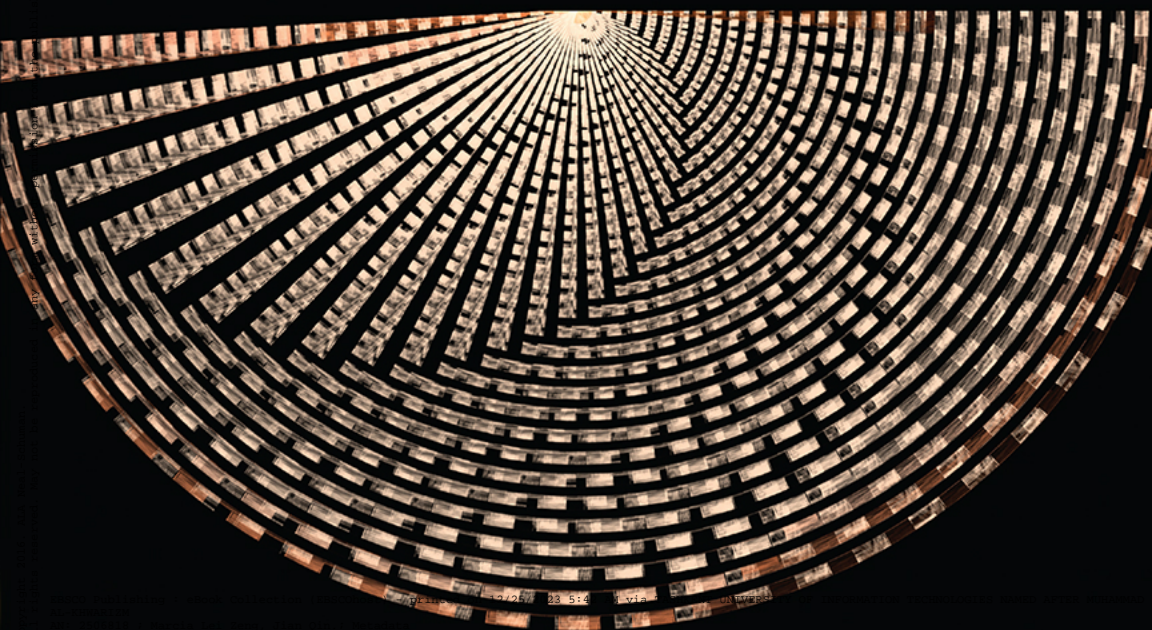




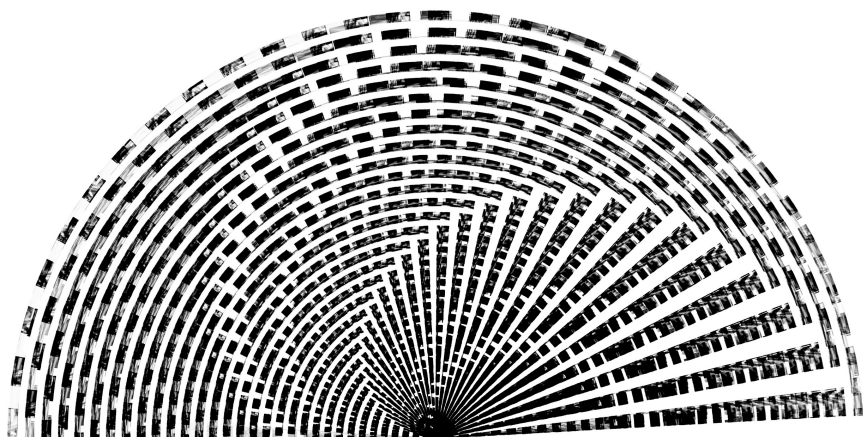
# METADATA

2ND EDITION > MARCIA LEI ZENG + JIAN QIN



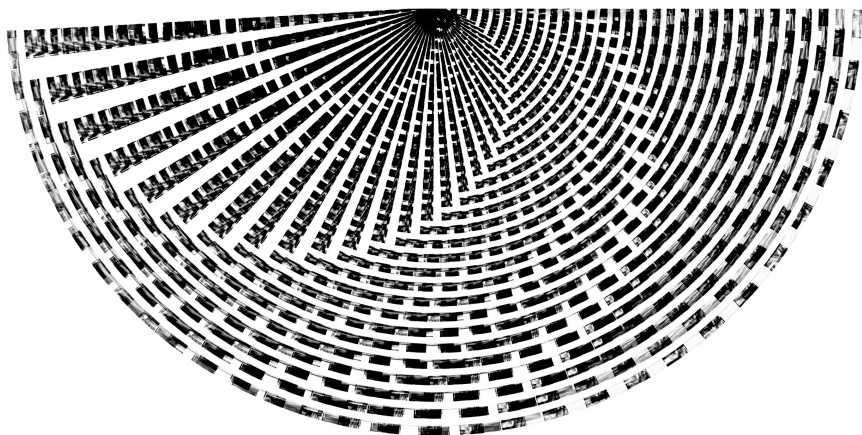
# METADATA

ALA Neal-Schuman purchases fund advocacy, awareness, and accreditation programs  
for library professionals worldwide.



# METADATA

2ND EDITION > MARCIA LEI ZENG + JIAN QIN



**Neal-Schuman**

An imprint of the American Library Association

CHICAGO 2016

## ADVISORY BOARD

**Jung Sun Oh**, Assistant Professor, School of Information Sciences, University of Pittsburgh

**Karen Snow**, Assistant Professor, Graduate School of Library & Information Science,  
Dominican University

**Katherine M. Wisser**, Assistant Professor, School of Library and Information Science,  
Simmons College

---

© 2016 by the American Library Association

Extensive effort has gone into ensuring the reliability of the information in this book; however, the publisher makes no warranty, express or implied, with respect to the material contained herein.

ISBN: 978-1-55570-965-5 (paper); 978-0-8389-4863-7 (PDF)

### Library of Congress Cataloging-in-Publication Data

Zeng, Marcia Lei, 1956-

Metadata/Marcia Lei Zeng and Jian Qin. —Second edition.

pages cm

Includes bibliographical references and index.

ISBN 978-1-55570-965-5 (paperback : alkaline paper) 1. Metadata. I. Qin, Jian, 1956- II. Title.

Z666.7.Z46 2015

025.3—dc23

2015013066

Cover design by Kimberly Thornton. Imagery © Shutterstock, Inc.

Text design by Alejandra Diaz in the Adobe Garamond Pro and Myriad Pro typefaces.

© This paper meets the requirements of ANSI/NISO Z39.48-1992 (Permanence of Paper).

Printed in the United States of America

20 19 18 17 16      5 4 3 2 1

# CONTENTS

<i>List of Illustrations</i> .....	<i>xiii</i>
<i>List of Exhibits</i> .....	<i>xxi</i>
<i>List of Tables</i> .....	<i>xxiii</i>
<i>Preface</i> .....	<i>xxv</i>

## PART I Fundamentals of Metadata

<b>ONE   Introduction</b> .....	<b>3</b>
1.1 Background .....	3
1.2 Definitions .....	11
1.3 A Brief History .....	14
1.4 Types and Functions .....	18
1.5 Standards .....	23
1.6 Principles .....	26
1.7 Examples of Metadata Descriptions .....	28
▶ Suggested Readings .....	36
▶ Exercises .....	36
<b>TWO   Understanding Metadata Vocabularies</b> .....	<b>37</b>
2.1 Metadata Element Sets .....	38
2.1.1 Components and Structures—An Overview .....	38
2.1.2 Flat Structure .....	42
2.1.3 Nested Structure .....	51
2.2 Application Profiles .....	54
2.2.1 The Concept of Application Profile .....	54
2.2.2 Examples of APs Consisting of Elements Drawn from Other Schemas .....	55
2.2.3 Sources of Reusable Elements .....	58

- 2.3 **Ontologies as Metadata Vocabularies** ..... 59
  - 2.3.1 Background ..... 59
  - 2.3.2 Modular Structure ..... 62
  - 2.3.3 Friend of a Friend (FOAF) ..... 62
  - 2.3.4 Schema.org ..... 64
- 2.4 **RDF Vocabularies for Metadata Terms** ..... 67
  - 2.4.1 An Introduction to RDF (Resource Description Framework) ..... 67
  - 2.4.2 DCMI Metadata Terms ..... 72
  - 2.4.3 Metadata Descriptions: From “Records” to “Statements” ..... 78
- ▶ **Suggested Readings** ..... 80
- ▶ **Exercises** ..... 80

**THREE | Creating Metadata Descriptions** ..... **81**

- 3.1 **Requirements for Metadata** ..... 82
- 3.2 **Basic Unit of Metadata** ..... 85
  - 3.2.1 Metadata Statement, Description, and Description Set ..... 85
  - 3.2.2 Relationships between Resources ..... 88
- 3.3 **Knowing the Difference** ..... 89
- 3.4 **Levels of Granularity** ..... 93
  - 3.4.1 Describing Individual Items That Constitute a Collection:  
Item-level Description ..... 93
  - 3.4.2 Describing the Entirety of a Collection: Collection-Level Description ..... 93
  - 3.4.3 Dataset-Level Metadata ..... 95
  - 3.4.4 Resource Decomposition ..... 97
- 3.5 **Metadata Sources** ..... 99
  - 3.5.1 Manual Generation of Metadata ..... 100
  - 3.5.2 Automatic Generation of Metadata ..... 102
  - 3.5.3 Combination of Manual and Automatic Methods ..... 104
  - 3.5.4 Harvested Metadata ..... 107
  - 3.5.5 Converted Metadata ..... 109
  - 3.5.6 User-Contributed Metadata through Social Media ..... 112
- 3.6 **Metadata Storage** ..... 114
  - 3.6.1 Internal Storage ..... 114
  - 3.6.2 External Storage ..... 121
- 3.7 **Expressing Metadata** ..... 124
  - 3.7.1 HTML ..... 125
  - 3.7.2 XML ..... 130
  - 3.7.3 RDF/XML and Other RDF Serialization Formats ..... 133

<b>3.8 Linkage, Wrapper, Display, and Parallel Metadata</b> .....	136
3.8.1 Linking between Descriptions for Different Resources .....	136
3.8.2 Wrapping .....	137
3.8.3 Encoding for Display .....	139
3.8.4 Encoding for Bilingual Metadata <i>Statements</i> .....	140
<b>3.9 Combining Metadata Descriptions and Linking Resources</b> .....	143
3.9.1 METS .....	144
3.9.2 RDF/XML .....	148
3.9.3 Aggregation .....	149
▶ <b>Suggested Readings</b> .....	152
▶ <b>Advanced Readings</b> .....	153
▶ <b>Exercises</b> .....	153

## PART II

### Metadata Vocabulary Building Blocks

<b>FOUR   Metadata Structures and Semantics</b> .....	<b>157</b>
<b>4.1 Modeling for Metadata</b> .....	160
4.1.1 Entity-Relationship Modeling .....	161
4.1.2 Ontological Modeling .....	165
4.1.3 Encapsulated and Modularized Approaches .....	171
<b>4.2 Enumerating Metadata Terms</b> .....	173
4.2.1 Communicating about the Functional Requirements .....	173
4.2.2 Identifying Desired Elements .....	175
<b>4.3 Element Set Specification</b> .....	176
4.3.1 Basic Components .....	177
4.3.2 Presentation .....	178
4.3.3 Principles for an Element Set to Follow .....	180
4.3.4 Methodologies for Working from an Existing Element Set .....	182
4.3.5 Testing the Element Set .....	186
<b>4.4 Value Spaces and Value Vocabularies</b> .....	187
4.4.1 Value Spaces That Should Follow Standardized Syntax Encoding Rules .....	188
4.4.2 Value Spaces That Require Standardized Vocabulary Encoding Schemes .....	190
4.4.3 Value Spaces That Require Predefined Lists of Terms .....	197



<b>4.5</b>	<b>Crosswalks</b> .....	202
4.5.1	Methods Used in Crosswalking .....	202
4.5.2	Aligning Elements with Indicators of Matching Degrees .....	205
<b>4.6</b>	<b>Best Practice Guides and Other Content Guidelines</b> .....	206
4.6.1	Best Practice Guides .....	207
4.6.2	Standard-Specific Guidelines .....	210
4.6.3	Community-Oriented Best Practice Guides .....	212
4.6.4	Data Content Standards .....	212
<b>4.7</b>	<b>Conclusion</b> .....	214
▶	<b>Suggested Readings</b> .....	214
▶	<b>Exercises</b> .....	215
<b>FIVE</b>	<b>  Metadata Schemas</b> .....	<b>219</b>
<b>5.1</b>	<b>Background</b> .....	219
<b>5.2</b>	<b>Resource Identification</b> .....	224
<b>5.3</b>	<b>Namespaces</b> .....	228
<b>5.4</b>	<b>Schema Encoding</b> .....	232
5.4.1	Relational Schema .....	232
5.4.2	XML Schema .....	233
5.4.3	Schema Encoding in Mixed Namespaces .....	237
<b>5.5</b>	<b>Encoding Examples of Metadata Vocabularies</b> .....	239
5.5.1	Dublin Core Schemas .....	240
5.5.2	EAD 2002 XML Schema .....	241
5.5.3	DLESE Metadata Framework XML Schemas .....	244
<b>5.6</b>	<b>Summary</b> .....	246
▶	<b>Suggested Readings</b> .....	246
▶	<b>Exercises</b> .....	247

## PART III

### Metadata Services

<b>SIX</b>	<b>  Metadata Services</b> .....	<b>251</b>
<b>6.1</b>	<b>Metadata Services as an Infrastructure</b> .....	252
<b>6.2</b>	<b>Metadata Registries</b> .....	258
6.2.1	Functional Requirements .....	259
6.2.2	Types .....	261
6.2.3	Essential Components .....	264

<b>6.3</b>	<b>Metadata Repositories</b>	267
6.3.1	Metadata Harvesting Model	269
6.3.2	OAI-PMH Commands	272
6.3.3	Support for Multiple Description Formats in OAI-PMH	275
<b>6.4</b>	<b>Metadata as Linked Data</b>	277
6.4.1	Two Roles of LAMs	278
6.4.2	Using Knowledge Organization Systems (KOS) as the Connectors of Linked Datasets	279
6.4.3	Metadata in Information Silos	284
6.4.4	Factors Impacting Metadata Linkability	285
<b>6.5</b>	<b>Ensuring Optimal Metadata Discovery and Increasing Findability</b>	294
6.5.1	Metadata Retrieval	295
6.5.2	Metadata Exposure Methods	299
<b>6.6</b>	<b>Summary</b>	313
▶	<b>Suggested Readings</b>	314
▶	<b>Exercises</b>	315
<b>SEVEN   Metadata Quality Measurement and Improvement</b>		<b>317</b>
<b>7.1</b>	<b>Quality of Metadata</b>	317
<b>7.2</b>	<b>Meeting the Functional Requirements</b>	319
<b>7.3</b>	<b>Quality Measurement with Different Granularities</b>	322
<b>7.4</b>	<b>Metadata Quality Measurement Indicators: CCCD</b>	325
7.4.1	Completeness	325
7.4.2	Correctness	327
7.4.3	Consistency	329
7.4.4	Duplication Analysis	331
<b>7.5</b>	<b>Metadata Evaluation</b>	334
<b>7.6</b>	<b>Enhancing Quality of Metadata</b>	337
<b>7.7</b>	<b>Entity-Level Quality for Reusable Metadata</b>	340
▶	<b>Suggested Readings</b>	345
▶	<b>Exercises</b>	346
<b>EIGHT   Achieving Interoperability</b>		<b>347</b>
<b>8.1</b>	<b>Definitions</b>	348
<b>8.2</b>	<b>Metadata Decisions at Different Stages of a Digital Library Project</b>	349
<b>8.3</b>	<b>Achieving Interoperability at the Schema Level</b>	351
8.3.1	Derivation	351
8.3.2	Application Profiles (APs)	353

8.3.3	Crosswalks .....	354
8.3.4	Frameworks .....	356
8.3.5	Metadata Registries .....	359
<b>8.4.</b>	<b>Achieving Interoperability at the Record Level .....</b>	<b>360</b>
8.4.1	Conversion of Metadata Records .....	360
8.4.2	Data Reuse and Integration .....	361
<b>8.5</b>	<b>Achieving Interoperability at the Metadata Repository Level .....</b>	<b>362</b>
8.5.1	Metadata Repositories Based on the Open Archives Initiative (OAI) Protocol .....	363
8.5.2	Metadata Repositories Supporting Multiple Formats without Record Conversion .....	364
8.5.3	Aggregation and Enriched Metadata Records in a Repository .....	365
8.5.4	Element-Based and Value-Based Crosswalking Services .....	366
8.5.5	Value-Based Mapping for Cross-Database Searching .....	367
8.5.6	Value-Based Co-Occurrence Mapping .....	367
<b>8.6</b>	<b>Alignment Approaches Used for Linked Data .....</b>	<b>370</b>
8.6.1	The Need for Alignment of Metadata Vocabularies .....	370
8.6.2	Alignment at Class Level .....	371
8.6.3	Alignment at Property Level .....	372
8.6.4	Mapping Degrees .....	373
<b>8.7.</b>	<b>Conclusion .....</b>	<b>375</b>
▶	<b>Suggested Readings .....</b>	<b>375</b>
▶	<b>Exercise .....</b>	<b>376</b>

## PART IV

### Metadata Outlook in Research

<b>NINE  </b>	<b>Metadata Research Landscape .....</b>	<b>379</b>
9.1	Overview .....	379
9.2	Research in Metadata Architecture .....	384
9.3	Research in Metadata Modeling .....	386
9.4	Research in Metadata Semantics .....	390
9.5	Metadata and Data-Driven X .....	393
9.6	Metadata Research Landscape: Conclusions .....	396
▶	Suggested Readings .....	397
▶	Exercise .....	397

## PART V

# Metadata Standards

<b>TEN   Current Standards</b> .....	<b>401</b>
<b>10.1 Metadata for General Purposes</b> .....	<b>402</b>
10.1.1 Dublin Core (DC) .....	402
10.1.2 MODS and the MARC Family .....	405
<b>10.2 Metadata for Cultural Objects and Visual Resources</b> .....	<b>414</b>
10.2.1 Introduction to CDWA .....	415
10.2.2 Important Concepts .....	417
10.2.3 Element Sets of CCO, CDWA Lite, LIDO, and VRA Core .....	420
10.2.4 Object ID Checklist .....	426
10.2.5 Value Vocabularies .....	428
<b>10.3 Metadata for Research Data</b> .....	<b>429</b>
10.3.1 Overview .....	429
10.3.2 Metadata Standards for Geospatial Data .....	432
10.3.3 Metadata Standards for Biodiversity and Ecology Data .....	438
10.3.4 Metadata for Social Sciences Research Data .....	441
10.3.5 Other Developments .....	444
<b>10.4 Metadata for Archives</b> .....	<b>445</b>
10.4.1 Background .....	445
10.4.2 Finding Aid Examples .....	446
10.4.3 EAD 2002 Record at a Glance .....	449
10.4.4 EAC-CPF .....	453
10.4.5 Related Standards .....	454
10.4.6 EAD3 .....	455
<b>10.5 Rights Management Metadata</b> .....	<b>456</b>
10.5.1 Rights Metadata Elements for User-Oriented Rights Information .....	457
10.5.2 Metadata Activities of Rights-Holder Communities .....	459
10.5.3 Open Digital Rights Language (ODRL) .....	461
<b>10.6 Metadata for Publishing and Press Communications</b> .....	<b>463</b>
10.6.1 ONIX (ONline Information Exchange) .....	464
10.6.2 EPUB .....	465
10.6.3 IPTC Metadata Standards .....	466
<b>10.7 Metadata for Multimedia Objects</b> .....	<b>467</b>
10.7.1 The MPEG standards .....	468
10.7.2 MPEG-7 .....	468
10.7.3 ID3v2 .....	471
10.7.4 PBCore, the Public Broadcasting Metadata Dictionary .....	471

