIO has the structure $ico = (h, SCDO, S)$, where:

- h is a unique handle that identifies ico;

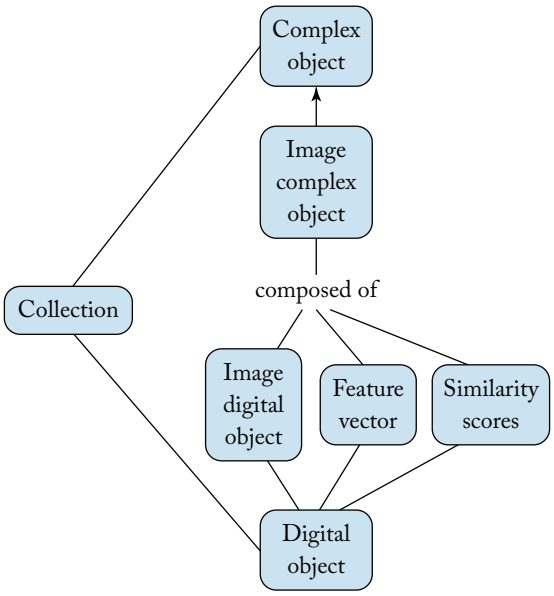

FIGURE 1.4: The CIO. (Adapted from [101])

- $SCDO = \{DO \cup SM\}$, where $DO = \{do_1, do_{21}, \ldots, do_{2k}, do_{31}, \ldots, do_{3k}\}$, where do_1 is an *image*, k is the number of descriptors, $do_{21}, \ldots, do_{2k}$ is a set of *feature vector digital objects*, and $do_{31}, \ldots, do_{3k}$ is a set of *StructuredFeatureVectors* (with the similarity measures, according to a specific descriptor k); and $SM = \{sm_a, sm_b, \ldots, sm_z\}$ is a set of streams;
- S is a structure that identifies how $do_1, do_{21}, \ldots, do_{2k}$, and $do_{31}, \ldots, do_{3k}$ are composed.

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

A *complex image object collection* $ImgCO$ is a tuple $(C, S_{imgdesc})$, where C is a collection (Def. MI B.19 in Book 1), and $S_{imgdesc}$ is a set of image descriptors. Function FV_{desc} defines how a feature vector was obtained, given a CIO $ico \in C$ and a image descriptor $\hat{D} \in S_{imgdesc}$.

1.5 CASE STUDY: FINGERPRINT DIGITAL LIBRARY

This section briefly describes a case study concerning the use of COs in the construction of a fingerprint digital library. For further details on other examples and services, the reader is referred to [98, 100, 101, 105]. A detailed description on concepts related to fingerprint analysis can be found in [19].

1.5.1 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, in a fingerprint digital library [104]. We offer this as an example of how database modeling approaches can be enriched with a theoretical handling of CO concepts. The goal is to use CO concepts to better support requirements analysis and the design and implementation of important database and/or DL applications.

Consider a fingerprint digital library which unifies four different digital libraries, from a complex object (CO) perspective.

- Those aware of law enforcement activities will know of the first type of DL (DL1), associated with databases of stored fingerprints.
- Another type relates to a project creating 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 National Institute of Justice (NIJ) funded research studies supporting experimentation with fingerprint image analysis techniques, quality measures, and matching methods (DL4).

The integration of these DLs faces several challenges: syntactic and semantic mismatches, service interoperability, transparency, etc.

In DL1, digital objects are used to identify a person. It manages large law enforcement databases that may have millions of individuals' sets of 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 systems is from the Federal Bureau of Investigation [56]. It has at least 66 million subjects in the criminal master file, along with more than 25 million civilian 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 later can be 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 and used parameter values. Examples include parameters that encode skin distortion and blurring effects. 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 different parameter values, for example, can synthetically generate about 10,000 images.

Through the integration, the DL unifies four different communities, allowing each one to see different perspectives, explore the system as a whole, or focus on a determined DL collection. In addition, users can take advantage of DL services (e.g., browsing and searching) to have a unified view of all collections.