<b>10.8 Preservation and Provenance Metadata</b> .....	473
10.8.1 Digital Preservation Metadata Standards .....	473
10.8.2 OAIS Reference Framework .....	474
10.8.3 Preservation Metadata: Implementation Strategies (PREMIS) .....	476
10.8.4 PROV for Provenance Interchange on the Web .....	479
10.8.5 DCMI Metadata Terms for Provenance .....	481
<b>10.9 Metadata Describing Agents</b> .....	482
10.9.1 vCard .....	482
10.9.2 FOAF (Friend of a Friend) .....	483
▶ <b>Suggested Readings</b> .....	484
▶ <b>Exercises</b> .....	484



*Appendixes: The following appendixes are available online only.*

**A** Metadata Standards: Metadata Schemas, Application Profiles, and Registries  
<http://www.metadataaetc.org/book-website/readings/appendixaschemas.htm>

**B** Value Encoding Schemes and Content Standards  
<http://www.metadataaetc.org/book-website/readings/appendixbencodschemes.htm>

<i>Glossary</i> .....	487
<i>References</i> .....	497
<i>About the Authors</i> .....	533
<i>Index</i> .....	535

# ILLUSTRATIONS

<b>Figure 1-1-1</b>	A food label as a metadata description for a single item .....	4
<b>Figure 1-1-2</b>	“Leonardo da Vinci” Knowledge Graph from Google and “Leonardo da Vinci” Infobox from Wikipedia .....	6
<b>Figure 1-1-3</b>	A digitized book presented with browsing categories on the top on the first screen, and page navigation options on the left of the second screen .....	8
<b>Figure 1-1-4</b>	A recovered stolen art work documented in the INTERPOL online database .....	9
<b>Figure 1-1-5</b>	Annotated screenshot of the filtering search result for the query “Raphael” .....	10
<b>Figure 1-2-1</b>	A simplified explanation of the variations in terminology found in different communities .....	13
<b>Figure 1-4-1</b>	A screenshot of a metadata instance for a satellite map image, showing automatically captured file properties on the Adobe Bridge metadata template .....	22
<b>Figure 1-5-1</b>	Types of metadata standards .....	23
<b>Figure 1-5-2</b>	Illustration of the relationships among various types of standards, using CCO as an example .....	25
<b>Figure 1-7-1</b>	A metadata description for a track in iTunes (right), with the View Options (left) and a display showing sortable properties (marked as #3) .....	29
<b>Figure 1-7-2</b>	The <i>Metadata etc.</i> web site: Front-end landing page, back-end page source, and “Page Info” interpreted into human readable format by the browser .....	30
<b>Figure 1-7-3</b>	A simplified example of embedding semantic markup in a web page .....	31
<b>Figure 1-7-4</b>	A metadata description embedded in a PDF file .....	32
<b>Figure 1-7-5</b>	Screenshots of a metadata description template (left) and the configuration at the back-end (right) .....	33

<b>Figure 1-7-6</b>	A metadata description example as seen from the front-end of eCommons@Cornell, the institutional repository at Cornell University .....	35
<b>Figure 1-7-7</b>	A metadata description about a plant drawing, as seen from the front-end of the online Botany Collections of the Smithsonian National Museum of Natural History .....	35
<b>Figure 2-1-1</b>	An entry for DC element <i>date</i> .....	39
<b>Figure 2-1-2</b>	An entry for VRA Core 4 element <i>title</i> .....	40
<b>Figure 2-1-3</b>	The 15 Dublin Core elements seen from three categories .....	44
<b>Figure 2-1-4</b>	Illustration on the importance of best practice recommendations provided by a standard, using the example of dc:date element .....	49
<b>Figure 2-1-5</b>	VRA Core 4.0 elements and sub-elements .....	52
<b>Figure 2-1-6</b>	VRA Core 4.0 <i>agent</i> element and sub-elements .....	53
<b>Figure 2-2-1</b>	Illustration of an application profile consisting of metadata elements and refinements drawn from one or more schemas .....	55
<b>Figure 2-2-2</b>	AVEL metadata element list .....	56
<b>Figure 2-2-3</b>	Portion of the metadata elements within the NSDL_DC metadata framework (NSDL 2007) updated 2013 .....	57
<b>Figure 2-3-1</b>	The Semantic Web Layer Cake .....	60
<b>Figure 2-3-2</b>	A FOAF file for a <i>Person</i> instance with selected properties describing personal information, work place and educational background, plus another person one knows .....	63
<b>Figure 2-3-3</b>	Explanation of the Type Hierarchy view and Individual Type view of “Book” type .....	65
<b>Figure 2-4-1</b>	Example of an RDF graph (center of the figure) that has two nodes ( <i>subject</i> and <i>object</i> ), with a <i>predicate</i> connecting them .....	68
<b>Figure 2-4-2</b>	An illustration of a group of triples in an RDF graph for a book .....	69
<b>Figure 2-4-3</b>	An illustration of a group of triples in an RDF graph for a person .....	69
<b>Figure 2-4-4</b>	Illustration of two connected graphs .....	70
<b>Figure 2-4-5</b>	RDF serialization formats at a glance .....	71
<b>Figure 2-4-6</b>	dcterms:creator .....	73
<b>Figure 2-4-7</b>	A simple explanation of “range” constraint .....	75
<b>Figure 2-4-8</b>	dcterms:accrualPeriodicity’s domain and range .....	76
<b>Figure 2-4-9</b>	Descriptions based on properties from DCMES and DC Terms .....	78
<b>Figure 3-2-1</b>	Basic model: Resource with properties and relationships to other resources .....	86
<b>Figure 3-2-2</b>	DC Abstract Model in a simplified graphical example .....	87

<b>Figure 3-2-3</b>	CDWA's Entity-Relationship Diagram .....	88
<b>Figure 3-4-1</b>	The decomposition of a resource .....	98
<b>Figure 3-5-1</b>	Subject headings and keywords provided by a manually created catalog description for a PhD dissertation (left) and the social tags and entity names generated by a semantic analysis tool (right) .....	105
<b>Figure 3-5-2</b>	Illustration of harvesting based on a common protocol .....	108
<b>Figure 3-6-1</b>	A screenshot of the front-end entry of a computer science technical report (Persistent URI: <a href="http://resolver.caltech.edu/CaltechCSTR:2002.003">http://resolver.caltech.edu/CaltechCSTR:2002.003</a> ) .....	116
<b>Figure 3-6-2</b>	Metadata statements embedded in the entry viewed from the Source Code (left) and the Page View (right) for the entry shown in figure 3-6-1 .....	117
<b>Figure 3-6-3</b>	Metadata partially captured automatically and partially entered manually are embedded in a Microsoft Word file's <i>Properties</i> section .....	118
<b>Figure 3-6-4</b>	Description template and data values partially captured automatically and partially entered manually in an Adobe Acrobat PDF file .....	119
<b>Figure 3-6-5</b>	Metadata workspace and template options provided by Adobe Bridge .....	120
<b>Figure 3-6-6</b>	A screenshot of record editing using CONTENTdm software .....	122
<b>Figure 3-6-7</b>	A web-accessible database displaying metadata records in a table ...	123
<b>Figure 3-6-8</b>	Individual metadata record in the Cleveland Memory Project database .....	123
<b>Figure 3-7-1</b>	A taxonomy of general markup languages .....	125
<b>Figure 3-7-2</b>	<i>Metadata etc.</i> web site's HTML document where, in the <body> section, RDFa codes are added (#1), from which the structured data (#3) are extracted using RDFa Play .....	129
<b>Figure 3-7-3</b>	Visualization of the structured data (#3 in figure 3-7-2), generated using RDFa Play software .....	130
<b>Figure 3-7-4</b>	Sample VRA Core description for <agent> when using an XML editor .....	131
<b>Figure 3-7-5</b>	VRA Core 4.0 Syntax with encoding examples .....	132
<b>Figure 3-7-6</b>	Graph of the metadata description set using the RDF/XML codes from exhibit 3-7-3, generated by W3C RDF Validation Service .....	135
<b>Figure 3-8-1</b>	An example of parallel metadata values .....	142
<b>Figure 3-8-2</b>	A localization process based on a translation memory .....	142
<b>Figure 3-9-1</b>	The METS Architecture .....	145
<b>Figure 3-9-2</b>	A METS record example with a MODS record in the description metadata section .....	147



**Figure 3-9-3** METS record example with a reference link to a MODS record ..... 147

**Figure 3-9-4** Visualization of three “resources” described and graphs connected .. 149

**Figure 3-9-5** Sources, storage, and redistribution of augmented metadata in the NSDL Metadata Repository ..... 151

**Figure 4-1-1** Illustration of E-R modeling of music recordings, showing the entities and their relationships ..... 162

**Figure 4-1-2** An interpretation of the FRBR model ..... 164

**Figure 4-1-3** A demonstration of ontological modeling for music recordings ..... 166

**Figure 4-1-4** A diagram illustrating the basics of the BIBFRAME model ..... 169

**Figure 4-1-5** Example of DBpedia ontology classes presented in a hierarchy (left side) and a class and its properties (right side) ..... 170

**Figure 4-2-1** Metadata functionalities ..... 176

**Figure 4-3-1** An entry for the element *Display Creation Date* from CDWA Lite ..... 179

**Figure 4-3-2** Element “dc.contributor” and localized refinement from ETD-MS..... 185

**Figure 4-3-3** Selected elements defined by the National Library of Medicine Metadata Schema ..... 186

**Figure 4-4-1** An authority record for “Clinton, Bill” in the FAST Authority File ..... 195

**Figure 4-4-2** Display of place name entries that match the query “Columbus” in TGN ..... 195

**Figure 4-4-3** Display of a TGN authority record for “Columbus, Indiana, USA” ..... 196

**Figure 4-5-1** Absolute and relative crosswalking ..... 204

**Figure 4-5-2** Different degrees of element equivalency in crosswalked schemas. A1 and B1 represent elements from A and B schemas ..... 204

**Figure 4-5-3** Portion of the crosswalk of CDWA to other major metadata vocabularies ..... 205

**Figure 4-5-4** An alignment map showing the degrees of mapping between properties from different vocabularies ..... 206

**Figure 4-5-5** A “map” of matching classes from different namespaces ..... 207

**Figure 4-6-1** Examples for the date element provided by VRA Core 4.0 ..... 211

**Figure 4-6-2** Illustration of professionals working with museum objects and the need for uniform documentation ..... 213

**Figure 5-1-1** Using an XML schema (#1) in an editor in order to create machine-understandable metadata descriptions (#2) ..... 220

**Figure 5-1-2** LAM metadata schemas represented in standardized syntaxes in XML and RDF schema languages ..... 223

**Figure 5-4-1** A sample relational schema for a database storing metadata ..... 232

<b>Figure 5-5-1</b>	Schema modes: (a) single encoding schema, (b) multiple encoding schemas, and (c) networked encoding schemas .....	240
<b>Figure 5-5-2</b>	DC XML schemas .....	241
<b>Figure 5-5-3</b>	The top layer structure of <i>EAD 2002 Schema</i> .....	242
<b>Figure 5-5-4</b>	DLESE application profile XML schema structure .....	245
<b>Figure 5-5-5</b>	A dropdown list presents controlled vocabulary choices for catalogers .....	245
<b>Figure 6-1-1</b>	The infrastructure for metadata services .....	257
<b>Figure 6-2-1</b>	CORES Registry Index page .....	260
<b>Figure 6-2-2</b>	A document about the vocabularies referring to one another and the vocabulary history (using Schema.org as an example) .....	262
<b>Figure 6-2-3</b>	Record for the <i>creator</i> element from the DCMI Metadata Registry ....	266
<b>Figure 6-2-4</b>	Record for the <i>creator</i> element from the CORES Registry .....	266
<b>Figure 6-3-1</b>	Illustration of the OAI-PMH structure .....	271
<b>Figure 6-4-1</b>	Explanation of a bibliographic-data-centered mash-up generated on the fly at the AGRIS portal .....	281
<b>Figure 6-4-2</b>	Illustration of an AGRIS search result with external resources .....	281
<b>Figure 6-4-3</b>	Illustration of KOS and controlled values used in a CONA record describing a painting .....	283
<b>Figure 6-4-4</b>	Illustration of the linkability of <i>Responsible Body</i> information based on the study samples .....	291
<b>Figure 6-5-1</b>	A simplified illustration of metasearching .....	296
<b>Figure 6-5-2</b>	A simplified illustration of federated searching .....	298
<b>Figure 6-5-3</b>	Exposing metadata to enhance the visibility of content .....	300
<b>Figure 6-5-4</b>	A snapshot of the approaches to exposing metadata from the back-end .....	301
<b>Figure 6-5-5</b>	Demonstration of querying DBpedia using the Virtuoso SPARQL Editor (#1), obtain the results (#2) and visualizing (#3) the results using Gephi software .....	310
<b>Figure 6-5-6</b>	Demonstration of querying BNB through the Flint SPARQL Editor (#1) and obtaining the results (#2) .....	311
<b>Figure 6-5-7</b>	Demonstration of querying AAT through the Getty Vocabulary LOD SPARQL editor (top), using the template (#1) provided by the dataset service, modifying the query (#2) and obtaining the results (#3) .....	312
<b>Figure 7-2-1</b>	A Search result display showing missing information .....	320
<b>Figure 7-2-2</b>	Advanced filters that might have been effective only in limited portions of a database's data .....	321

<b>Figure 7-3-1</b>	A metadata record is compared with the original item .....	323
<b>Figure 7-3-2</b>	An embedded metadata record is compared with the web page .....	323
<b>Figure 7-3-3</b>	Comparing records between pre- and post-conversion .....	324
<b>Figure 7-4-1</b>	A description of physical item is displayed in a search result .....	330
<b>Figure 7-4-2</b>	A description of content is displayed after adjustment and re-indexing .....	331
<b>Figure 7-4-3</b>	Search results leading to the same source with virtually equivalent URLs (with or without “index.html”) .....	332
<b>Figure 7-4-4</b>	Same source and URL linked by different titles and descriptions .....	332
<b>Figure 7-4-5</b>	Not duplicates: titles are same or alike but lead to different sources .....	334
<b>Figure 8-2-1</b>	Various levels of metadata projects .....	350
<b>Figure 8-3-1</b>	Examples of schema derivation .....	352
<b>Figure 8-3-2</b>	Example of schema adaptation/modification by ETD-MS .....	353
<b>Figure 8-3-3</b>	Three models for developing application profiles .....	354
<b>Figure 8-3-4</b>	Establishing a crosswalk between two schemas .....	354
<b>Figure 8-3-5</b>	Cross-switching when multiple schemas are involved .....	355
<b>Figure 8-3-6</b>	A framework and the schemas associated with the framework .....	356
<b>Figure 8-3-7</b>	IIIF Image URI syntax explained .....	358
<b>Figure 8-3-8</b>	A metadata registry in relation to schemas .....	359
<b>Figure 8-4-1</b>	Record conversion .....	360
<b>Figure 8-5-1</b>	Metadata records are integrated into a metadata repository .....	363
<b>Figure 8-5-2</b>	Enriched metadata records .....	365
<b>Figure 8-5-3</b>	Illustration of the co-occurrence mapping approach .....	368
<b>Figure 8-5-4</b>	An ADL record showing assigned values from two vocabulary sources .....	369
<b>Figure 9-1-1</b>	Topic distribution by year in 207 full papers published in the <i>Proceedings of the DCMI Annual International Conference,</i> 2000–2014 .....	382
<b>Figure 9-1-2</b>	Distribution of types of studies reported in 207 full papers published in the <i>Proceedings of the DCMI Annual International</i> <i>Conference, 2000–2014</i> .....	383
<b>Figure 10-1-1</b>	MODS top level elements .....	410
<b>Figure 10-1-2</b>	Legend used in graphics generated from XML schemas .....	412
<b>Figure 10-1-3</b>	Sub-elements, attributes, and predefined list of attribute values for the <i>subject</i> element in MODS .....	413

<b>Figure 10-2-1</b>	CDWA broad categories .....	416
<b>Figure 10-2-2</b>	A many-to-many work-image relationship scenario .....	418
<b>Figure 10-2-3</b>	CDWA Lite Elements .....	423
<b>Figure 10-2-4</b>	Outline of the 7 areas of information in a LIDO record .....	425
<b>Figure 10-2-5</b>	Object ID Checklist metadata elements with examples .....	427
<b>Figure 10-3-1</b>	Metadata requirements for research data in support of data management, data quality, data discovery, and data use .....	430
<b>Figure 10-3-2</b>	Seven description areas and main elements in CSDGM .....	433
<b>Figure 10-3-3</b>	The graphical view of XML encoding for the elements at the first few levels of the Spatial Data Organization Information section and Spatial Reference Information sections in CSDGM .....	434
<b>Figure 10-3-4</b>	(a) Portion of an FGDC metadata record in XML format that conforms to the CSDGM standard. (b) Portion of an FGDC metadata record in (a). .....	435
<b>Figure 10-3-5</b>	Core and optional elements and the mandatory topical categories in ISO 19115:2003 <i>Geographic information — Metadata</i> .....	437
<b>Figure 10-3-6</b>	Categories of metadata elements in Darwin Core .....	439
<b>Figure 10-3-7</b>	Element groups in the Access to Biological Collection Data (ABCD) .....	440
<b>Figure 10-3-8</b>	Ecological Metadata Language (EML) structure and modules .....	440
<b>Figure 10-3-9</b>	Five modules in DDI-Lifecycle .....	442
<b>Figure 10-4-1</b>	A screenshot of a finding aid's Summary Information display .....	447
<b>Figure 10-4-2</b>	A screenshot of the Title Page display of the finding aid shown in figure 10-4-1 .....	447
<b>Figure 10-4-3</b>	A screenshot of the archival content from the finding aid displayed in figure 10-4-1 .....	448
<b>Figure 10-4-4</b>	A graphical presentation of the EAD [2002] structure .....	449
<b>Figure 10-4-5</b>	Examples of metadata statements under <archdesc> and <did> elements .....	452
<b>Figure 10-4-6</b>	EAD3 .....	456
<b>Figure 10-5-1</b>	Sub-elements of <copyright> element defined by the <i>copyrightMD</i> schema .....	458
<b>Figure 10-7-1</b>	The main MPEG-7 elements .....	469
<b>Figure 10-7-2</b>	Functional groups of the MPEG-7 Multimedia Description Schemes ...	470
<b>Figure 10-7-3</b>	BPCore's Content Classes and Containers .....	472
<b>Figure 10-8-1</b>	OAIS functional entities .....	475
<b>Figure 10-8-2</b>	Entities defined in the PREMIS Data Model .....	477
<b>Figure 10-8-3</b>	The Starting Point terms of the PROV-O Ontology .....	480



# EXHIBITS

<b>Exhibit 1-7-1</b>	Portion of the metadata description for the <i>Metadata etc.</i> web site .....	31
<b>Exhibit 3-4-1</b>	Collection-level metadata for NMNH Vertebrate Zoology Fishes Collections (Smithsonian Institute—Fishes) .....	97
<b>Exhibit 3-5-1</b>	A record from Alexandria Digital Library (ADL) Gazetteer .....	110
<b>Exhibit 3-5-2</b>	The ADL record retrieved from NSDL, after converting into DC format (data harvested by NSDL.org in 2004) .....	111
<b>Exhibit 3-6-1</b>	Examples of metadata statements from IBM web sites for different countries and languages .....	115
<b>Exhibit 3-7-1</b>	Demo #1. A Dublin Core metadata description set for the <i>Metadata etc.</i> web site, expressed in HTML and embedded in the <head> section .....	127
<b>Exhibit 3-7-2</b>	Demo #2. The metadata description set for the <i>Metadata etc.</i> web site, expressed in XML .....	133
<b>Exhibit 3-7-3</b>	Demo #3. A shorter version of the same metadata description set for the <i>Metadata etc.</i> web site, expressed with RDF/XML .....	134
<b>Exhibit 3-7-4</b>	Demo #4. The same metadata description set for the <i>Metadata etc.</i> web site, expressed with N3 (Notation3), generated by RDF Translator .....	135
<b>Exhibit 3-8-1</b>	Example of wrapper, element set, and sub-elements in CDWA .....	138
<b>Exhibit 3-8-2</b>	Example of a top-level element and its sub-elements in MODS .....	138
<b>Exhibit 3-8-3</b>	Example of a top-level element and its sub-elements in VRA 4.0 .....	139
<b>Exhibit 3-8-4</b>	Example of statements of <i>measurements</i> encoded for display .....	141
<b>Exhibit 3-8-5</b>	Example of statements about an <i>agent</i> encoded for display .....	141
<b>Exhibit 3-9-1</b>	Options for including a metadata description in a METS record .....	146
<b>Exhibit 3-9-2</b>	Demo #5. An RDF/XML document containing three “resources” described, Validated by W3C RDF validation service .....	148

<b>Exhibit 4-1-1</b>	Example class definitions from the BBC Music Ontology Specification .....	167
<b>Exhibit 4-1-2</b>	Examples of classes defined in the Bibliographic Ontology (BIBO) ....	168
<b>Exhibit 4-3-1</b>	Full presentation for the Darwin Core term (element) <i>occurrenceID</i> .....	179
<b>Exhibit 4-6-1</b>	Values assigned to <i>date</i> element in a metadata repository .....	209
<b>Exhibit 5-2-1</b>	URI examples for web document and RDF document .....	226
<b>Exhibit 5-4-1</b>	XML encoding example of the data model from figure 5-4-1 .....	234
<b>Exhibit 5-4-2</b>	An example of XML schema for the XML document shown in Exhibit 5-4-1 .....	235
<b>Exhibit 5-4-3</b>	Portion of the EDM XML schema .....	239
<b>Exhibit 5-5-1</b>	Portion of EAD 2002 XML schema .....	243
<b>Exhibit 5-5-2</b>	XML encoding for a portion of a finding aid based on the schema in Exhibit 5-5-1 .....	244
<b>Exhibit 6-3-1</b>	An example of OAI-PMH response to the <i>getRecord</i> request .....	273
<b>Exhibit 6-3-2</b>	Modified "ListMetadataFormats" responding to include ADN format ...	277
<b>Exhibit 7-4-1</b>	Values associated with <i>format</i> element found in a research sample ....	329
<b>Exhibit 10-1-1</b>	A MARC 21 (2709) record viewing by machine .....	407
<b>Exhibit 10-1-2</b>	A MARCXML record based on the MARC 21 (2709) record shown in exhibit 10-1-1 .....	408
<b>Exhibit 10-1-3</b>	A DC record converted from the MARCXML record shown in exhibit 10-1-2 .....	409
<b>Exhibit 10-1-4</b>	A MODS record converted from the MARCXML record shown in exhibit 10-1-2 .....	411
<b>Exhibit 10-3-1</b>	Excerpt of the <i>ref</i> attribute for the element FundingInformation from DDI's schema file "studyunit.xsd" .....	442
<b>Exhibit 10-3-2</b>	An excerpt for the element <i>FundingInformation</i> from DDI 3.2's schema file "reusable.xsd" .....	443
<b>Exhibit 10-4-1</b>	Outline of the EAC-CPF elements .....	454

# TABLES

<b>Table 2-1-1</b>	A sample recipe .....	41
<b>Table 2-1-2</b>	Side-by-side comparison of representations of the cake recipe structures .....	43
<b>Table 2-1-3</b>	Dublin Core Version 1.1 (DCMES) Elements with Refinements and Encoding Schemes .....	46
<b>Table 2-1-4</b>	Qualifiers become properties .....	47
<b>Table 2-1-5</b>	Two types of encoding schemes .....	48
<b>Table 2-1-6</b>	VRA Core 3.0 elements and qualifiers .....	50
<b>Table 2-1-7</b>	Example of VRA Core <i>agent</i> element in a description .....	53
<b>Table 2-4-1</b>	DCMI Namespaces .....	72
<b>Table 2-4-2</b>	Overview of DC Terms usage constraints .....	77
<b>Table 4-2-1</b>	A list of functional requirements for a digital repository of scholarly publications .....	174
<b>Table 4-3-1</b>	Matching situations and possible actions between an existing element set and one's desired element set .....	183
<b>Table 4-3-2</b>	ETD-MS Element Set (based on Dublin Core) actions and results .....	185
<b>Table 5-2-1</b>	A list of digital identifier systems for people, organizations, and objects .....	227
<b>Table 5-2-2</b>	Examples of reusing existing identifiers in permanent HTTP URIs .....	229
<b>Table 6-2-1</b>	Administration and identification attributes in a registry record and occurrence constraints .....	265
<b>Table 6-3-1</b>	The OAI-PMH requests and examples .....	272
<b>Table 6-3-2</b>	OAI-PMH record components and examples .....	274



<b>Table 6-4-1</b>	Scenarios and Examples from MARC 505 (Formatted Contents Note) and 511 (Participant or Performer Note) Fields (With No 7XX Additional Entries) .....	291
<b>Table 8-5-1</b>	Controlled vocabularies required or recommended for use in VRA Core metadata records .....	368
<b>Table 9-1-1</b>	Topics of metadata by research perspectives .....	384
<b>Table 10-2-1</b>	CCO—chapters and elements .....	421
<b>Table 10-3-1</b>	Metadata standards for research data, an incomplete list .....	431
<b>Table 10-5-1</b>	Elements dedicated to rights information in selected metadata standards .....	457
<b>Table 10-8-1</b>	Categorization of the DCMI Metadata Terms .....	481

# PREFACE

The field of metadata has undergone a great deal of change in the seven years since the first edition of this book was published. Many of these changes are closely tied to advances in Semantic Web technologies and in information technologies in general. As a result of these developments, some of the terminologies and standards used in the 2008 edition became outdated; at the same time, newer innovations and practices in metadata were noticeably absent from our coverage. Hence, the text badly needed an update.

We started planning for a second edition about two years ago, thinking we would finish the revision within a year. It turned out that this revision was ambitious in terms of the new content to be added, and that we had far underestimated its scope. During the revision, we updated the book's contents with our analysis of job descriptions, theses and dissertations, published journal articles, and conference proceedings related to metadata from the last several years. We drew on feedback from the community—from instructors of metadata classes, students, practitioners, and researchers—and on our own experience of teaching metadata courses and conducting metadata research in an attempt to uncover the most significant metadata developments and practices from the last seven years.

Returning readers may notice that the new edition has undergone a major structural change. Formerly comprised of four parts, it is now divided into five:

1. Fundamentals of metadata
2. Metadata vocabulary building blocks
3. Metadata services
4. Metadata outlook in research
5. Metadata standards

These new divisions can be described as follows:

Part I, Fundamentals of Metadata, includes chapters 1 through 3. Chapter 1 has been almost completely rewritten. In addition, it includes more images for an easy understanding of the basics of metadata, in order to provide readers with

a strong footing for the challenges of later chapters. Chapter 2 continues this discussion by addressing metadata vocabularies, including metadata element sets, application profiles, ontologies, and RDF vocabularies for metadata terms, the latter two of which are newly added topics. Completing this sequence, chapter 3 focuses on the creation of metadata descriptions.

Part II, Metadata Vocabulary Building Blocks, spans two chapters. The first of these, chapter 4, discusses the structures and semantics of metadata elements necessary for describing resources in a domain or of a certain type in addition to detailing metadata modeling, enumerating metadata terms, and using value vocabularies. Proceeding logically, chapter 5 moves on to encode these into machine-processable schemas, covering resource identification, namespaces, and schema encodings that are essential for implementing metadata models.

Part III, Metadata Services, retains most of the content from the first edition. However, new material in chapter 6 that is worth highlighting includes sections on metadata services as infrastructure and metadata as Linked Data. Similarly, in both chapter 7 on metadata quality and chapter 8 on interoperability, new sections related to Linked Data have been added. We understand that it is impossible to include complete coverage of the subject of Linked Data within this book; thus, its sections focus on areas most concerned with metadata.

Chapter 9, The Metadata Research Landscape, forms part IV of this edition. We had originally decided to cut this section due to the expansion of other chapters, but after considering the comments of reviewers, we agreed to restore it. This new version includes updates on metadata research and a brief discussion of the “data-driven x” phenomenon, as well as considerations of metadata’s role as a source of Big Data research and as an infrastructure for supporting data-driven x research and learning.

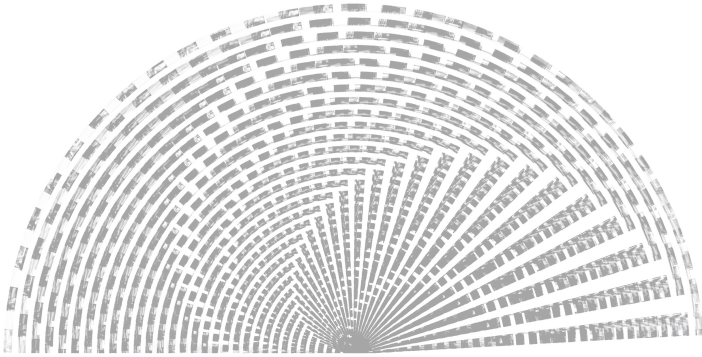
Finally, part V is devoted to standards. Chapter 10, Current Standards, which was the second chapter in the first edition, has been relocated to the last chapter of this new edition. The reason for moving it to the back is mainly due to its size and the aversion of readers to the overwhelming number of standards that were cited early in the old version. (In this edition, a limited number of standards are introduced in chapter 2 as examples of different structures and types of metadata vocabularies.) Chapter 10 includes a more extensive list of standards for general purposes, cultural objects and visual resources, research data, archives, rights management, publishing and press communications, multimedia objects, preservation and provenance, and metadata describing agents. Many sections in this chapter were rewritten, and facts were verified. We painstakingly checked all links to web sites and all timestamps for significant updates to standards, and then updated them to reflect the most recent changes. Some of these may already have had

new updates as this preface was being written! During the course of checking all references used in the first edition, we found that the URLs for over one hundred sources had been changed. These were updated or deleted accordingly.

There are some interesting facts about the illustrations in this new edition that are also worth sharing. First, when creating the illustrations, we have used the form of the tangram (“seven boards of skill”), a dissection puzzle consisting of seven flat shapes used to form various figures in thousands of configurations, which originated over 1,000 years ago. This type of puzzle requires forming a specific shape using all seven pieces without any overlap among them. For us, the tangram intuitively corresponds to the “modular structure” that we observed in metadata vocabulary development in recent years. Second, we captured many examples from the web and created annotations for the screenshots, which was our way of visualizing the practical suggestions and “walk-throughs” in the text. The annotations on the figures use a different font to serve two purposes: to distinguish our comments from the original text and to simulate the use of the book in a classroom or self-guided learning setting.

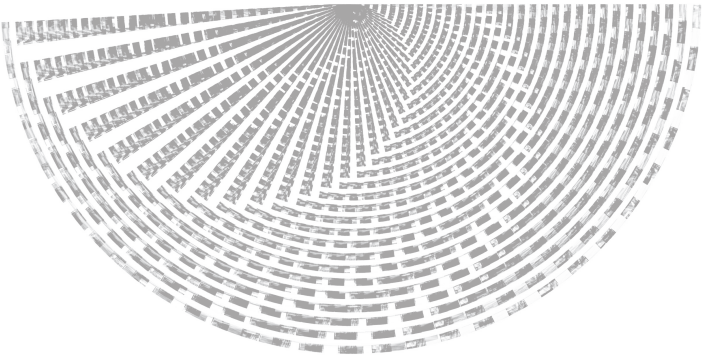
We are pleased with this final product of our more than two-year odyssey, and we owe a great debt to a number of colleagues, friends, and students whose feedback and assistance made this project possible. We thank Sean Petiya for the suggestions regarding the structure of the book; Rachel Chance, Jung Sun Oh, Karen Snow, and Kathy Wisser for reviewing the manuscript; Sean Dolan, Ryan Johnson, Amber Rodriguez, and David Todd for editorial assistance; and Jiangzhong Gu for his tireless support and assistance.





# PART I

## Fundamentals of Metadata







ONE

# Introduction

---

**WHETHER FOR EVERYDAY** life or for scientific endeavors, the search for information depends on two basic questions: “What is this ‘thing?’” and “How does this thing relate to other things?” In both library, archive, and museum (LAM) communities and in information industries, the use of metadata—best defined as the structured, encoded data that describe characteristics of information-bearing entities (i.e., *things*)—aids in the identification, discovery, assessment, and management of the described entities we seek (CC:DA 2000). From the early days of handwritten and printed catalogs and indexes to the modern days of web services and apps, the nature and goal of describing information-bearing entities have remained more or less unchanged. However, the methods and technologies have changed significantly. As the world around us becomes increasingly complex, and as we continue to experience information overload, organizing and managing information becomes a mission-critical task for organizations as well as individuals. This chapter will provide a context for metadata uses in our life and work and a brief history of the metadata movement. It will review fundamental concepts, including metadata types, categories of metadata standards, and metadata principles. Finally, it will present additional examples of metadata descriptions.

---

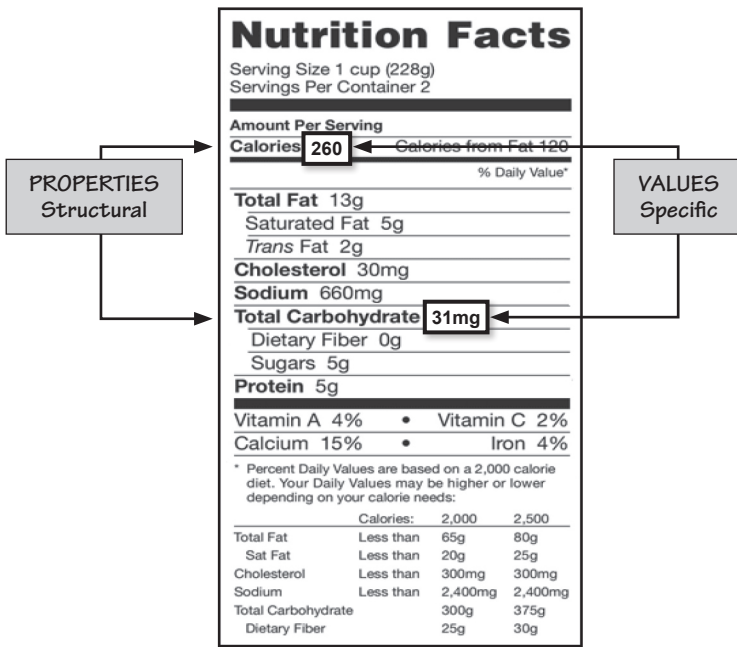
## 1.1 Background

The instances of structured data describing information-bearing entities can be as simple as the labels on food packages or bottled drinks that inform consumers about the ingredients and nutritional elements they contain. Such information can



be vitally important if someone is allergic to a certain ingredient in the product. A metadata description uses *property-value* paired statements to clearly describe the characteristics of a specific thing, as demonstrated by this food label (figure 1-1-1). A property specifies the meaning of a data value. For example, the property “calories” brings the meaning of the data value “260” in this food label. The structure for describing any food item, using certain properties contained in a unified format, is standardized by the United States Food and Drug Administration (FDA).

**FIGURE 1-1-1** A food label as a metadata description for a single item



Source: Based on the US Food and Drug Administration “Example of Graphic Enhancements” image ([www.fda.gov/ucm/groups/fdagov-public/documents/image/ucm070108.gif](http://www.fda.gov/ucm/groups/fdagov-public/documents/image/ucm070108.gif)).

Another scenario involves using a prescription drug information database to look for details about a drug’s intended effects and possible side effects. When searching for the information you need, you may wonder whether your findings are reliable. The metadata description about the database should be able to tell consumers who its maintainer is, how frequently the information is updated, and where the information on each drug comes from.

These scenarios are just two examples of using structured data to describe things. There are many different kinds of structured data with more complex

structures for use in our daily life, such as employee profiles, medical records, financial records, and academic records. These categories may share some characteristics, but also possess their own distinguishing ones. For instance, in the examples mentioned above, the unique information-bearing entities are employees, patients, customers, and students; they are described by the structured data specifically designed for them.

Besides these descriptions of our jobs, well-being, finance, and education, there are information sources we consult for various other needs. Say, for example, you are interested in learning about the landscape of New York State. In searching for the right map to satisfy this information need, the required information would include place names, geographic coverage, geographic coordinates, purpose, and type. Each of these map *properties*, when filled with proper data *values*, becomes a *statement*. In this scenario, statements about a map with an ID of “MAP12345” could include:

MAP12345	coverage:	“New York State”
MAP12345	type:	“Topographic Map”

These statements may match your information needs well. In such cases, individual statements describe the characteristics of a specific thing individually, and may not appear as a full *record* container. From this explanation, we can recognize that one *property-value* paired *statement* or more than one such *statements* make up a *description* about one resource (Powell et al. 2007). More importantly, these kinds of statements are machine-understandable and machine-processable when coded following certain syntax standards. In other words, the data become more actionable and inter-actionable in these forms (Coyle 2009). In today’s web-based environment, structured data describing the characteristics of things are exchanged and used in more flexible ways than a bounded *record*. In this book, we attempt to use the more general term *description* instead of *record* when referring to data that describes a specific thing. A description may contain one or more property-value paired *statements*, as the food label example shows.

Take one more scenario: when you look for information on “Leonardo da Vinci,” Google or another search engine will present you with a Knowledge Graph atop its first page of results (figure 1-1-2, left), whereas a Wikipedia page will provide an Infobox in its upper right-hand corner (figure 1-1-2, right). Both of these are good examples of the metadata *description* of the artist, which contains multiple *property-value* paired *statements* (e.g., for Leonardo da Vinci, Born: “April 15, 1452, Vinci, Italy”). They may lead you to other related things through the *values* (e.g., the actual birthplace of the artist, or the most notable artworks). The search engine also aggregates other metadata and includes them; the Knowledge

**FIGURE 1-1-2** “Leonardo da Vinci” Knowledge Graph from Google and “Leonardo da Vinci” Infobox from Wikipedia

The figure compares two information sources for Leonardo da Vinci. On the left is the Google Knowledge Graph, which provides a structured overview including his profession (Painter), a biographical summary, key dates (Born: April 15, 1452; Died: May 2, 1519), full name, period (High Renaissance), and parents. It also features a section for his artwork with thumbnails and titles like 'Mona Lisa' and 'The Vitruvian Man'. On the right is the Wikipedia Infobox, which lists biographical details such as birth name (Leonardo di ser Piero da Vinci), birth date and location, death date and location, and a list of notable works including 'Mona Lisa' and 'The Last Supper'. It also includes his style (High Renaissance) and a signature image.