1.5.2 INTEGRATION OF DIGITAL LIBRARIES

According to [191], 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. 1.5), embracing components from different domains and unifying them with a single concept.

Figure 1.6 presents the concept map of the main classes as a summary of the entity-relationship diagram [103]. Class Individual, for example, aggregates all the information from the 10 fingers,

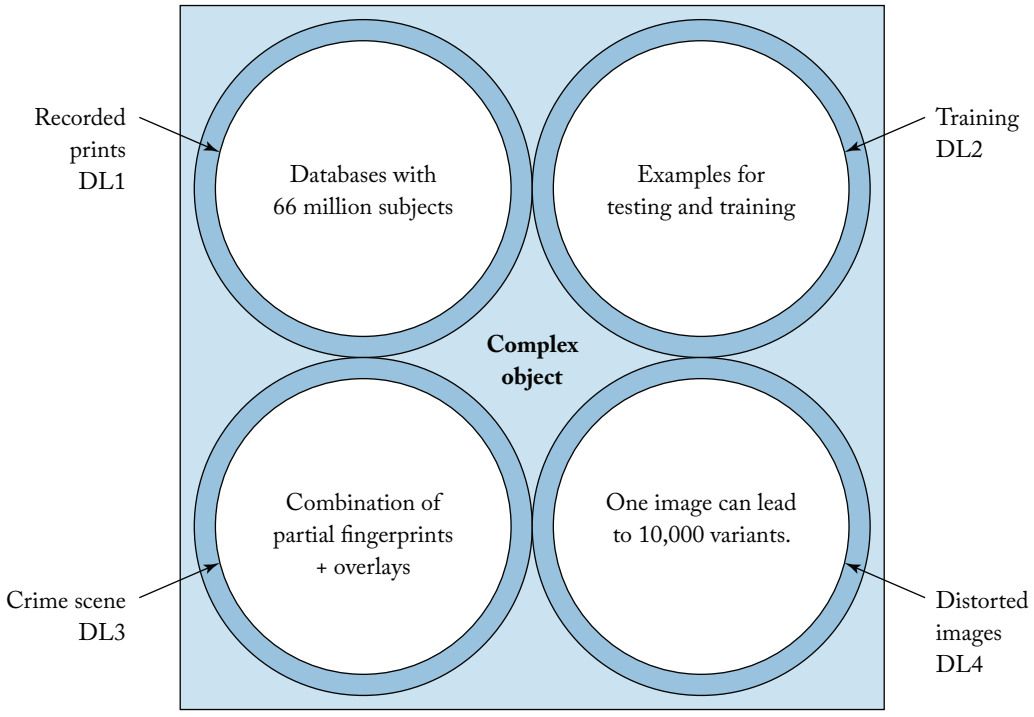

FIGURE 1.5: The integration of fingerprint digital libraries.

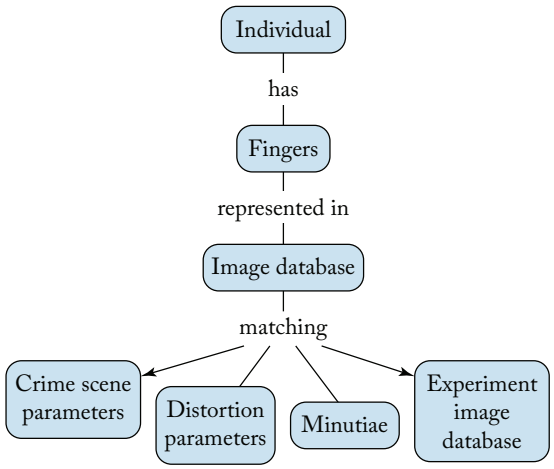

FIGURE 1.6: The main classes representing the fingerprint DL.

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 at the police station.