Source: Based on screenshots of Google and Wikipedia displays. Captured on April 15, 2014.

Graph, for example, has information about other artists that people also search for (figure 1-1-2, left).

You may wonder where the descriptions are stored. In general, metadata descriptions can be embedded in digital files, managed in a local database, or stored in the cloud. Various syntaxes can be used to express them so that machines can process the data at the back-end and display them to you in human-readable formats at the front-end. Products and services built on or incorporating structured data can be delivered on web pages or through mobile apps suitable for various kinds of devices flexibly and dynamically.

Information-bearing entities come in a wide variety of types. They may be physical objects (including born-digital and non-digital objects), digitized surrogates of physical objects, or simply sets of information *about* digital or physical objects (e.g., any metadata dataset). Different information-bearing entities require different structured data to describe them so they can be identified, discovered, obtained, and used. The structured, encoded data describing the characteristics of these information-bearing entities are called “metadata.”

The rationale for metadata can be explained from various perspectives. As digital information becomes ubiquitous, we face constant challenges in managing all types and formats of digital information, and in providing tools that enable effective, timely, and precise information discovery from anywhere and at any time. Since computer technology began to come into widespread use in libraries and the information industry in the 1960s, libraries and information database producers have been the primary agents organizing and providing both information and the tools for searching and using information. Along with the rise of the Internet, web-based technologies have enabled mass information creation and publication through a low entry-barrier platform—anyone who is able to use a text or image capturer is now able to create digital objects and publish them directly onto the web. This has greatly democratized the publication and dissemination of information and has resulted in an exponential increase in the volume and complexities of digital resources. Individuals, organizations, communities, businesses, and governments now face the task of organizing the massive amounts of digital information and data in their information systems before they can effectively discover, locate, use, and reuse that information when needed.

At present, the Internet and the web have in many ways become the new library catalogs, indexing databases, dictionaries, encyclopedias, newspapers, schools, museums, entertainment centers, travel agencies, shopping centers, as well as many other things and places we used to only *physically* access. How do we find the sources we need and the places we want to go on the web or within an intranet? Through search engines, of course. But how do search engines take us to where we need to go in the world of digital information? What makes them work? Moreover, what makes them work *effectively*? The answers to these questions lie in the invisible hand of the efficient organization of information, which is embodied in metadata at the back-end of information services. When one enters a keyword into the text box of a search engine, or better yet, into a digital library's search box, the chances are that the keyword is one of thousands in an index. The index may be compiled from a metadata repository or generated by extracting words from full-text documents. The information displayed in the search results is a typical instance of metadata: it describes what a web document is (type) and what basic characteristics it has (title, link, time, and description). The Knowledge Graph display (refer to figure 1-1-2), which was added to Google in 2012 and was soon adopted by other search engines, provides a further step for information discovery by exposing and linking structured and detailed descriptions about *things* instead of merely giving a list of links and basic data about a web *document*.

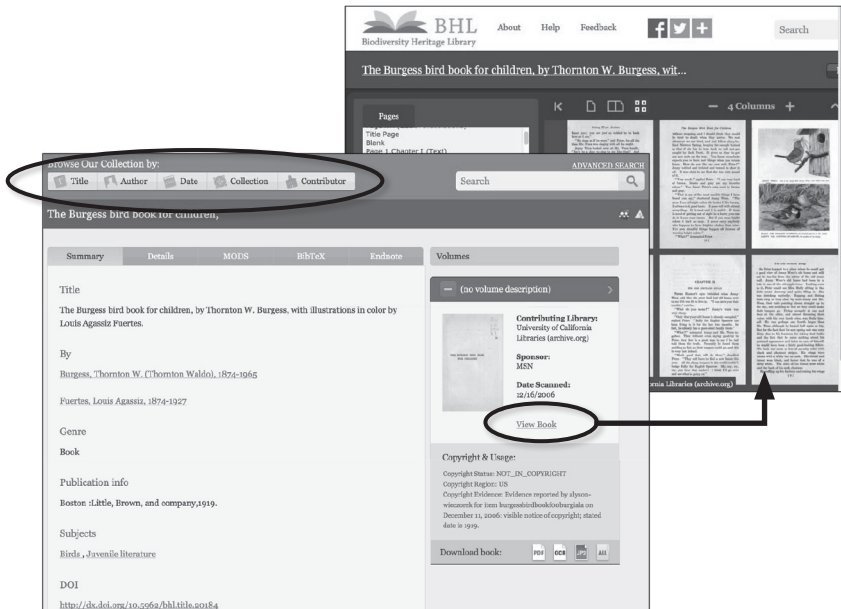
Metadata have been widely used in organizing and managing born-digital resources. Numerous examples can be found on social media sites, where simple metadata properties are captured automatically and tags are added when a digital

image, video, or blog is submitted. We can also find similar approaches in digital library systems and data repositories that require content contributors to enter metadata for digital resources to be submitted or added. Flickr ([www.flickr.com](http://www.flickr.com)), TED (Technology, Entertainment, and Design, [www.ted.com](http://www.ted.com)), the eCommons@Cornell ([ecommons.cornell.edu/](http://ecommons.cornell.edu/)), and the Knowledge Network for Biocomplexity ([knbc.ecoinformatics.org/index.jsp](http://knbc.ecoinformatics.org/index.jsp)) are just a handful of representatives of born-digital collections with metadata descriptions (see also the figures in section 1.7).

Metadata have also been widely used in organizing and managing digitized resources. As an example, consider *The Burgess Bird Book for Children*, which was published almost one hundred years ago. The digitization of this book likely would have generated a large number of image files. A metadata description will need to be created not only for the digitized book pages (i.e., the “digital surrogates”) but also for the relationship between the digitized and physical resources to tell users what the digital surrogate is for, who the creators and contributors of the original work are, and what the subject is, as shown in figure 1-1-3. The broad browsing categories located above the search area (title, author, date, collection) are made possible through the metadata hidden behind the front-end.

FIGURE 1-1-3

**A digitized book presented with browsing categories on the top on the first screen, and page navigation options on the left of the second screen**



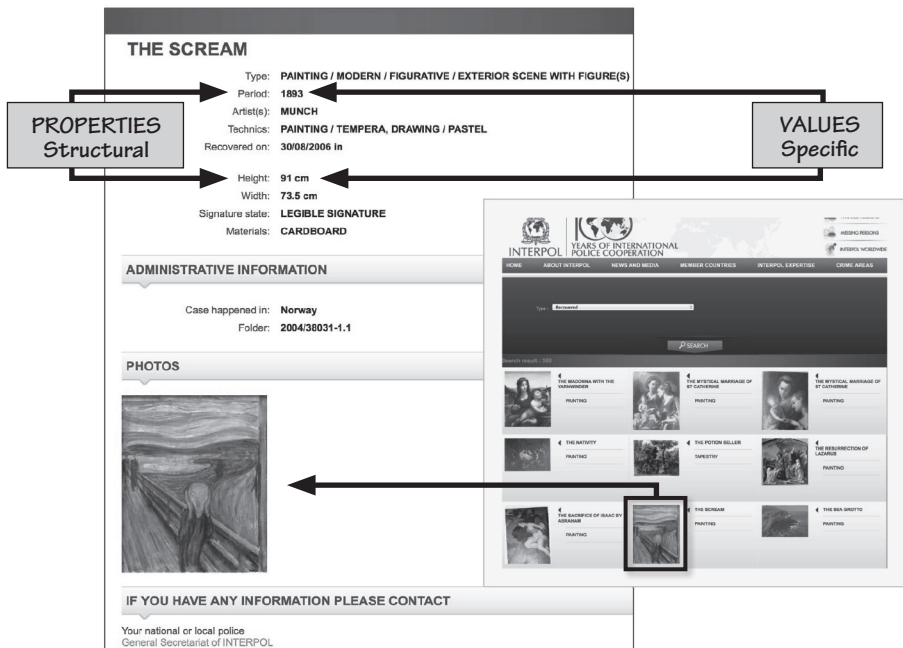
Source: Courtesy of Biodiversity Heritage Library ([www.biodiversitylibrary.org/bibliography/20184#/summary](http://www.biodiversitylibrary.org/bibliography/20184#/summary)).

Using metadata to collect and manage information about physical objects has been and continues to be a fundamental and powerful method. For example, a metadata standard called “Object ID,” released in 1997 ([archives.icom.museum/object-id/](http://archives.icom.museum/object-id/)) has been adopted by the United Nations Educational, Scientific and Cultural Organization (UNESCO); the International Council of Museums (ICON); INTERPOL (the world’s largest international police organization); and national governments around the world to help combat the illegal appropriation of art objects by facilitating the documentation of cultural property. This international standard aims at describing art, antiques, and antiquities through a set of properties that address the key questions in locating a stolen object: *Type of Object, Materials & Techniques, Measurements, Inscriptions & Markings, Distinguishing Features, Title, Subject, Date or Period, Maker, and Description*. With an attached photograph of the object, these metadata descriptions assist in registering and communicating the information necessary to identify stolen or missing objects throughout the world (figure 1-1-4).

Metadata are equally useful for the post-search processing and display of search results. A search in a digital collection often results in hundreds, even thousands,

FIGURE 1-1-4

**A recovered stolen art work documented in the INTERPOL online database.**  
**Source: Annotation on screenshots of INTERPOL online database**



Source: Courtesy of INTERPOL ([www.interpol.int/notice/search/woa/1030880](http://www.interpol.int/notice/search/woa/1030880); accessed 2014).

of hits. The user must then wade through that vast morass in order to select relevant information (or eventually to give up in frustration). The utilization of metadata, fortunately, allows the system to perform post-search processing and to present the results in categorized groups. Take, for example, Europeana—Europe’s integrated digital library, archive, and museum—which, as of 2014, aggregates data from over 2,000 institutions. In the next example, the post-search options (noted on the left in figure 1-1-5) allow users to refine search results by *media type*, *language of description*, *year*, *providing country*, *rights of use*, *copyright owner*, and *provider*. When clicking on an image, the user can view the details of metadata statements, which are essential to supporting all of these filtering options and to providing pathways to the location of the desired information. The search is also supported by the data values controlled in metadata descriptions, for instance, by standardized terms used to indicate the media types (image, text, sound), as in the example shown in figure 1-1-5.

Metadata descriptions of individual resources help group together similar resources based on user needs. In the Europeana example, metadata describing the artworks created by Raphael and other works about Raphael are aggregated from distributed datasets provided by Europeana member libraries, archives,

**FIGURE 1-1-5** Annotated screenshot of the filtering search result for the query “Raphael”

Filtering by relevant criteria, with the support of metadata descriptions at the back-end.

The screenshot shows the Europeana search interface. At the top left is the Europeana logo with the tagline 'think culture'. The search bar contains the query: "Raphael" OR ("Raffaello Santi") OR ("Raffaello de Urbino") OR ("Rafael S...". Below the search bar, it indicates 'Results per page: 48' and 'Results 1 - 48 of 1,331'.

The left sidebar, titled 'Matches for:', contains the following information:
 

- "Raphael" OR ("Raffaello Santi") OR ("Raffaello de Urbino") OR ("Rafael Sanzio de Urbino") OR ("Raffaello Sanzio") OR ("Raffaello Sanzio")
- By copyright: <http://creativecommons.org/publicdomain/mar/1.0/>
- Refine your results:**
  - ▼ Add more keywords (with a '+ Add' button)
  - ▼ By media type:
    - IMAGE (960)
    - TEXT (470)
    - SOUND (1)
  - ▶ By language of description
  - ▶ By year
  - ▶ By providing country
  - ▶ Can I use it?
  - ▶ By copyright
  - ▶ By provider
  - ▶ By data provider

The main results area displays a grid of search results, each with a thumbnail image and a title:
 

- Adam und Eva : (Wandgemälde i...)
- Auszug aus der Arche (Ost und West)
- Der Bau der Arche (Ost und West)
- Der Durchzug der Juden durchs R...
- Jeremias : Kirche S. Agostino (Menor...
- [Kopia aniolka z obrazu "Madonna Sykst...

Source: Based on Europeana #OnThisDay (<http://bit.ly/Raphael-Europeana>; captured on November 19, 2014).

and museums. This kind of function can be found in every online catalog service. Another case is a collection displayed on the visualized collection wall in a museum, such as the one in the Cleveland Museum of Art's Gallery One ([www.clevelandart.org/gallery-one/collection-wall](http://www.clevelandart.org/gallery-one/collection-wall)). Hundreds of digital images of cultural objects are routinely pushed to the wall according to *theme, color, culture, and work type*, and visitors can also choose one of the images on the wall and aggregate a set of objects around it according to *time period, theme, and gallery*. A visitor may create his or her own tour based on these aggregated items, download the tour to a mobile device with one touch, and go to other galleries to see the real objects, in all cases accompanied by detailed descriptions, annotations, and media guides embedded within the images representing the objects.

Together, these examples demonstrate that metadata are capable of performing the following functions:

- ▶ describing what resources are and what they are about, and organizing them according to controllable criteria
- ▶ allowing resources to be found by relevant criteria, aggregating similar resources, and providing pathways to the location of desired information
- ▶ facilitating metadata exchange and enabling interoperability
- ▶ providing digital identification and description for archiving and the preservation of resources (NISO 2004).

---

## 1.2 Definitions

The term “metadata” also appears in literature as “meta data” and “meta-data.” As with “data,” metadata can be either singular or plural. It is used as singular in the sense of a kind of data; however, in plural form, the term refers to things one can count (Turner, Moal, and Desnoyers 2003). Academic and research communities such as the geographic information system (GIS) community commonly adhere to the plural usage of the term (FGDC 2000).

The simplest definition for metadata is “data about data” (FGDC 2000; NISO 2004) or “information about information” (NISO 2004). Metadata as a concept has been used in different contexts to refer to information that is about specific things: for example, catalogs of published materials or museum objects, finding aids for archival materials, and indexes of journal articles are examples of metadata commonly seen in LAMs. Broadly speaking, metadata encapsulate the information that describes *any* information-bearing entity. However, this preliminary definition does not convey the full connotation of the term. The American Library



Association Committee on Cataloging: Description and Access (CC:DA) Task Force on Metadata defines metadata as structured, encoded data that describe the characteristics of information-bearing entities and, as such, enable functions for identifying, discovering, assessing, and managing the entities (CC:DA 2000).

Metadata such as those used for publications and scientific data traditionally have been used in research and learning, and are more familiar to users. However, metadata exist not only in the traditional bibliographic data universe, but also in our daily lives. Metadata may appear in forms and places many people would not notice. As explained at the beginning of this chapter, labels that are fixed on food packages can be considered as nutrition and ingredient metadata; similarly, business directories contain the identity, contact, product, and service data about enterprises. Both are perfect examples of nonbibliographic metadata. When you take digital photos or create audio or video files, values of metadata properties such as time stamp, resolution, file size, and color scheme are captured automatically.

As the research and application of metadata evolved, its definition was refined to be more specific and explicit: metadata became “structured information that describes, explains, locates, or otherwise makes it easier to retrieve, use, or manage an information resource” (NISO 2004: 1), and it became “data associated with either an information system or an information object for purposes of description, administration, legal requirements, technical functionality, use and usage, and preservation” (DCMI Glossary 2005). Typically, individual metadata statements and packed descriptions or records are often referred to as “metadata” of a thing. In this book, we attempt to use more precise phrases for the many related concepts that are sometimes mistakenly conflated.

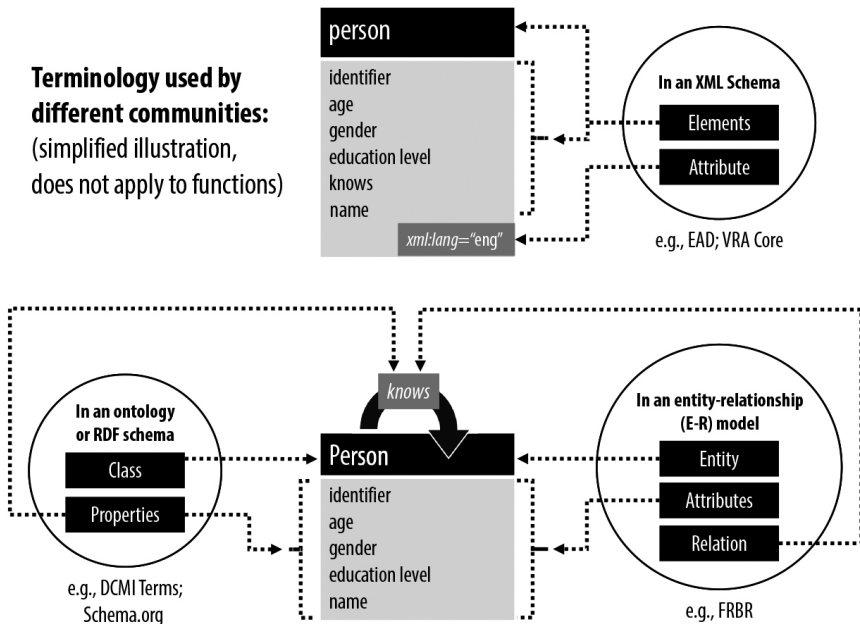
A large number of metadata vocabulary standards have been developed or proposed by communities in different domains. A standard is a formal document that establishes uniform criteria, methods, processes, and practices. A key component in these standards is the *element set*, which defines the structure and semantics of elements. For example, the international standard *Dublin Core Metadata Element Set* (DCMES, or “Dublin Core”) defines 15 core elements that are used to describe information resources. The metadata community has used a number of terms in different contexts to refer to metadata *element sets*, in addition to the machine-processable *schemas* through which they are encoded. The terms “metadata standards,” “element sets,” “metadata vocabularies,” “metadata schemas,” “property vocabularies,” “data dictionaries,” and “metadata dictionaries” have subtle differences, but are used interchangeably in literature. These phrases will be explained and differentiated in the book as necessary.

Different communities also refer to the same metadata components with different terminology. For instance, in computer science metadata is information

about database objects and/or program objects, such as tables and stored procedures. A database designer may call metadata elements “data fields” instead of “elements.” Other terms such as “properties” and “attributes” are also frequently used in metadata literature, but their meanings change as the context in which these terms are used changes. An ontological view of metadata, for example, treats Person as a *class* with “name,” “age,” “gender,” and “education level” as *properties* of the Person class. Meanwhile, an entity-relationship model calls Person an *entity* with “name,” “age,” “gender,” and “education level” as *attributes* of the Person entity. In a schema encoded with an encoding language (e.g., Extensible Markup Language [XML]), an *attribute* has yet another meaning: it is an integral part of an XML *element* that defines the features of that element. For instance, in a “name” element encoded in XML as <name xml:lang=“eng”> ABC </name>, the attribute “xml:lang” specifies the language of the appellation (figure 1-2-1).

FIGURE 1-2-1

### A simplified explanation of the variations in terminology found in different communities



The diverse use of this terminology is an indication that the concept of “metadata” draws on a wide variety of research fields and requires background knowledge and skills in databases, ontological modeling, knowledge organization

systems, and markup languages in order to thoroughly understand and properly apply it.

---

### 1.3 A Brief History

The organization of objects and phenomena into either classes or sets of relationships is one way in which humans communicate. Memory institutions such as LAMs traditionally govern their resource organization tasks by rules or standards. Present-day library cataloging practices can all be traced back to the nineteenth century when (1) Antonio Panizzi and his assistants at the British Museum created and implemented *Ninety-One Cataloguing Rules* (Panizzi 1841), (2) Charles Coffin Jewett started the task of building the Smithsonian Institution's library by soliciting catalogs from prominent libraries and mechanically duplicating individual entries in the 1850s, and (3) Charles Ammi Cutter issued his *Rules for a Printed Dictionary Catalog* (Cutter, 1875) as Part II of a special report, *Public Libraries in the United States of America: Their History, Condition, and Management*, by the United States Bureau of Education (1876). The content representation in library bibliographic catalogs has been guided by semantically rich classification schemes and lists of subject headings, such as *Dewey Decimal Classification* (DDC, first published in the United States by Melvil Dewey in 1876) and the *Library of Congress Subject Headings* (LCSH, first printed in the 1910s). Since the 1960s, the creation of the library community's bibliographic records has been governed by the *Anglo-American Cataloguing Rules, Revised, Second Edition* (AACR2)—which is being replaced by *Resource Description and Access* (RDA), released in 2010—and by the MARC (*MAchine-Readable Cataloging*) format. “Just as standardization of catalog card sizes enabled interchange of catalog records, so has MARC made possible interchange of machine readable catalog records” (Kilgour 1997, 225).

In the pre-Internet era, the information objects being organized in libraries were primarily physical. In other words, they are containers or packages of information in forms such as books, journals, music or narrative recordings, movies, and technical reports. Creating cataloging descriptions for these information objects requires a significant amount of human intelligence and labor, not only because of the physical nature of information objects, but also due to the complexities and rigidity of rules and standards. Pre-Internet cataloging played a significant role in helping users to find what they needed *and* to determine if an item was located in the stacks, as well as whether collocated items of the same subject area made a trip into the musty rows of holdings worthwhile. The

purposes of pre-Internet cataloging were twofold: (1) to provide rich bibliographic descriptions and relationships between and among data of heterogeneous items, and (2) to facilitate the sharing of these bibliographic data across library boundaries. All of these factors made it very difficult, if not impossible, for computer programs to take over metadata creation tasks from human catalogers. However, the level of sophistication and maturity of information technology also played an important role in pre-Internet cataloging. Fred Kilgour, the first president of OCLC, described the period from 1954 to 1970—after the initiation of computerization but before online systems were in effect—as roughly split between the computerization of user-oriented subject information retrieval and the computerization of library-oriented procedures (Kilgour 1970). With the innovations of information and communication technologies, the library community advanced to online systems; this allowed the implementation of online union catalogs, shared cataloging systems, online interlibrary loan management systems, remote catalog access by readers, and circulation control (Kilgour 1997; Rayward 2002). The OCLC Online Library Computer Center (originally named the Ohio College Library Center when it was established in 1967) remains the central library service for descriptive and technical cataloging, and provides bibliographic and other formatted packages for the Online Public Access Catalogs (OPAC) that provide users access to library collections.

While the last half-century has proven the importance of library catalogs to society, it also has revealed the emerging demands directly associated with changing information technologies, especially in the Internet era. AACR2 and MARC have done a meritorious job in accomplishing the two basic purposes discussed above; nevertheless, they have fallen short on several important fronts in the environment of Internet-based resource descriptions, for example, for the purposes of managing digital rights, preservation of digital objects, and evaluation of resources based on authenticity, user profile, and other more specialized emerging description needs. Meanwhile, starting from the early 1990s, distributed repositories on the Internet experienced an exponential growth. More importantly, the repositories were contributed by communities of LAMs and beyond. The Internet-based information explosion phenomenon called for mechanisms of description, authentication, and management, which encouraged new guidelines and architectures to be developed by different communities.

Metadata development in the Internet era took off in the first half of the 1990s. Caplan (2000) described the metadata movement as a blooming garden, traversed by crosswalks, atop a steep and rocky road. There were several parallel development areas in metadata at that time. The scientific community began to look for solutions to organize the rapidly increasing amount of scientific data,

which prompted the introduction of the *Content Standards for Digital Geospatial Metadata* (CSDGM) in 1992 by the Federal Geographic Data Committee (FGDC). In the humanities community, the Text Encoding Initiative (TEI), an international organization founded in 1987, released the first version of its *Guidelines for Electronic Text Encoding and Interchange* (TEI Guidelines) in 1994. As an international and interdisciplinary standard, the TEI Guidelines focuses primarily (though not exclusively) on the encoding of documents in the humanities and social sciences, and in particular on the representation of primary source materials for research and analysis. It is more a markup schema than a metadata vocabulary. The Art Information Task Force (AITF), funded by the J. Paul Getty Trust and a two-year matching grant from the National Endowment for the Humanities (NEH) to the College Art Association (CAA), was initiated in the early 1990s, and published *Categories for the Description of Works of Art* (CDWA) in 1996.

The library community also took action to develop metadata standards as a solution to resource description and discovery problems. OCLC initiated a project in 1994 to experiment with cataloging web resources using AACR2 and MARC formats. More than 200 volunteer librarians created over 2,500 records for Internet resources; this became the precursor of the Metadata Workshop that was held in 1995 at OCLC in Dublin, Ohio (Weibel et al. 1995). The Dublin Core was born at this historically significant workshop. Following the first Dublin Core workshop, the metadata movement soon spread rapidly to other continents and throughout research, educational, and governmental institutions, as well as businesses and many other types of organizations.

The following are some of the many standards for metadata structures that have been developed from the 1990s to 2005, listed chronologically according to development and formal release (a more complete list of metadata standards is available at this book's accompanying web site):

- ▶ IPTC Photo Metadata Standards
- ▶ Content Standards for Digital Geospatial Metadata (CSDGM)
- ▶ Guidelines for Electronic Text Encoding and Interchange (TEI Guidelines)
- ▶ Categories for the Description of Works of Art (CDWA)
- ▶ Encoded Archival Description (EAD)
- ▶ Dublin Core Metadata Element Set (DCMES, or DC)
- ▶ Darwin Core (for sharing information about biological diversity)
- ▶ Visual Resources Association Core Categories (VRA Core)
- ▶ ONline Information eXchange (ONIX)
- ▶ Learning Object Metadata (LOM)

- ▶ Metadata Object Description Schema (MODS)
- ▶ MPEG-7 (the standard for description and search of audio and visual content)
- ▶ Friend of a Friend (FOAF)
- ▶ PREMIS: PREservation Metadata Implementation Strategies
- ▶ Public Broadcasting Metadata Dictionary (PBCore)

There was a great increase in the number of metadata projects during the late 1990s. The DCMI (Dublin Core Metadata Initiative) web site formerly maintained a list of many metadata experiments and projects from around the world. Publications about metadata dominated Internet forums, professional journals, and conferences in a variety of disciplines. The main reason for this proliferation is that, in point of fact, there is no limit for the type or amount of resources that metadata can describe, nor are there any limits to the number of overlapping metadata standards for any type of resource or subject domain.

The metadata standards listed above are designed and used for data structures. Despite their size difference (from 15 to over 500 elements, with the majority comprising between 100 and 300), they profoundly influenced the later development of metadata vocabularies. Since 2003, newly developed metadata vocabularies from many communities have mainly derived from, or have been built on, these standards, as have the initiation and launching of more and more digital collection projects, services, and aggregators in LAMs and beyond. As of early 2015, a service called Linked Open Vocabularies ([lov.okfn.org/dataset/lov/](http://lov.okfn.org/dataset/lov/)) has registered nearly 470 metadata vocabularies (including element sets and ontologies) that are published with Semantic Web languages and are used by or for datasets in the Linked Data Cloud (LOD). It is noteworthy that almost all of these have reused and combined the elements or properties defined by multiple existing namespaces. (A namespace is a collection of names that is identified by a Uniform Resource Identifiers [URI] reference. For instance, the URI “<http://purl.org/dc/elements/1.1/>” refers to the Dublin Core Metadata Element Set [DCMES], Version 1.1.) Opening any metadata schema that surfaced after 2005, shows that multiple sources’ namespaces are listed.

A giant metadata vocabulary, Schema.org, appeared online ([schema.org/](http://schema.org/)) in 2011. This vocabulary, created by major search engines such as Bing, Google, Yahoo!, and Yandex, aims to provide many schemas under one namespace so that webmasters can describe and expose web sites of any kind to search engines. Schema.org is still being extended. Its impact can be seen on an increasing number of web site products across almost all domains and areas. More importantly, it accelerated metadata parallel production using non-LAM formats or languages,

as demonstrated by the addition of Schema.org markup on the existing WorldCat records by OCLC in 2012 (to be discussed in chapter 2).

The last two decades of metadata development have witnessed a continual expansion and evolution of metadata research and practices at almost all levels and in almost all disciplines. In addition to many repositories and catalogs provided by institutions, consortia, and networks, large-scale national and international digital libraries have been established. The most visible ones are the United States National Science Digital Library (NSDL), initiated at the beginning of the twenty-first century; Europeana, established in 2008; and the Digital Public Library of America (DPLA), launched in 2013. These digital libraries aggregate metadata from many data providers and serve them through their unique platforms and portals (see figure 1-1-5 for an example from Europeana). In recent years, the concepts and technologies of the Semantic Web, Linked Open Data, and Big Data have changed the world. Metadata are receiving even wider and greater attention than ever before. LAMs were among the first to publish their data (including bibliographic data, name authorities, and controlled vocabularies) as Linked Data. They are also taking steps to go outside their own metadata silos by aligning and directly using non-LAM metadata standards such as Schema.org. In the meantime, the mechanism of structured and encoded data has become widely accepted by society. Data is considered the “new oil” today, in terms of more than only its monetary value. “In its raw form, data is just like crude oil; it needs to be refined and processed in order to generate real value. Data has to be cleaned, transformed, and analyzed to unlock its hidden potential” (TiECON East, [www.tieconeast.org/2014/big-data-analytics](http://www.tieconeast.org/2014/big-data-analytics)). In the digital humanities literature, metadata produced with the guidelines of TEI (Text Encoding and Interchange) and other standards are considered to be “smart data” (Schöch 2013) that reflect the important processes of organizing and integrating structured, semi-structured, and unstructured data that can make the Big Data smarter. These emerging concepts bring new dimensions to metadata research and implementations. (Chapter 9 of this book is dedicated to the metadata research landscape.)

---

## 1.4 Types and Functions

Defining the types of metadata is both contextual and dependent upon application domains. In 1998, the J. Paul Getty Trust’s Getty Information Institute (renamed the Getty Research Institute in 1999) published *Introduction to Metadata: Pathways to Digital Information* and made an electronic version available online (Baca 2000–2008). Its chapter entitled “Setting the Stage” identifies five types of metadata and their functions:

- ▶ *Administrative metadata.* Metadata used in managing and administering collections and information resources (examples include acquisition information, rights and reproduction tracking, legal access requirements, and location information).
- ▶ *Descriptive metadata.* Metadata used to identify and describe collections and related information resources (examples include cataloging records, finding aids, specialized indexes, and curatorial information).
- ▶ *Preservation metadata.* Metadata related to the preservation management of collections and information resources (examples include documentation of the physical condition of resources, actions taken to preserve physical and digital versions of resources (e.g., data refreshing and migration), and changes occurring during digitization or preservation).
- ▶ *Technical metadata.* Metadata related to how a system functions or metadata behaves (examples include information about hardware and software requirements, technical digitization (such as formats, compression ratios, and scaling routines), and authentication and security data (e.g., encryption keys and passwords).
- ▶ *Use metadata.* Metadata related to the level and type of use of collections and information resources (examples include circulation records, physical and digital exhibition records, use, reuse, search, and user tracking) (Gilliland 2008).

A booklet published by the National Information Standards Organization (NISO) categorizes metadata types into three summative groupings. Note the grouping of those types (technical, rights management, and preservation metadata) under “administrative” and the addition of “structural” metadata are different when compared with the categories listed above by Gilliland:

- ▶ *Descriptive metadata* describes a resource for purposes such as discovery and identification.
- ▶ *Structural metadata* indicates how compound objects are put together.
- ▶ *Administrative metadata* provides information to help manage a resource, and includes technical, rights management, and preservation metadata (NISO 2004, 1).

To explain these major types of metadata, we’ll begin by reviewing the most prevalent and traditional type, *descriptive metadata*. Describing a publication, for example, means capturing essential information such as title, creator, keywords, date of creation or publication, and type of resource. This process usually follows certain standards that control which data need to be captured and how that data



should be entered into a computer-readable format. Cataloging records, finding aids, indexes, and curatorial information are all primarily descriptive.

Several important characteristics of metadata make them distinct from traditional cataloging products. The nature of library cataloging has changed significantly as the ever-growing diversity of nontraditional formats of resources entered into library collections. Contemporary library metadata is increasingly concerned with digital resources, as reflected in RDA. In addition to managing new, nontraditional formats of resources coming from formal publication distribution channels, libraries also assume the role of the host or assistant of institutional repositories of documents and datasets. Various types of libraries are also taking on the task of collecting web sites documenting important events. The best example of this is the Tweet Archive of the Library of Congress, in which tweets from the debut of Twitter in 2006 to the year 2010 are being organized into hourly files (Library of Congress 2013). Whereas the change of containers along with technological advances, such as the shift in music formats from records to cassettes to CDs to mp3 files, requires new ways to describe things, a more fundamental change involves the demand for discovery and access of the basic units in these containers (e.g., the pieces of music themselves). Such demand can be seen for all other types of resources, especially academic publications, news, and datasets. Organizing these resources and providing services for retrieving and using them is a complex process that requires various types of metadata for different purposes and functions. As Gilliland (2008) outlined, all information objects, regardless of the physical or intellectual form they take, have three features: (1) content (what the object contains or is about), (2) context (the who, what, why, where, and how aspects associated with the object's creation), and (3) structure (the formal set of associations within or among individual information objects)—all of which can and should be reflected through metadata. These are far more than *descriptive*.

Rights management metadata is the most important type of *administrative metadata*. Digital resources can be easily accessed, copied, modified, or deleted, which in turn can trigger violations of copyright, access permissions, and licensing rules. Metadata must record the rights information of a resource, which will be used for administrative tasks and management. *Technical metadata* is critical for the life cycle of any digital resource. In addition to describing the resource's characteristics, metadata may also be used to describe the platform and software required to render a digital object. Difficulties may arise in preserving digital resources for use by future generations of software (think of those materials stored on floppy disks or run on obsolete systems), but the technical metadata will provide the clues about their characteristics. Moreover, *meta-metadata* fulfills the *administrative* function of metadata descriptions themselves: a meta-metadata

description provides information on when and how a metadata description was created (by whom, from where, and with what standards), what technical details it contains, and who has access privileges to the stored metadata.

Using various types of metadata in one example, let's take a look at the digitized book identified in figure 1-1-3. In the simplest case, each page of the book is scanned into an image, which may further create multiple files (images and text pages) for different purposes. Besides the *descriptive* metadata description of the whole, the digitized pages must be clearly organized in the sequence of chapters, sections, and pages for human readers' consumption. Thus, *structural* metadata is needed for maintaining the correct sequence for this digital resource. Because a digitized resource often results in multiple digital files for each page, *technical* metadata needs to record the information about how the digitized pages were created and what auto or manual treatment was done. Some of the technical metadata statements, such as the resolution, color mode, bit depth, file size, dimensions, compression ratio, and file format of each image, can be automatically generated. *Rights management* metadata takes care of the intellectual property rights of the book in both its original format and in the digitized form. Thumbnails, web-ready images, and low-resolution images may have rights and licenses that are different from those of the high-resolution images. The *preservation* metadata would document the physical conditions of the whole and parts of the physical resource, plus the history of all actions performed on the original and digitized resources. The *use metadata* would report the information on search, circulation, and exhibition of the book. Together, these types of metadata will offer functions necessary for organizing and managing electronic resources, facilitating interoperability, assisting the digital resources' authentication, archiving and preservation, and enabling efficient resource discovery and use.

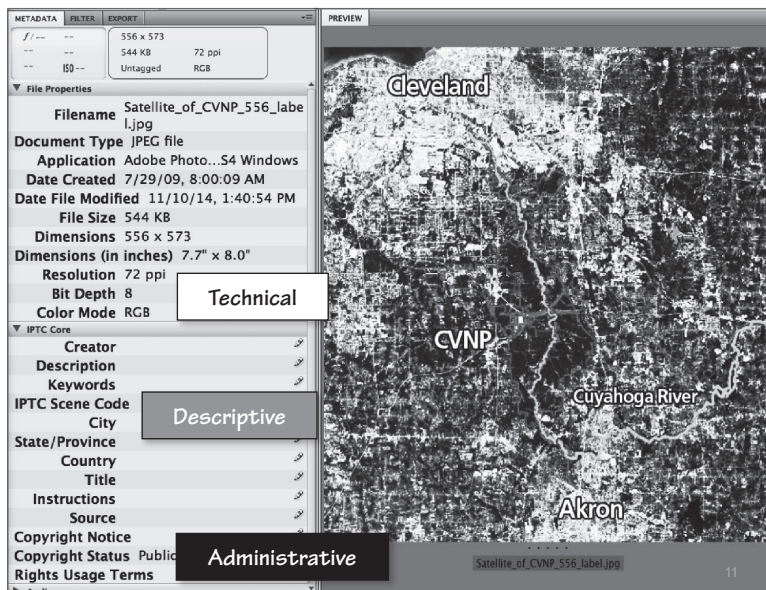
*Use metadata* can be learned. Publishers, social media, and marketing services have been employing such data collected based on usage, for example, the number of times the published or posted items have been viewed, downloaded, discussed, reviewed, recommended, shared, or cited. Amazon's recommendation for a book uses data such as *frequently bought together*, *customers who bought this item also bought*, and *average customer review* (star ranging). Metadata descriptions accompanying the item usually not only show what this thing *is* but also reveal how this thing *is related to* other things.

Metadata creation can benefit from modulated processes because certain types of metadata can be generated by different providers or even by software. For example, administrative metadata on rights and licenses can be batch-processed with default values and managed by a different unit from the one that produces descriptive metadata. Technical metadata may be directly generated by computer programs. To explain this, we will use a real-life example. Assume you need to

add a satellite image to a web site of a national park. You found a satellite image showing the Cuyahoga Valley National Park between Akron and Cleveland, Ohio, and the Cuyahoga River flowing north through the park to Lake Erie. Great! But some questions must be answered before using this image: Is the quality of the image good enough to be put on the web? What is its resolution? What is its size? Will it be too large to download to a smart phone or too small to zoom-in? The *technical metadata* will be useful in answering these questions. In the screenshot shown in figure 1-4-1, you can see that when opening an image file with the Adobe Bridge software, the answers to these questions are provided by technical metadata statements that were automatically captured by the software (see the “File Properties” section at the left of the figure). Other descriptive information can be manually added, and rights statements can be batch-processed or individually added. The descriptive metadata elements follow the IPTC Core (see the “Descriptive” section on the left side of the figure), a photo metadata standard developed by the International Press Telecommunications Council (IPTC). This leads us to the next section, which discusses metadata standards.