The integration of the four sub-systems is exemplified in Figure 1.7. Complex 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 are identified by CO1.A.1, CO1.B.1, CO1.C.1, and CO1.D.1, respectively. The CO1 structure could 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 1.7, we have CO1= $(h, SCDO, S)$ where:

1. h is a unique handle that represents CO1, and $h \in H$, where H is a set of universally unique handles (labels);
2. $SCDO = \{DO \cup SM\}$, where $DO = \{A.1, B.1, C.1, D.1\}$; and $SM = \{sm_a, sm_b, \ldots, sm_z\}$ is a set of streams;
3. S is structured by means of an XML file, aggregating the complex object cdo into its parts in $SCDO$.

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

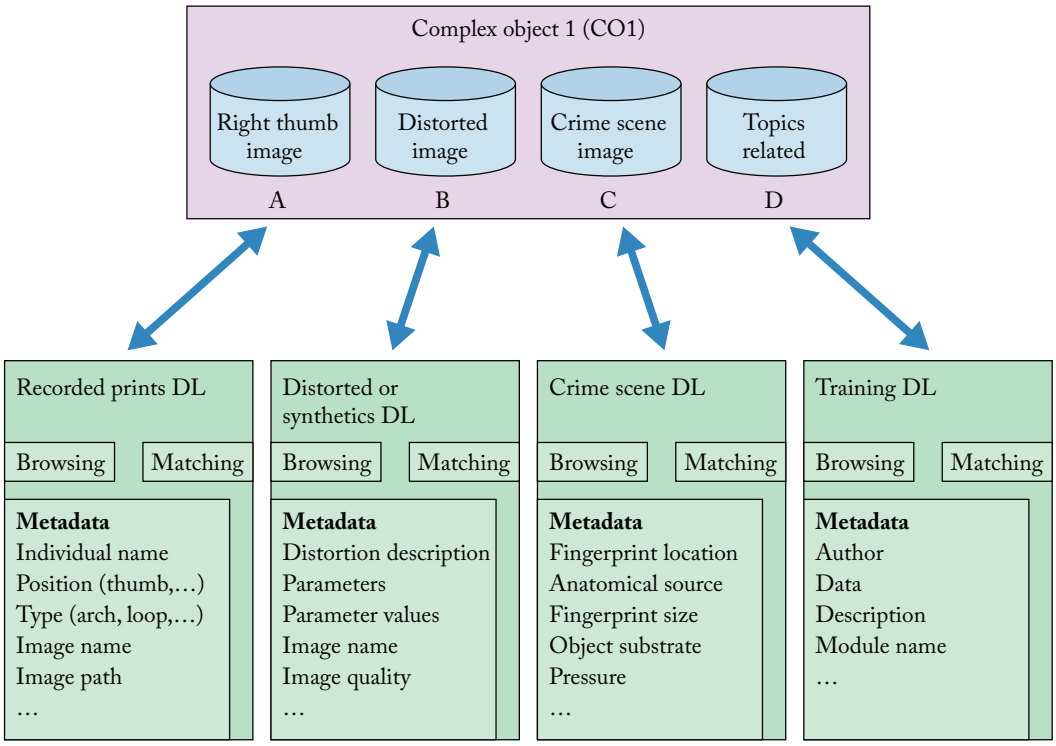

FIGURE 1.7: An example of complex object using four digital libraries [104]: (A) Recorded Prints, (B) Distorted Images, (C) Crime Scene Images, and (D) Training Material.

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

Examples of structures in a fingerprint DL include the information organization (such as Figure 1.6). Each person has 10 fingers, and each finger can produce different images depending on whether it is from a police station, a distortion, or from a crime scene. If they belong to the same finger, structures are used to represent this hierarchy.

Streams refer to the different types 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 under the 5S perspective on a project in an early development stage was important to highlight the amount of information and details needed to manage and aggregate. DCCs, for example, 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 to 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 [131]: (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.

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 [202].

In the case of complex 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 possible aggregation. For this, we used the 5S framework along with the CO technologies to analyze the integrated fingerprint DL, from the identity, components, structure, and boundary perspectives. Finally, we discussed how the components can be accessed later, along with their individual metadata.

1.5.3 IMPLEMENTATION

This section presents issues regarding the implementation of a fingerprint DL prototype that integrates the four digital libraries discussed in the previous section.

Considering the large size and types of variance of the fingerprint images, as well as the computational costs of fingerprint verification algorithms, for the prototype we used a pre-processing phase, using Content-Based Image Retrieval (CBIR) techniques (see Chapter 1 of Book 4 of this series). 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 effectiveness 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 about 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 their 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 definition of the fingerprint CO under the 5S perspective; (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 definition of the fingerprint CO under the 5S perspective played a key role in understanding 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 2, we defined that the aggregation would comprise the individual concept (as shown in Figure 1.6). The identification and relation of the compound parts were stored in a DBMS, matching images to respective fingerprint DL, metadata, image content descriptors, and similarity distances.

Phase 3

In phase 3, the integration of the CBIR process allowed a pre-categorization of the image, using comparisons based on texture features. For this, the Statistical Analysis of Structural Information (SASI) [28] descriptor was used. The CBIR processing of Figure 1.8 (fingerprint 11), for example, generates a feature vector, and the similarity distances to the other images in the collection. Figure 1.10 shows the ranking for Figure 1.8 (fingerprint 11) according to the texture comparison. The 10 top-down images are the most similar images compared to Figure 1.8 (fingerprint 11).

The CBIR processing of a second image (Figure 1.9—fingerprint 3) generates a second feature vector, and another set of similarity distances. Figure 1.11 presents the search results for Figure 1.9 (fingeprint 3) regarding the employed texture-based comparison.

Phase 4

In phase 4, a 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 composition and decomposition of components. The main characteristics

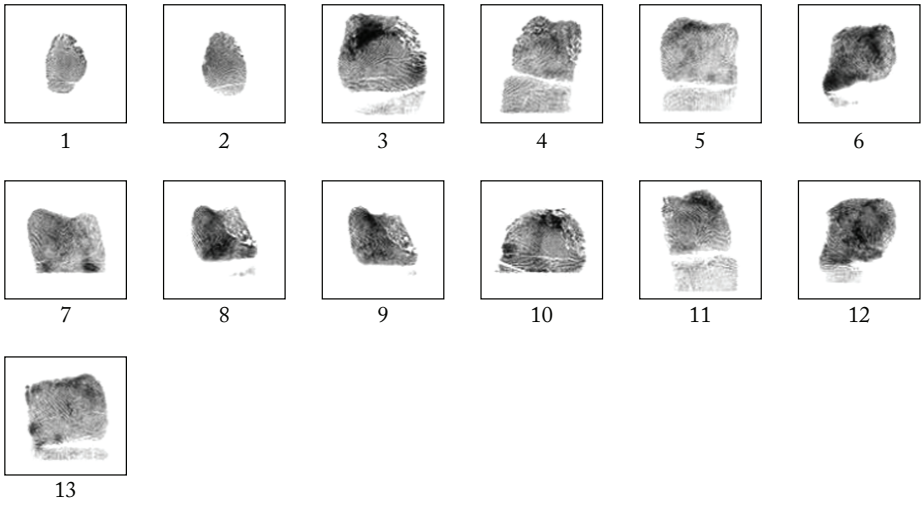

FIGURE 1.8: Samples of images from a recorded print DL from the police.