FIGURE 1-4-1

### A screenshot of a metadata instance for a satellite map image, showing automatically captured file properties on the Adobe Bridge metadata template

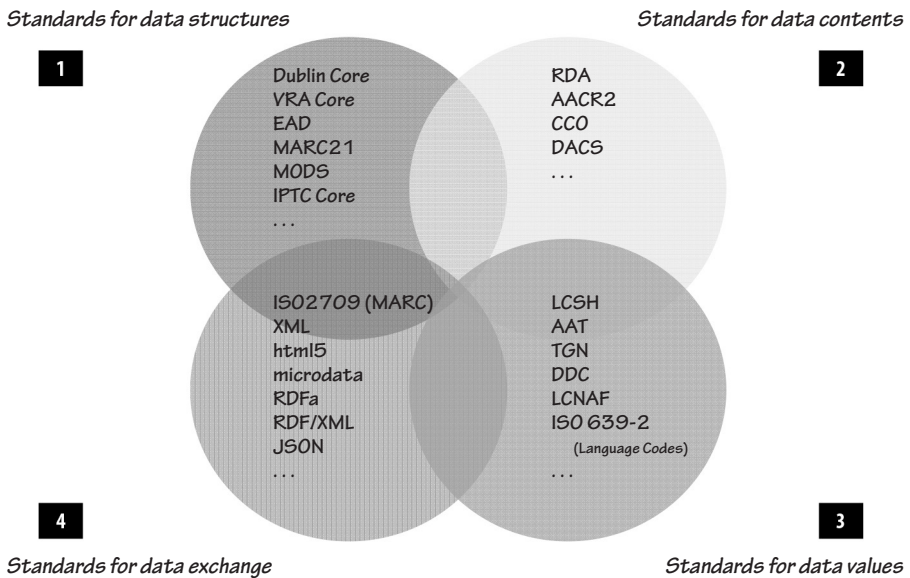


Source: Original satellite image credited to the National Park Service ([www.nps.gov/cuva/forkids/park-basics.htm](http://www.nps.gov/cuva/forkids/park-basics.htm)).

## 1.5 Standards

Metadata-related standards are formal documents that establish uniform criteria, methods, processes, and practices. They have been created for specific purposes: to guide the design, creation, and implementation of data structures, data values, data contents, and data exchange in an efficient and consistent manner. Based on the purpose for which a metadata standard is developed, metadata standards can be divided into four categories, in the context of LAMs (figure 1-5-1).

**FIGURE 1-5-1**  
**Types of metadata standards**



1. Standards for *data structures*. Usually referred to as “element sets” or “metadata vocabularies,” these standards define data structures and semantics. The most well-known and widely used is the Dublin Core Metadata Element Set (DCMES), which is intended for general use. Each community follows a set of standards that are developed for its specialized description needs: the MARC family (MARC, MARCXML, MODS) has the library community as its primary adopter, EAD remains the dominant standard in archival descriptions,

and CDWA and VRA Core are designed for the museum and visual resources communities. The IPTC Core standard (discussed in the section 1.4 and illustrated in figure 1-4-1) was developed by the International Press Telecommunications Council (IPTC), and focuses on news and stock photos. Regardless of the purpose for which the standard is developed, elements in a metadata standard can be arranged in a flat, nested, or modular style. Chapter 2 will discuss metadata element structures in greater detail.

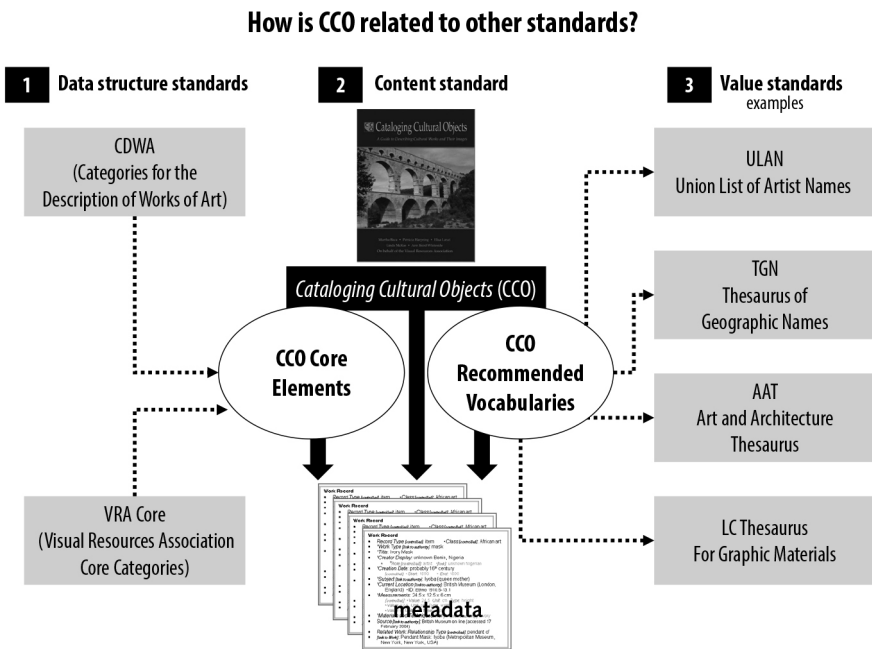
2. Standards for *data contents*. Data content standards are created to guide the practices of metadata generation and cataloging. Examples of guidelines or rules used in LAMs include: (a) AACR2, which was superseded by RDA; (b) *Describing Archives: A Content Standard* (DACS); and (c) *Cataloging Cultural Objects: A Guide to Describing Cultural Works and Their Images* (CCO). An examination of CCO shows that it covers the rules for cataloging core data elements needed to describe cultural objects. The details for each element encompass all possible scenarios, for example, in documenting a creator of a work (e.g., the person responsible for a historical architectural work), who could be known by name, or be anonymous or unknown, and whose time period and roles may be ambiguous or uncertain. Guidelines about cataloging levels, treating relationships between a work and its images, terminology sources, and rules for vocabularies and authority control are just a small set of examples of contents in this standard.
3. Standards for *data values*. These are different kinds of knowledge organization systems, generally referred to as “value vocabularies,” and sometimes as “value encoding schemes,” in a metadata vocabulary specification. These include, but are not limited to: (a) controlled term lists (e.g., the *MARC Code List for Relators*); (b) classification schemes (e.g., DDC); (c) thesauri (e.g., *Art and Architecture Thesaurus* [AAT] and *Thesaurus of Geographic Names* [TGN]); (d) authority files (e.g., *LC Name Authority File* [LCNAF] and *Virtual International Authority File* [VIAF]); and (e) lists of subject headings (e.g., *LC Subject Headings* [LCSH]). Some of these include thousands of terms representing concepts and the semantic relationships between and among the concepts. However, there are also many more that only contain a small number of terms or codes. The recommendations for using these are usually spelled out in the texts of both types of standards discussed above.

4. Standards for *data exchange*. These are referred to as different “formats” when discussed in the context of data exchange and communication. The *MARC 21 Format for Bibliographic Data* is an example of an international standard, ISO 2709 — *Information and documentation — Format for information exchange*. Most metadata standards for data structures (e.g., MARCXML, MODS, EAD, and VRA Core) now use a generalized markup language, Extensible Markup Language (XML), in order to express their element sets and create their metadata schemas. Since the later 2000s, various serialization formats for Resource Description Framework (RDF) have also been increasingly used by metadata vocabularies. For example, the DCMI Metadata Terms (DC Terms) vocabulary is available as different RDF schemas.

A metadata standard may incorporate both data structures and exchange format in one document, as EAD does. An element set may cover detailed content rules, as do CDWA and VRA Core. This is why figure 1-5-1 displays overlapping areas. Using CCO as an example, the next illustration (figure 1-5-2) shows how various types of standards for cataloging cultural objects are related to each other.

FIGURE 1-5-2

### Illustration of the relationships among various types of standards, using CCO as an example



What about non-LAMs? Consider the example of a nutrition facts label that was used at the beginning of this chapter (refer to figure 1-1-1). Imagine what it would look like without standards from FDA regarding (1) the structure governing the properties that must appear on any such label (e.g., *Cholesterol*, *Sugar*, *Total Fat*); (2) the guidelines for providing the values (e.g., the definition of *Total Fat*, the values to be used for calculating *Daily Values*, the decisions of whether to round a value of 47 calories up to 50 or down to 45); (3) the controlled value terms (e.g., the major food allergens); and (4) the label formats and graphics requirements (e.g., multilingual labeling, colors, and font sizes for ingredient lists). This example reminds us that metadata creation is guided or ruled by various types of standards for data structures, data contents, data values, and data exchange. It also tells us that a food label is not as simple as it looks!

A good understanding of these categories is important for using this book, which is organized mainly around the standards for data structures. Other types of standards will be introduced in the context of metadata descriptions (in relation to data content standards), value spaces (in relation to data value standards), and schema encoding (in relation to data exchange standards).

---

## 1.6 Principles

The Dublin Core Metadata Initiative (DCMI) was motivated by the guiding principle of producing a metadata element set that would be simple enough for creating and maintaining metadata records that the elements would conform to existing and emerging standards on an international scale, and that the elements would be interoperable among collections and indexing systems (Weibel and Hakala 1998). The initial Dublin Core workshop valued simplicity, so that “ordinary” users would be able to formulate descriptive records based on a relatively simple schema of fifteen free-text elements (Lagoze 2000). The term “simplicity” has two implications: first, users should be able to simply take only those data elements (i.e., properties) necessary, thus maintaining a minimum set of data elements for easy deployment, and second, users should be able to introduce new elements and constraints for localized description needs without significant structural and semantic changes. As metadata development has evolved, its early requirements of simplicity, extensibility, and interoperability have been extended and elucidated to become a more inclusive and refined set of principles.

Principles are those concepts judged to be common to all domains of metadata and which might inform the design of any metadata schema or application. Practicalities are the rules of thumb, constraints, and infrastructure issues that emerge from bringing theory into practice in the form of useful and sustainable systems. (Duval et al. 2002, [www.dlib.org/dlib/april02/weibel/04weibel.html](http://www.dlib.org/dlib/april02/weibel/04weibel.html))

The following set of primary principles for the construction of ideal metadata are derived from Duval et al. (2002), Dempsey and Weibel (1996), Moen (2001), and NISO 2004. The term *metadata schema* is often used interchangeably with *metadata standard*. In metadata-related literature, the word “schema” usually refers to an entire element set as well as the encoding of the elements and structure using a standard language.

- ▶ *Modularity* means to build metadata into blocks, so that data elements, vocabularies, and other building blocks in different metadata schemas may be assembled into new schemas in a syntactically and semantically interoperable way (Duval et al. 2002). Different metadata modules (e.g., for discovery, rights management, geospatial and temporal coverage, provenance, or preservation) expressed in a common syntactic idiom (such as an XML schema, a Web Ontology Language [OWL] ontology, or an application profile) should be able to be combined in compound schemas as needed to embody the functionality of each constituent.
- ▶ *Extensibility* is generally understood in at least two senses: (1) the ability of a metadata schema to offer a core set of elements that will unify different models of resource description, and (2) the ability to link a simple metadata record to a richer, more complex description of resources (Dempsey and Weibel 1996). Metadata systems must allow for extensions so that specific needs of a given application can be accommodated (Duval et al. 2002). Such extensions could include the addition of new elements and/or sub-elements to the existing ones in a schema.
- ▶ *Refinement* aims at a precision of detail that determines decisions about how much description a schema should require. These include: (1) refining or creating more specific the meaning of an element (e.g., various kinds of dates), and (2) specifying particular value encoding schemes (e.g., a controlled vocabulary) to be used for a given element (Duval et al. 2002).
- ▶ *Multilingualism* focuses on aspects of language and culture, and requires that a designer take into account linguistic and cultural diversity when adopting metadata architecture (Duval et al. 2002).



- ▶ *Interoperability* is defined as “the ability of multiple systems with different hardware and software platforms, data structures, and interfaces to exchange data with minimal loss of content and functionality” (NISO 2004, 2). Interoperability issues must be addressed not only at the syntactic and functional levels, but also at the semantic level (Moen 2001). This principle overlaps with other principles and deserves special emphasis (see chapter 8).

These principles address the issues one may encounter in one form or another in both metadata schema design and description generation processes. As principles, they represent a set of basics that all metadata designers should take into consideration when conceiving a metadata project. These principles also have a direct effect on how to implement metadata projects and simultaneously make them both sustainable for long-term utilization and preservation as well as interoperable for sharing and reuse. The metadata principles discussed in this section can be seen in the practices of many examples used throughout the book.

---

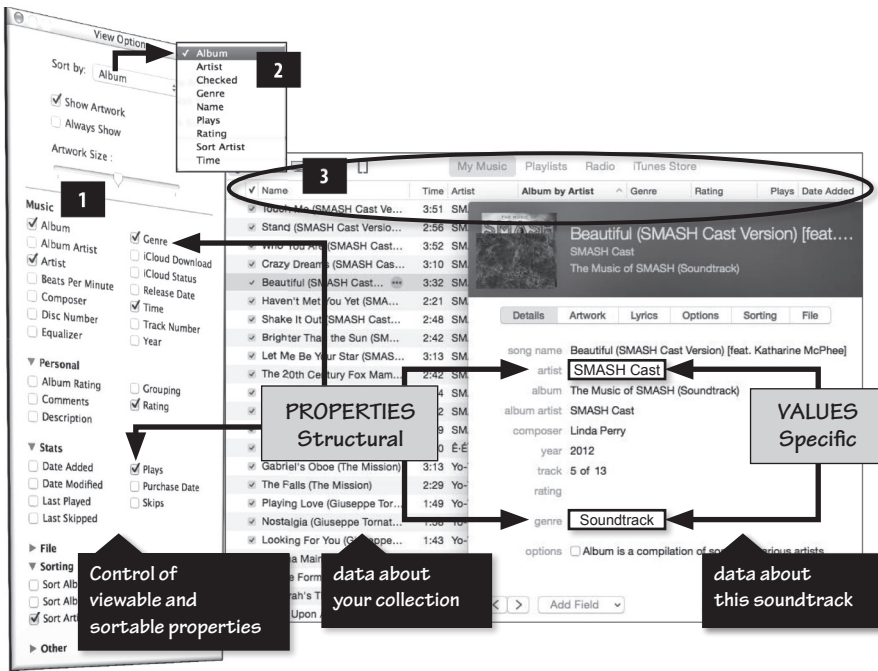
## 1.7 Examples of Metadata Descriptions

A metadata description may be in the form of a single statement or a set of statements, may be expressed in “thing-property-value” triples independently, or may be bound in confined records. These descriptions are the vehicles that carry the information about objects or things. This section will provide some examples of them for the purpose of getting familiar with metadata. Chapter 3 will provide more examples and detailed explanations.

An example would be descriptions for music recordings. Suppose you bought an album from an online vendor and stored it in your iTunes software (or other media program). When examining the information about a track in the album (as shown in figure 1-7-1), you will find the metadata description (on the right side of the image), which contains the property-value pairs for song title, album name, file size, date modified, last played, date of purchase, playing time, artists, and other technical details. Using the default setting or choosing from among the View Options (left side of the image), the properties to be displayed and sorted can be controlled. When listed in a table format, values of those properties (e.g., name, time, artist, album, genre, rating, as seen across the top of the table) become sortable.

FIGURE 1-7-1

A metadata description for a track in iTunes (right), with the View Options (left) and a display showing sortable properties (marked as #3)

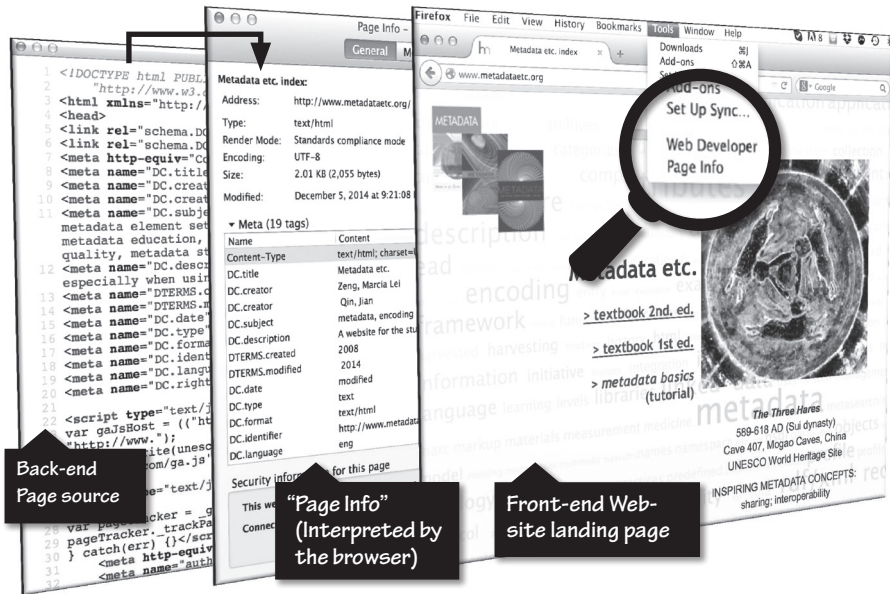


Source: Screenshot from personal computer.

When surfing the ocean of the web, it should be understood that a metadata description encoded with HTML (HyperText Markup Language) is usually embedded in a web page. Exhibit 1-7-1 shows a set of metadata statements written in the tag `<meta>`. This tag is “hidden” in the `<head>` section of an HTML file and is invisible to web page viewers, but it is available for web crawlers (see figure 1-7-2 and exhibit 1-7-1). (To view from Firefox, choose: Tools → Web developer → Page source). This type of approach allows authors to markup web pages with brief and descriptive names (`name = "..."`) for their properties. These metadata statements are critical for a website to be exposed to search engines and data aggregators.

FIGURE 1-7-2

The *Metadata etc.* web site: Front-end landing page, back-end page source, and “Page Info” interpreted into human readable format by the browser



In this type of source code, each line of `<meta>` element contains a “name” attribute for a data field or property and a “content” attribute for its value. The example below (exhibit 1-7-1) uses Dublin Core elements (which have the prefix “DC”); for instance, `DC.title` indicates that this statement is about the title of the web site.

Rather than embedding metadata only into the `<head>` section as shown in exhibit 1-7-1, more and more web pages embed metadata in the main `<body>` section of an HTML file, thanks to *schema.org*, a metadata vocabulary that provides a standard set of schemas for structured data markup on web pages of all kinds (e.g., creative works, job postings, sporting events, medical tests, people and organizations, places, etc.). In other words, the semantic meaning of a particular component in a web page can be encoded with *schema.org* properties using microdata (a set of tags introduced with HTML5) or RDFa (Resource Description Framework in Attributes) syntax. The result is that the original unstructured web page content becomes semantically structured content and exposed to search engines. Figure 1-7-3 uses a simplified example to demonstrate the changes that arise from adding some microdata codes on a web page’s HTML source. Viewing the web page itself, nothing seems to have changed; however, machine-understandable codes would generate extracted structure data for search engines and Web crawlers to use.