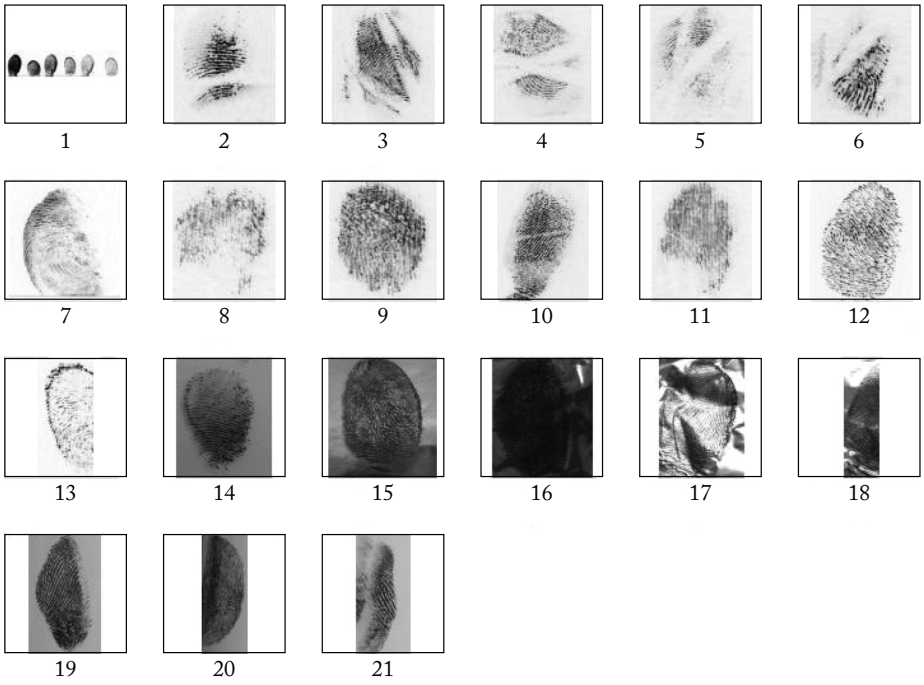

FIGURE 1.9: Samples of fingerprints from a DL which simulates a crime scene.

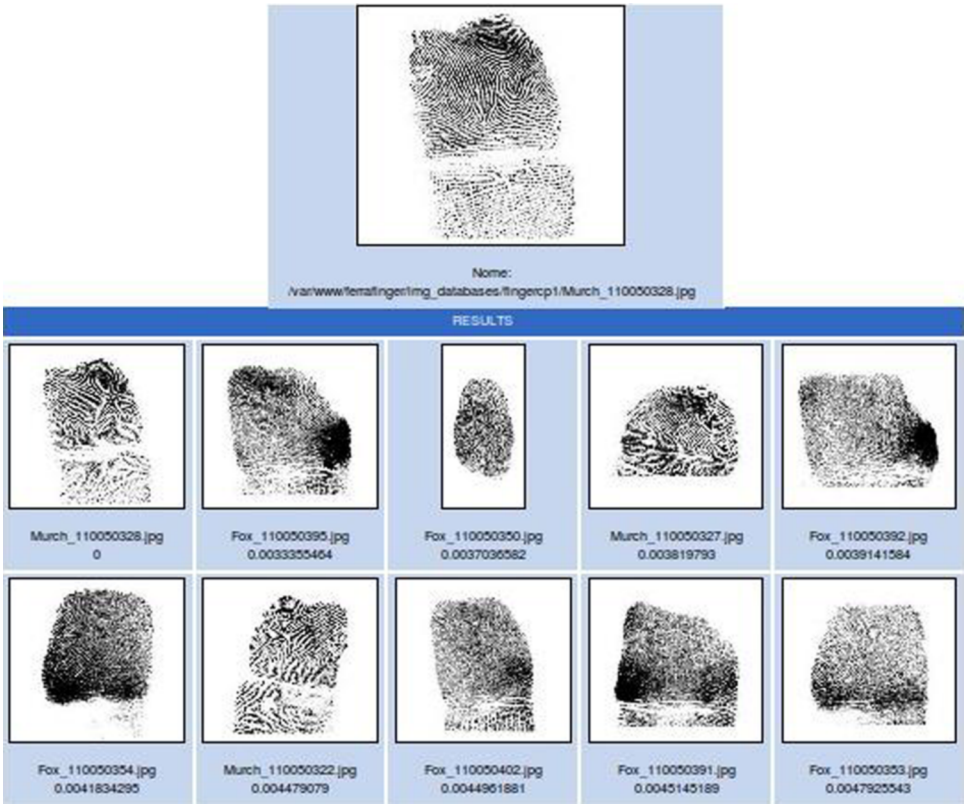

FIGURE 1.10: CBIR process for Figure 1.8—fingerprint 11.

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 platform or programming environment.

The encapsulation of resources was built in a three-layer model (as shown in Figure 1.12): (i) the CIO aggregating the CBIR and image information (encapsulated in the DCC entitled ImageCODCC); (ii) the individual fingerprint DL, represented by the police fingerprint DL (encapsulated in the DCC entitled PoliceCODCC) and the crime scene DL (encapsulated in the DCC entitled CrimeCODCC); and (iii) the individual complex object, aggregating all the images and metadata for a same person (encapsulated in the DCC entitled IndividualDCC).

In the mentioned example, Figure 1.8 (fingerprint 11) and Figure 1.9 (fingerprint 3) were aggregated into two ICOs. They are represented by the ImageCODCC, which centralizes the

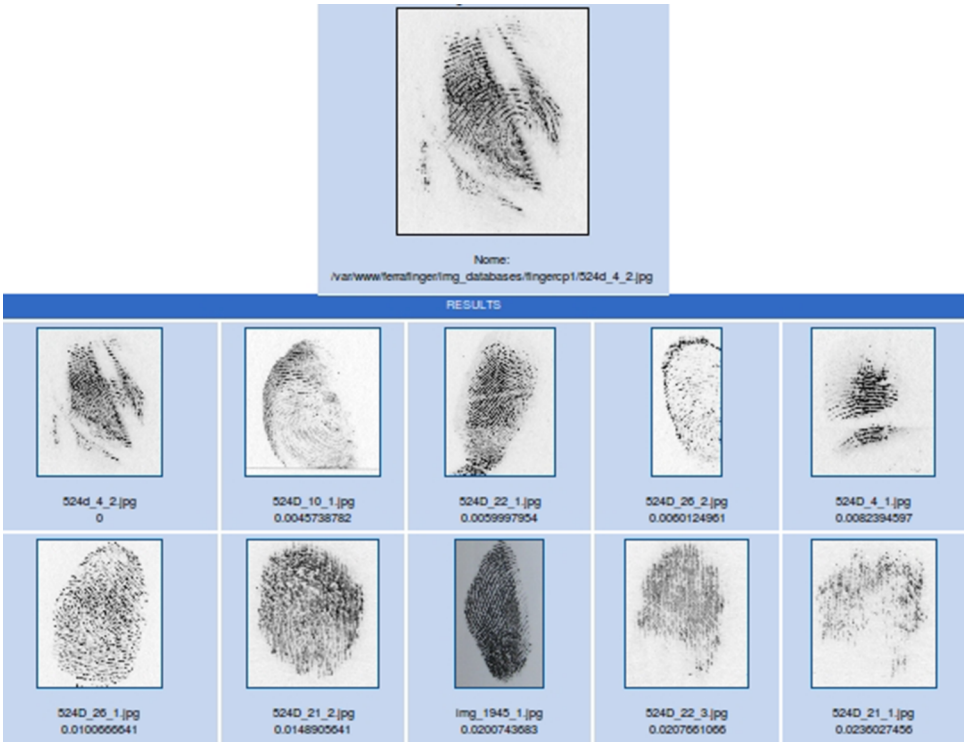

FIGURE 1.11: CBIR process for Figure 1.9—fingerprint 3.

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.

The second layer contains the information aggregation relative to the respective fingerprint library. In this case, Figure 1.8 (fingerprint 11) belongs to an individual from the police fingerprint DL and is encapsulated in PoliceCODCC. Figure 1.9 (fingerprint 3) belongs to the same individual, but now in the crime scene DL, which is encapsulated in CrimeCODCC. Operations available include the generation of the CO compressed file for the respective fingerprint DL. DCC metadata includes the individual name, the finger position, and the object substrate of the crime scene fingerprint.