## EXHIBIT 1-7-1

Portion of the metadata description for the *Metadata etc.* web site

```

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" >
<head>
  <link rel="schema.DC" href="http://purl.org/dc/elements/1.1/" />
  <meta name="DC.title" content="Metadata etc." />
  <meta name="DC.subject" content="metadata, learning metadata, ..." />
  <meta name="DC.description" content="A website for the study of metadata, ..." />
  <meta name="DC.date" content="2008" />
  <meta name="DC.format" content="text/html" />
  <meta name="DC.identifier" content="http://www.metadataetc.org/" />
  <meta name="DC.language" content="eng" />
</head>
<body>
...
</body>
</html>

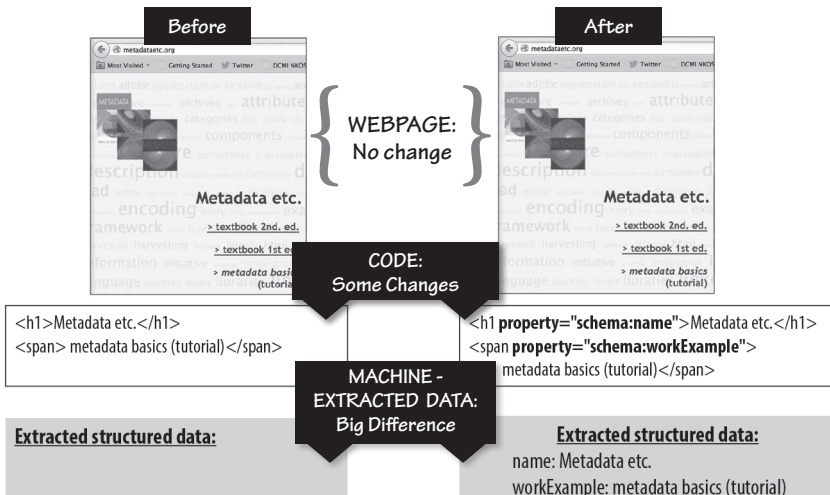
```

Source: <http://metadataetc.org>.

## FIGURE 1-7-3

## A simplified example of embedding semantic markup in a web page

## Using RDFa to encode in the &lt;body&gt; section of a HTML page



Schema.org encourages on-page inline markup so that search engines can understand the information on web pages and provide richer search results. Other schemas for similar purposes have existed for more than a decade. (Chapter 10 will introduce some other standards that are used for embedding semantic markup in photos and news.)

For many materials that are not open to the web, productivity tools such as Microsoft Office and Adobe Bridge (as shown earlier in figure 1-4-1) have the function of capturing basic metadata about objects and embedding it within the file itself. The following example (figure 1-7-4) is a PDF file's metadata description, which displays embedded information about the authors, article title, file name, and file path/location. The authors' names are captured from the "author" identity stored in the software, whereas title is extracted directly from the title of the document. (To view from Adobe Acrobat, choose: File → Properties.) Authors can also add additional metadata values, for example, keywords and subject, to the metadata description to provide more specific information about the documents they create.

FIGURE 1-7-4

### A metadata description embedded in a PDF file

DC2012 science metadata model rev-4.docx Properties

General Summary Statistics Contents Custom

Title: **Functional and Architectural Requirements for Metadata**

Subject: **Metadata**

Author: **Jian Qin, Alex Ball, Jane Greenberg**

Manager:

Company: **Syracuse University**

Category: **Conference paper**

Keywords: **Scientific data, metadata, metadata models**

Comments: **Online access to the full paper: <http://eslib.ischool.syr.edu/jqin/wp-content/uploads/2011/02/DC2012-science-metadata-model-rev-final.pdf>**

Hyperlink base:

Template: **DC-1col.dot**

Save preview picture with this document

The preceding three examples (exhibit 1-7-1, figure 1-7-3, and figure 1-7-4) are for embedded metadata descriptions. The fact that information objects or documents contain metadata descriptions makes them “self-descriptive.” This feature of self-description is of critical importance with digital objects for two reasons: (1) it improves the objects’ discoverability, and (2) it reduces the cost of metadata generation while maintaining reasonable quality.

Although many digital objects are embedded with self-descriptive metadata, not all objects—especially physical objects (e.g., two- and three-dimensional objects not in digital format)—have embedded, self-descriptive metadata. The metadata descriptions for physical objects are usually created by trained personnel through predefined template forms (figure 1-7-5) and stored in a database as the “content infrastructure” for information search and retrieval services. As pictured on the left side of the figure, a template from the digital collection management software CONTENTdm allows a professional user to input data values and create descriptions in a pre-set structure. The records generated will go through the workflow of evaluation, validation, and approval before being indexed by the system and displayed online. The pre-setting, as shown on the right side, follows an application profile and controls how the contents are to be included and indexed

**FIGURE 1-7-5** Screenshots of a metadata description template (left) and the configuration at the back-end (right)

**Input data values and create descriptions using a template**

**Manage metadata fields behind the template, set them as searchable, viewable, mandatory, vocabulary-control, etc.**

Field name	DC map	Data type	Large	Search	Hide	Required	Vocab	add field
1 Title	Title	Text	No	Yes	No	Yes	No	move to   edit   delete
2 Subject	Subject	Text	No	Yes	No	No	Yes	move to   edit   delete
3 Description	Description	Text	Yes	Yes	No	No	No	move to   edit   delete
4 Creator	Creator	Text	No	Yes	No	No	No	move to   edit   delete
5 Publisher	Publisher	Text	No	Yes	No	No	No	move to   edit   delete
6 Date	Date	Date	No	Yes	No	No	No	move to   edit   delete
7 Type	Type	Text	No	Yes	No	No	Yes	move to   edit   delete
8 Format	Format	Text	No	Yes	No	No	Yes	move to   edit   delete

Source: Based on the Kent Class Collection established using a free educational account granted by CONTENTdm.

at the back-end, and what is to be displayed or searchable at the front-end. To get a sense of the online experience, visit the CONTENTdm portal at [www.contentdm.org/](http://www.contentdm.org/) and browse its various collections. (Chapter 3 of this book also gives examples of table-layout online display and individual item display.)

This content infrastructure is also what our digital society relies on for information discovery and use in business, research, learning, and leisure activities. The next two examples address this type of metadata descriptions as seen from the front-end user interface (as opposed to the back-end view of encoding formatted for machine consumption). Both examples on the next page (figures 1-7-6 and 1-7-7) show that these metadata descriptions appear in a set to describe all of the important features of the object being described, which may be much more comprehensive than the embedded metadata showed previously. These metadata descriptions may be drawn from more than one source of data. The description in figure 1-7-6 is for a resource from a digital repository. It displays author, title, keyword, and abstract information for the resource, as well as technical details for the files. Because this resource is archived in an institutional repository, its description does not have the usual detail about who published it and where it was published, because its presence in the institutional repository at Cornell University renders such information redundant. The example in figure 1-7-7 provides a description for a drawing of a cactus with information on the scientific name, original location, type of collection, and the portion of the cactus shown in the drawing. For this artistic work, the artist's name and type of artwork are included in the description as well.

A majority of metadata created by the library, archive, and museum communities are stored in databases or data stores. The databases are structured to store and connect data effectively to eliminate redundancies (e.g., for repeating person or organization information) while ensuring consistency. The complexity of metadata standards used by different communities for different audiences or purposes requires a significant amount of human intelligence to perform metadata creation tasks. Depending on the software and format used for encoding and managing metadata, a description could be expressed differently from one context to another. (Chapter 3 uses a number of different examples to demonstrate how metadata descriptions are found at the back-end, such as in editing templates or XML editors, as well as in web-accessible database displays.)

You may recall that this chapter started with an introduction of metadata that can be found in our daily lives and work. In that sense, the intention of this chapter is to plant a seed in your mind to underscore that metadata is everywhere. In libraries, archives, museums, and beyond, metadata generally play a low-key

FIGURE 1-7-6

## A metadata description example as seen from the front-end of eCommons@Cornell, the institutional repository at Cornell University

The screenshot shows the eCommons@Cornell interface. On the left is a navigation menu with links like Home, About, Help, Browse All, Title, Author, Subject, etc. The main content area displays the following metadata:

- Title:** Defining the Librarian's Role in VIVO
- Authors:** Albert, Pajul J., Cuddy, Colleen
- Keywords:** VIVO
- Issue Date:** 18-May-2012
- Abstract:** Objectives VIVO is an open source semantic web application that enables the discovery of research through interlinked profiles of people and other research-related information. Librarians play invaluable roles in implementing and improving VIVO, assisting with data quality and provenance as well as characterizing researchers' information needs. This poster will define the expertise librarians bring to VIVO implementations and suggest future roles. Methods Building upon the work of librarians at our institution, such as identifying target data sources, negotiating with data stewards, modeling data in a semantic way, resolving gaps and conflicts, and defining policy, the authors will develop a survey tool and survey librarians at institutions with VIVO implementations. The survey tool will query librarians about their current role and contributions as well as anticipated contributions. Additional questions will define the amount of time devoted to VIVO as well as funding sources for their efforts. The poster will graphically display the key results of the survey and suggest future directions for libraries and VIVO, providing data for libraries considering their role in implementations to help make their case for their role on the implementation team and funding. The survey was administered January 18-24, 2012.
- URI:** <http://hdl.handle.net/1813/28704>
- Appears in Collections:** Publications and Presentations by WMC Library Staff

Below the abstract is a table titled "Files in This Item:" with columns for FILE, DESCRIPTION, SIZE, and FORMAT. It lists two files: "LibrariansRolesinVIVOimplementations\_Data-Cleaned.xls" (Raw data, 17 KB, Microsoft Excel) and "2012-AlbertCuddy-MLA.pdf" (PDF of poster, 1.62 MB, Adobe PDF). There are also buttons for "Show full item record" and "Export to: @ BIBLIOTHECA".

Source: Cornell University Library (<http://ecommons.cornell.edu/handle/1813/28704>; accessed 2015).

FIGURE 1-7-7

## A metadata description about a plant drawing, as seen from the front-end of the online Botany Collections of the Smithsonian National Museum of Natural History

The screenshot shows the Smithsonian Botany Collections metadata page. On the left is a text block with the following information:

- From the Catalog of Botanical Illustrations, Department of Botany, Smithsonian Institution**
- [Close Window](#)
- [View image use policy.](#)
- [Request permission](#) to use the image.
- Plate Number:** 1805
- Publication:** The Cactaceae Vol. 3 Pl 24, Fig 2 and 3
- Client:** Britton, N.L. and Rose, J.N. - Size: 11x14
- Cactus harlowii** (Cactaceae) - Type; Collection: , Cuba, Guantanamo Bay; top of fruiting plant, crown.
- Artist:** Eaton, Mary Emily - Date unknown - watercolor

On the right is a botanical illustration of a cactus, showing a top view of a fruiting plant and a side view of the crown. The drawing is signed "M.E. Eaton" in the bottom right corner.

Source: Botany Collections. Permission by Smithsonian Institution (<http://botany.si.edu/botart/showImage.cfm?myimage=images/1805.JPG&mynumber=1805>; captured 2014).



role at the back-end that supports services and products; nonetheless, they are essential for core contents and hubs of information to be integrated and delivered to front-end users. Operating systems change and upgrade constantly, as do the interfaces of web services. In addition, apparatuses used to access information also have evolved to include a wide range of options: from location-bound technologies (office computers and peripherals) to mobile devices, from touch screens to wearables, and from tiny audiobook carriers to giant collection visualization walls. A key strategy for sustainable, effective access to information is to have good quality and linkable metadata ready in order to support the needs for management, retrieval, browsing, discovery, use, and reuse of resources on any device, responsively and dynamically.

### ► SUGGESTED READINGS

- Duval, Erik, Wayne Hodgins, Stuart Sutton, and Stuart L. Weibel. 2002. “Metadata Principles and Practicalities.” *D-Lib Magazine* 8 (4). doi: 10.1045/april2002-weibel.
- Gilliland, Anne J. 2008. “Setting the Stage.” In *Introduction to Metadata: Pathways to Digital Information*, edited by Murtha Baca. Online Edition (Version 3.0). Los Angeles: Getty Research Institute. [www.getty.edu/research/publications/electronic\\_publications/intrometadata/setting.html](http://www.getty.edu/research/publications/electronic_publications/intrometadata/setting.html).
- NISO. 2004. *Understanding Metadata*. Bethesda, MD: NISO Press. [www.niso.org/publications/press/UnderstandingMetadata.pdf](http://www.niso.org/publications/press/UnderstandingMetadata.pdf).
- Weibel, Stuart, Jean Godby, Eric Miller, and Ron Daniel. 1995. *OCLC/NCSA Metadata Workshop Report*. [dublincore.org/workshops/dc1/report.shtml](http://dublincore.org/workshops/dc1/report.shtml).

### ► EXERCISES

These exercises are designed to help you to become more familiar with metadata descriptions. For the most updated links related to the specific tools and examples they address, please consult the book’s web site.

1. First, use structured data to describe yourself (e.g., name, degree program, interests, and any other special elements you would like to include) without following any standard. Then, employ Friend of a Friend (FOAF), one of the standards mentioned in section 1.3, to describe yourself, using a template. Links to templates can be found at the book’s web site.
2. Compare and analyze a variety of metadata instances found in real cases, including web sites, PDF and Word files, digital collections, Google Knowledge Graphs, and Wikipedia Infoboxes. Specific suggestions for further inquiry can be found at the book’s web site.



# Understanding Metadata Vocabularies

**METADATA STANDARDS PROVIDE** guidelines for data structure, data values, data content, and data exchange. They are also the basis for developing software programs and tools that can lead to good descriptive cataloging, consistent documentation, shared records, and increased end-user access (Baca et al. 2006). Chapter 1 provided a list of well-known metadata standards for data structures. However, what are the components of a specification? What roles do these components play in formulating a schema and structuring a metadata description (a document that states an explicit set of requirements to be satisfied by a material, design, product, or service)? This chapter first introduces a selected group of *metadata element sets*. (Note: a full range of standards will be introduced in chapter 10.) They are selected not only because they have been widely adopted but also because they demonstrate some common structures for organizing the elements. An understanding of the approaches used in organizing the elements in a specification (often abbreviated as “spec”) will be essential for learning, selecting, and implementing metadata standards. This chapter also reviews the concept of *application profiles* and introduces metadata vocabularies that are self-described as *ontologies*, *schemas* (or *ontology schemas*), and *RDF vocabularies*.

## 2.1 Metadata Element Sets

### 2.1.1 Components and Structures—An Overview

A metadata element set (in RDF terms, a “property vocabulary”) contains a group of elements used to describe resources of a specific type, or for a particular purpose. A metadata element set is created to fulfill two functions: to define the meanings of the elements and their relationships (semantics) and to provide general instructions on what and how values should be assigned to the elements in an application (content). In a specification, each element is defined by a number of attributes such as the name and label of the element as well as other essential information—definition, identifier, date of release and/or update, and comments. To this end, it is vitally important for all element sets to be defined in a consistent format that can be understood and shared across user communities and platforms. An international standard, ISO/IEC 11179, *Specification and Standardization of Data Elements*, was developed just for this reason (ISO11179, 1995–2012). The following is an example of the element definitions in Dublin Core Metadata Element Set (DCMES) version 1.1, which adopted ten attributes from this standard. The element *date* shown in figure 2-1-1 demonstrates that DCMES defines the semantics and contents of each element (called “term” since 2007) through the following attributes (ANSI/NISO Z39.85-2007; DCMES 1999):

- ▶ *[Term] Name.* A non-ambiguous token for use in machine-processing.
- ▶ *Label.* A descriptive label for human understanding.
- ▶ *Identifier.* The unique identifier assigned to the data element.
- ▶ *Definition.* A statement that clearly represents the concept and essential nature of the data element.
- ▶ *Comment.* A remark concerning the application of the data element.

Note that the metadata term’s *name* (the token for machine processing), “date,” uses lowercase, whereas the human-readable *label*, “Date,” is capitalized. As a best practice, the token is always a unique, machine-understandable string (which may contain one or multiple words), intended to simplify the syntactic specification of elements for encoding schemes. For example, an element for date of birth would be syntactically named as “dateOfBirth.” It follows the convention of capitalizing the first character of each word except the first word, and compounding the name (“cellPhoneNumber”) unless the first word is a proper noun (“AppleSerialNumber”). Adhering to case conventions in element names will avoid conflicts in situations where metadata is used in case-sensitive environments, such as XML.

FIGURE 2-1-1

An entry for DC element *date*

element (name, URI, label)	Term Name: date
definition	URI: <a href="http://purl.org/dc/elements/1.1/date">http://purl.org/dc/elements/1.1/date</a>
best practices + data value restriction	Label: Date
reference	Definition: A point or period of time associated with an event in the lifecycle of the resource.
	Comment: Date may be used to express temporal information at any level of granularity. Recommended best practice is to use an encoding scheme, such as the W3CDTF profile of ISO 8601 [W3CDTF].
	References: [W3CDTF] <a href="http://www.w3.org/TR/NOTE-datetime">http://www.w3.org/TR/NOTE-datetime</a>

Source: Annotation based on an entry in DCMES 1.1 documentation (<http://dublincore.org/documents/dces/>).

When we design or implement a metadata schema, the case conventions in element names should always be enforced.

A *definition* is a statement that represents the conceptual (and essential) nature of the term. The definition should be concise, clear, and whenever possible, no longer than one sentence; it should include only the minimally necessary information. The definition states what the concept *is*, rather than what *it is not* (see example of definition in figure 2-1-1). More examples of definitions can be found in DCMES 1.1. Note that not all element sets, however, formally define the elements. Using comments to further specify an element's meaning and how it should be used in a specific environment is another method used in element set specifications (refer to figure 2-1-2 for an element defined by VRA Core 4.0).

The *comment* attribute is used in DCMES to give explanations or instructions on which, and how, values should be assigned to the elements when creating a metadata description. They typically explain allowable content values: whether values must be taken from a specified controlled vocabulary, supplied by authors or metadata creators, or derived from text. Metadata standards may use *comment*, *note*, or *description* attributes to explain content rules or best practices for how content (e.g., date) should be recorded.

In order to ensure interoperability between different versions of a specification or between different element sets, links to corresponding elements in a related standard or standards (known as “crosswalks”) may also be provided with the element. Taking an example from VRA Core 4.0, each element is crosswalked to the

**FIGURE 2-1-2**  
**An entry for VRA Core 4 element *title***

<p>element (name, attributes)</p>	<p><b>TITLE</b></p> <p>Attributes: <i>type</i></p>
<p>definition + best practices</p>	<p><b>Definition:</b> The title or identifying phrase given to a Work or an Image. For complex works or series the title may refer to a discrete component or unit within the larger entity (a print from a series, a panel from a fresco cycle, a building within a temple complex) or may identify only the larger entity itself. Record multiple titles in repeating instances of the title element. Indicate the preferred title with <i>pref</i> = "true" and alternate titles with <i>pref</i> = "false." For an Image record this category describes the specific view of the depicted Work or Collection, and corresponds to the CCO View Description.</p>
<p>data value restriction</p>	<p><b>Data Values:</b> formulated according to data content rules for titles of works of art.</p> <p><b>Restricted schema values for WORK title <i>type</i>:</b> brandName, cited, creator, descriptive, former, inscribed, owner, popular, repository, translated, other</p> <p><b>Restricted schema values for IMAGE title <i>type</i>:</b> generalView, or partialView.</p>
<p>crosswalks</p>	<p><b>VRA Core 2.0:</b> W2 Title; V7 Visual Document View Description</p> <p><b>VRA Core 3.0:</b> Title</p> <p><b>CDWA:</b> Titles or Names-Text; Related Visual Documentation-Image View; Related Visual Documentation-Image View Type</p> <p><b>Dublin Core:</b> TITLE</p>
<p>obligation</p>	<p><b>CCO:</b> Part TWO: Chapter 1: <i>Object Naming</i></p> <p><b>Not required; Repeatable</b></p>

Source: Annotation based on an entry in VRA Core 4 documentation ([www.loc.gov/standards/vracore/VRA\\_Core4\\_Element\\_Description.pdf](http://www.loc.gov/standards/vracore/VRA_Core4_Element_Description.pdf)).

corresponding elements in its two previous versions (VRA Core 2.0 and 3.0), two related standards (CDWA and DC), and a content standard (CCO) (figure 2-1-2).