The third layer corresponds to the Complex Object 1 presented in Figure 1.7, aggregating information from one individual using different fingerprint DLs. In the mentioned example, this represents the aggregation of all images from Figures 1.8 and 1.9 (since they represent the same

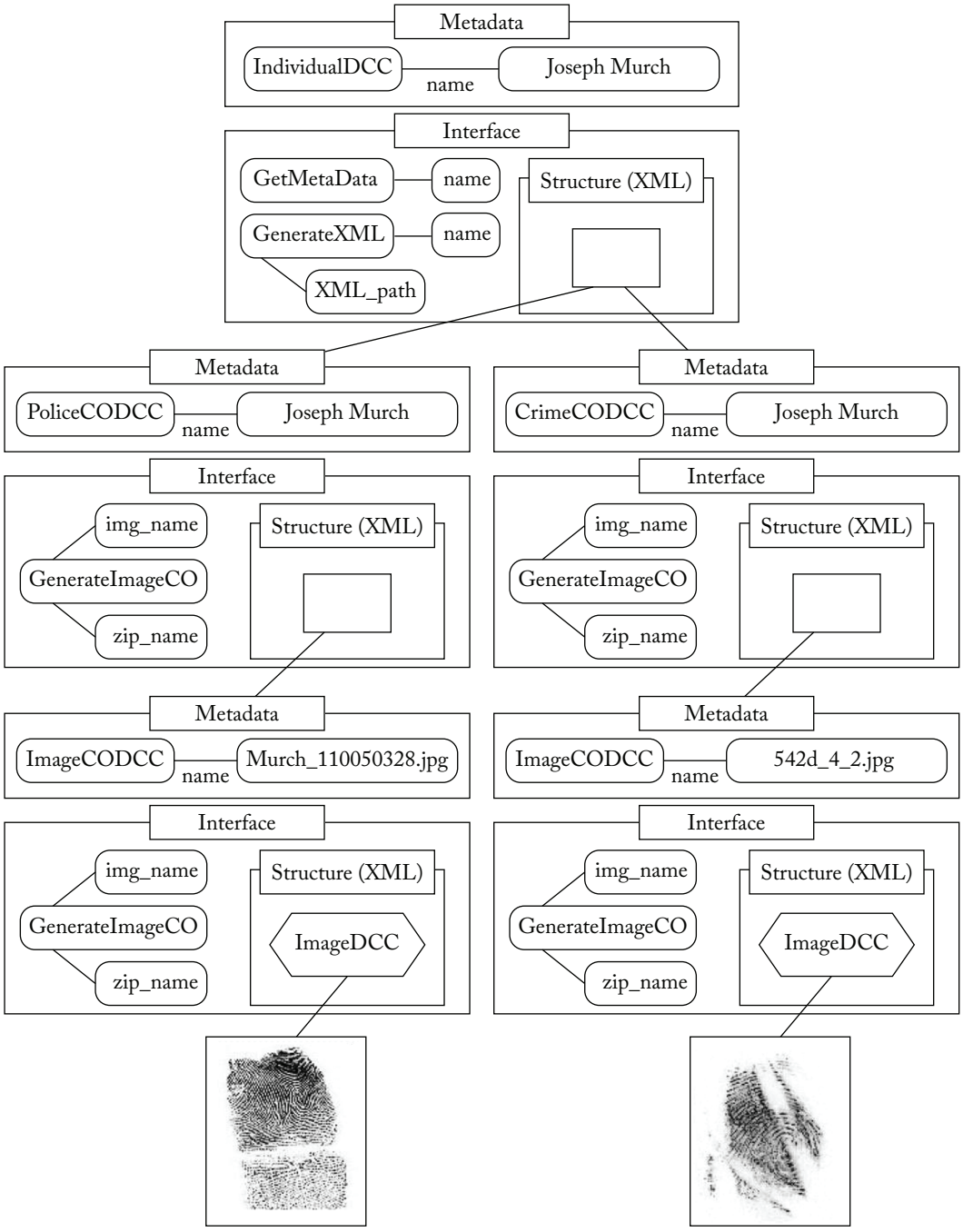

FIGURE 1.12: Structure for IndividualDCC.

```
<?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_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 1.13: XML for the individual aggregation.

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

Phase 5

In phase 5, the OAI-PMH protocol was used for the publishing of the individual CO metadata. It also can be used to understand which complex 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) [28] descriptor was used. The library was implemented in C, the DCCs in Java [104]. The functions and parameters available for each DCC are described in the PostgreSQL database. The image COs are published using the jOAI software [222]. 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.

1.6 SUMMARY

Many DL 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 chapter, we introduce formal definitions and descriptions of Complex Objects. The proposed constructs take advantage of formalism to help one to understand clearly and unambiguously the characteristics, structure, and behavior of the main concepts related to CO components, technologies, and applications. Later, these definitions are used in a case study, to exemplify how CO concepts can be explored to define the CIO. 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 CIOs in applications.

The set of definitions may impact future development efforts of a wide range of DL experts since it can guide the design and implementation of new DL services based on COs. Another straightforward benefit of this work is the use of these formal definitions to construct applications (as with image collections), including requirement gathering, conceptual modeling, prototyping, and code generation, similar to initiatives presented in [71, 232, 110]. As an example, consider the use of 5S formal theory to integrate an archaeological DL (Appendix C in Book 1), using applications such as 5SGraph [232]. From the implementation perspective, COs also can be used for service reuse and combination [98].

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 comparison and interaction with other technologies (such as the use of metadata in METS and Dublin Core), and the use of COs in other domains and specific services (such as content-based and multimodal search, and annotation).

1.7 EXERCISES AND PROJECTS

1.1 Pick your favorite DL. Identify three different types of complex objects that are important in that DL.

1.2 For each type of complex object mentioned previously, identify possible services and users. Why can COs help different users to have different views/layers of information?

1.3 How could one extend the 5S framework to define other CO-related concepts, such as CO packaging and content-based search services?

1.4 Besides data aggregation, the concept of complex object could be used for service integration. Identify two services that could be combined in the DL mentioned previously (in question 1).

1.5 Please give a 5S-oriented description for the fingerprint case study presented in this chapter. Be sure to cover each of the Ss separately, first. Then, consider combinations of pairs or triples of Ss too, as seems appropriate.

1.6 Using the CO definition, how can we formally describe DCC, OAI-ORE, and Buckets?

1.7 Please list and explain another example of CO technology that could be easily integrated with DCC.

1.8 Consider the DCC illustrated in Figure 1.12. How can we formally describe this CO? What would be the formal description, if we consider CIOs instead of images?

1.9 Consider the XML illustrated in Figure 1.13. How can we formally describe the CIOs presented in the aggregation? How many digital objects are present in this figure? How could we modify the CO for supporting image annotations?

1.10 Still considering Figure 1.13, list the advantages of publishing/integrating individual digital objects and COs. Please write another scenario where aggregations are useful for publishing.

In this case, what are the advantages of using XML? What other kind of structure would you suggest?

1.11 Consider a software Soft_SO that is composed of individual components DO1 and DO2. Can we apply the complex object definition under Soft_SO? What are the main differences compared to other objects, such as documents and images?

1.12 This chapter discussed three CO technologies. Pick some other technology used to aggregate objects, and formalize it by taking into account 5S aspects. Discuss the key limitations of this technology compared to DCC, OAI-ORE, and Buckets.

1.13 Chapter 2 of Book 2 discusses the DL integration problem. Can we extend the same problem under the complex object management perspective? What are the similarities?

1.14 Three students are working together in a group to solve an assignment from a DL class, and then one of them will send the group solution to the professor on behalf of everyone in the group. Discuss different scenarios of how the assignment was divided. Can we apply complex object definition in order to describe these scenarios? Are there limitations?

1.15 Pick some aspect of CO that has not been formally described in this chapter. Building upon the discussion in this chapter and in the prior books in the series, add to the formalisms given, to characterize the aspect of concern. Explain how the 5S framework has helped, or made more difficult, this formal approach.