A number of other attributes—registration authority, and language—are also required for documenting element sets, according to ISO/IEC 11179. Because these attributes carry the same values throughout the whole metadata element set, they are usually presented only once at the complete element set level rather than listed under each element.

Definitions for individual elements in a set may be revised from time to time. For example, DCMES 1.1 (also reconfirmed as US standard NISO Z39.85 and international standard ISO 15836) has revised the definitions of several elements used in earlier years' specifications, as illustrated by the two versions of definition for the *format* element:

*format*, as defined in the previous ANSI/NISO Z39.85-2001 and ISO 15836-2003, is "The physical or digital manifestation of the resource."  
*format*, as defined in ANSI/NISO Z39.85-2007 and ISO 15836-2009 documents that correspond to DCMES version 1.1, 2006-12-18, is "The file format, physical medium, or dimensions of the resource."

Although the version number for DCMES has remained as version 1.1 since 2001, many changes have been made to its elements. The examples above show that even within the same version, there can be significant changes to individual elements. Any implementer and application profile creator should be mindful of the revisions and should correctly cite the version as well as the revision date of a specification. *Version*, as an attribute that functions as a specific historical description of a term, is applied to each element in Dublin Core specifications.

The semantics of a metadata element are not only about the element's meaning but also about the relationships it has with other elements in the same element set. There are different ways that metadata elements may be organized. Each will result in a distinct representation or structure of an element set scheme. One standard may simply arrange all elements in a linear style, that is, a flat structure, whereas others may use a nested structure to indicate relationships between elements (despite variations in the way they are presented).

We can use a cake recipe as an example to start thinking about what relationships there are between the elements and how they can be structured to best reflect these relationships (see table 2-1-1).

TABLE 2-1-1

**A sample recipe** (some details omitted)

**KATE'S CAKE**

**Total Time:** 1 HOUR

**Prep Time:** 15 MINUTES

**Cook Time:** 45 MINUTES

**Serving Size:** 20

**Ingredients:** 2 cups sugar, 1¾ cups flour, ¾ cup cocoa powder, 1½ teaspoons baking powder, 2 eggs, . . .

**Directions:** 1. Heat oven to 350°F, 2. . . ., 10. Cool 15 to 20 minutes before removing cake from pan.

Elements in this structured description may be arranged either in a flat structure (as shown in the first column of table 2-1-2) or in a nested structure (the right two columns of table 2-1-2). The flat structure places every element at the same level—showing no parent or child relationships. The nested structures are more fluid and leave room for flexibility in organizing the elements. The two columns for the nested structure in table 2-1-2 have a parent element *time* or *timeNeeded* for all three child elements regarding time. They also have a structure for *ingredients* that organizes child elements by type and measure. It is apparent that the nested structures are poised to facilitate a finer representation of resources than the flat structure.

Each of the structures has pros and cons in terms of generating metadata descriptions in a contained environment or in a situation involving the conversion or integration of metadata descriptions between two or more metadata collections that used different standards. All of these structures can be found in the element sets to be introduced in this book. In the following sections, selected metadata element sets are used to demonstrate the characteristics of various structures when examining their semantic aspects.



## 2.1.2 Flat Structure

### 2.1.2.1 Dublin Core Metadata Element Set

A good representative of metadata element sets with a flat structure is the *Dublin Core Metadata Element Set* (DCMES), also known as *Dublin Core* or *DC*. Other names referring DCMES include “simple Dublin Core” and “15 Dublin Core elements.” The name “Dublin” comes from its origin at an invitational workshop held in Dublin, Ohio, in 1995, and “core” implies its broad and generic elements, useful for describing a wide range of resources (DCMES version 1.1, updated 2012). The initial question that DCMES intended to address was, “Can a simple metadata format be defined that sufficiently describes a wide range of electronic objects?” (Weibel 1995, Baker 2012). This basic description mechanism was designed to be both simple *and* powerful, able to be used in all domains, applicable to any type of resource, and extensible in order to work for specific solutions (Baker 2005). The design of the Dublin Core element set, from its origin, sought the basic principles of simplicity, compatibility, and extensibility. After a few years of testing and discussions, the Dublin Core Metadata Initiative (DCMI) officially confirmed its 15 core elements (DCMES Version 1.1, 1998, <http://dublincore.org/documents/1998/09/dces/>). DCMES version 1.1 was approved as an American standard (ANSI/NISO Z39.85-2001 *The*



**TABLE 2-1-2** Side-by-side comparison of representations of the cake recipe structures

Different Structures		
Flat Structure	Nested Structure	
		
Element A ... .. Element N	Element A . Sub-element Aa . Sub-element Ab ... .. Element N	Element A (attribute="[value]") . Sub-element Aa (attribute="[value]") . Sub-element Ab (attribute="[value]") ... .. Element N
Seeing the structures in the Cake example		
<b>title:</b> Kate's Cake	<b>title:</b> Kate's Cake	<b>title</b> (type="short"): Kate's Cake
<b>totalTime:</b> 1 Hour <b>prepTime:</b> 15 minutes <b>cookTime:</b> 45 minutes	<b>time</b> . <b>totalTime:</b> 1 Hour . <b>prepTime:</b> 15 minutes . <b>cookTime:</b> 45 minutes	<b>timeNeeded</b> . <b>time</b> (type="total" unit="hour"): 1 . <b>time</b> (type="prep" unit="min."): 15 . <b>time</b> (type="cook" unit="min."): 45
<b>servingSize:</b> 20 <b>cakeSize:</b> 8"	<b>size</b> . <b>servingSize:</b> 20 . <b>cakeSize:</b> 8"	<b>size</b> . <b>serving:</b> 20 . <b>cakeSize</b> (shape="square" unit="inch"): 8
<b>ingredients:</b> 2 cups sugar, 1¾ cups flour, ...	<b>ingredients</b> . <b>ingredient:</b> . . <b>type:</b> sugar; . . <b>measure:</b> 2 cups . <b>ingredient:</b> . . <b>type:</b> flour . . <b>measure:</b> 1¾ cups ...	<b>Ingredients</b> . <b>ingredient</b> . . <b>ingredientType:</b> sugar . . <b>ingredientMeasure</b> (unit="cup"): 2 . <b>ingredient</b> . . <b>ingredientType:</b> flour . . <b>ingredientMeasure</b> (unit="cup"): 1¾ ...
<b>directions:</b> 1. Heat oven to 350°F; 2. ...	<b>directions:</b> 1. Heat oven to 350°F; 2. ...	<b>directions:</b> 1. Heat oven to 350°F; 2. ...



*Dublin Core Metadata Element Set*) and an international standard (ISO 15836:2003 *Information and documentation — The Dublin Core metadata element set*). In 2007, DCMES version 1.1 was further revised and issued as NISO Z39.85-2007 and ISO 15836:2009 (the most current version at the time of this writing). The history of Dublin Core and other metadata standards will be presented in detail in chapter 10.

The Dublin Core semantics are documented in three types of specifications developed by DCMI: the 15 metadata elements in the element set, extensions and refinements (other elements and element refinements), and encoding schemes.

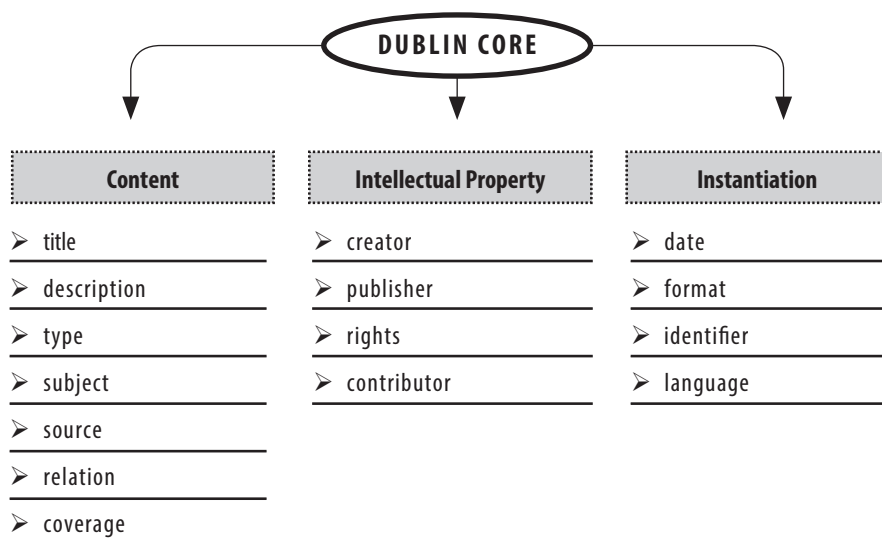
### 1. Metadata Elements

The 15 core elements in DCMES represent an interdisciplinary consensus on basic element sets for resource discovery. They are designed as a flat structure; specifically, there are no hierarchical relationships among the DC elements. All elements are optional, repeatable, and can occur in any order. The occurrence of each element in a metadata record could range from zero to many.

Despite the fact that DCMES has a flat structure and no relationships between its elements are defined, we can still group the elements according to their primary functions. In order to understand them better, we can group the elements into three categories (figure 2-1-3).

FIGURE 2-1-3

#### The 15 Dublin Core elements seen from three categories



Source: Composite based on Dublin Core Metadata Element Set, Version 1.1 (<http://dublincore.org/documents/dces/>).

1. *Content* of any information-bearing object typically has attributes such as title, subject, type, description, and source.
2. Content also has an *intellectual property* relationship with the creator and contributor of the content, the publisher of the object, and the rightful owner (rights).
3. *Instances* of the content have attributes such as date, format, language, and identifier.

Using a technical report as an example, the report could be released as an HTML document, a PDF file, or a Microsoft Word file at different times. These may be created/published on different dates, in varying formats, and identified by different URLs on the web, but their content (e.g., *title, subject, type, description*) will remain the same. When an international organization (e.g., the United Nations) manifests its homepage in multiple languages, the content of the web page will stay the same, although its instantiation attributes (e.g., *language, date, identifier*) might be different. On the other hand, different instances may be released by different publishers with varying access and usage permissions, or contributors could change when the same content is expressed in different formats and languages. Situations such as these make it useful to reuse some metadata statements (e.g., *title, description*) when the instantiation (e.g., *date, format, or identifier*) is different.

Grouping metadata elements into these categories helps us better understand Dublin Core's "One-to-One principle" and apply it in practice. As Hillmann's *Using Dublin Core* (2005) explains, Dublin Core metadata generally describes one manifestation or version of a resource, rather than assuming that manifestations stand in for one another. A digital image of the original Mona Lisa painting, for instance, has much in common with the original painting, but it is not equivalent to the painting. As such, the digital image should be described as a resource unto itself, and most likely the creator of the digital image needs to be included as a *creator* or *contributor* of the digital image, rather than the painter—Leonardo Da Vinci—of the original painting. In this case, two descriptions will be created: one for the digital image and the other for the original painting. The metadata description for the original will have the necessary elements that describe the original *creator, type, format*, etc. The relationship between the reproduction and original work can be established with attributes such as *subject, description*, part of the *title*, and *relation*. This will assist the user in determining whether he/she needs to go to the Louvre to view the original, or if his/her needs can be met by viewing a reproduction (Hillmann 2005).

TABLE 2-1-3

## Dublin Core Version 1.1 (DCMES) Elements with Refinements and Encoding Schemes

Element identifier	Refinement	Encoding Scheme		
		URI		
	bibliographicCitation			
title	alternative			
creator				
subject		DDC LCSH	LCC UDC	NLM MESH
description	tableOfContents abstract			
publisher				
contributor				
date		W3CDTF		
	created valid issued modified	dateCopyrighted dateSubmitted available dateAccepted		
type		DCMIType		
format		IMT		
	extent medium			
source		URI		
language		RFC 4646 ISO639-2 ISO639-3	RFC1766 RFC3066 RFC 5646	
relation	isVersionOf isReplacedBy isRequiredBy isPartOf isReferencedBy isFormatOf conformsTo	hasVersion replaces requires hasPart references hasFormat		
coverage	spatial  temporal	Box Point Period	ISO3166 TGN W3CDTF	
rights	accessRights license			

Source: Compiled according to *DCMI Metadata Terms* 2012-06-14 version (<http://dublincore.org/documents/dcmi-terms/>).

## 2. Extensions and Refinements

Using qualifiers to disambiguate or narrow down a term is a technique frequently used in controlled vocabularies. A qualifier is a word or phrase added to the end of a term to avoid possible misunderstandings, as in “Pool (game),” or to narrow down a broad term as in “Coverage.Temporal.” In order to improve the semantic precision of the DC elements, a set of *qualifiers* was first introduced in an official document, *Dublin Core Qualifiers* ([dublincore.org/documents/2000/07/11/dcmes-qualifiers/](http://dublincore.org/documents/2000/07/11/dcmes-qualifiers/)), and issued by the Dublin Core Usage Committee in 2000. Examples include *tableOfContents*, which narrows down the *description* element, and *issued* that narrows down the *date* element. These qualifiers, together with the 15 core elements, resulted in a version that has been referred to as the “Qualified Dublin Core.” In turn, the version with the original 15 elements is referred to as the “Simple Dublin Core.” At present, the term “qualifiers” has been replaced by “refinements.” Table 2-1-3 presents all of the refinements together with the value encoding schemes (which is integral to the third part of DC semantics, discussed below).

Originally a refinement (e.g., *alternative*) had to be used together with the element (e.g., *title*) that it refines. For example, to encode the alternative title “ABC” of a web site we use *title.alternative*: “ABC.” Since 2003, all refinements have been declared to be DC “terms,” and are maintained in the *DCMI Metadata Terms* (to be introduced in section 2.4.2 of this chapter), an RDF vocabulary. Using the RDF terminology, the elements are “properties.” The refinements are also considered properties (Baker 2012). Properties can be used to refine other properties. In the example mentioned above, *alternative* is a property that refines *title*. A property may also stand alone in metadata descriptions, e.g., *alternative*: “ABC” (table 2-1-4, note the difference for the underlined property name). Both usages, however, may coexist in practice.

TABLE 2-1-4

### Qualifiers become properties

Previous convention ( <i>alternative as a refinement</i> )	Current convention ( <i>alternative as a property</i> )
title: "XYZ"	title: "XYZ"
<u>title.alternative</u> : "ABC"	<u>alternative</u> : "ABC"

**TABLE 2-1-5**  
**Two types of encoding schemes**

Examples of Encoding Schemes	
Vocabulary encoding schemes	Syntax encoding schemes
<ul style="list-style-type: none"> <li>• ISO-639-2 <i>Codes for the Representation of Names of Language</i></li> <li>• <i>Library of Congress Subject Headings (LCSH)</i></li> <li>• <i>Dewey Decimal Classification (DDC)</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>W3C Date and Time Formats (W3CDTF)</i></li> <li>• <i>DCMI Point Encoding Scheme</i></li> </ul>

### 3. Encoding Schemes

Encoding schemes for controlling the data values form another essential part of the DC semantics. The DCMI community identified and registered a number of encoding schemes. Two types of encoding schemes are involved: vocabulary encoding schemes and syntax encoding schemes (table 2-1-5).

*Vocabulary encoding schemes* are employed to control the values to be entered in a metadata statement; for example, the correct way to enter a value about the written *language* used in this document could be: “English,” “en,” “eng,” or “en-US.” The governing controlled vocabulary in this case is the international standard, ISO 639-2, *Codes for the representation of names of languages — Part II: Alpha-3 code*, in which “English” is assigned the code “eng.” The vocabulary encoding schemes provide the authoritative vocabularies to be used. Examples of vocabulary encoding schemes include:

- a. widely used subject headings such as LCSH and *Medical Subject Headings (MeSH)*
- b. classification schemes such as *Dewey Decimal Classification (DDC)*, *Universal Decimal Classification (UDC)*, and *Library of Congress Classification (LCC)*
- c. lists such as the *DCMI Type Vocabulary* and *[Internet] Media Types (IMT)*.

The other type of encoding scheme is the *syntax encoding scheme*. The identified syntax encoding schemes are related to language, URI, geospatial coverage, and time. They set rules about how a string should be formatted in a standardized way. For example, “2007-12-10” follows the World Wide Web Consortium (W3C)’s *Date and Time Formats (W3CDTF)*, which ensures that the meaning of this string is interpreted as December 10, 2007, rather than July 12, 2010, or any other possible date (figure 2-1-4). As the following illustration demonstrates, the recommendation of using an encoding scheme for *dc:date* helps to eliminate many possible mistakes and confusion in communication (figure 2-1-4).

FIGURE 2-1-4

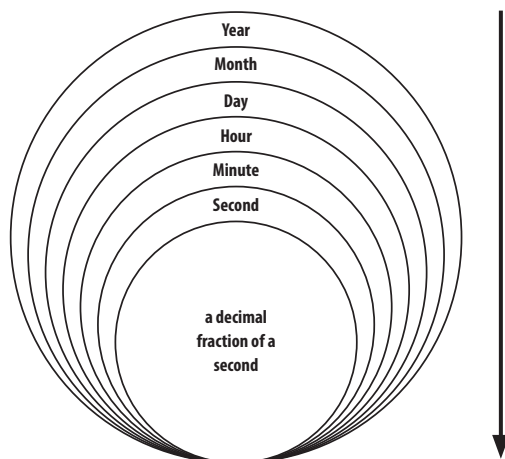
### Illustration on the importance of best practice recommendations provided by a standard, using the example of *dc:date* element

“Recommended best practice is to use an encoding scheme, such as the W3CDTF profile of ISO 8601 [W3CDTF].” → <http://www.w3.org/TR/NOTE-datetime>

EXPIRATION  
DATE: ----/----/----

07/12/10  
yy-mm-dd (2007 Dec. 10)?  
mm-dd-yy (July 12, 2010)?  
mm-yy-dd (July 2012 10th)?  
dd-mm-yy (7 Dec. 2010)?

Complete date:  
✓ YYYY-MM-DD (e.g., 2007-12-10)



As the most widely used metadata element set, DCMES has been published in multiple specifications by using DC-Text, HTML/XHTML, XML, and RDF graphs. Today’s Dublin Core is no longer the same version as the one used in 2000. The 15 elements described in the preceding section are considered “*classic* Dublin Core.” Section 2.4.2 of this chapter will discuss how Dublin Core has been further developed into an RDF vocabulary.

#### 2.1.2.2 VRA Core 3.0

The development of VRA Core Categories by the Visual Resources Association (VRA) Data Standards Committee was in response to needs specific to describing visual resource collections (see also the details of VRA Core standards in chapter 10). VRA Core followed the “One-to-One principle” developed by the Dublin Core community, that is, only one object or resource may be described within a single metadata set (VRA Data Standards Committee, [2000]). This enables a database to contain and associate both *work* records that describe an actual art object and *image* records that describe representations of views of that object (slides, digital images, or others) held by



**TABLE 2-1-6**  
**VRA Core 3.0 elements and qualifiers**

VRA Core 3.0 Elements and Qualifiers		
Record Type	Date	<b>Style/Period</b>
Type	• Date.Creation	• Style/Period.Style
Title	• Date.Design	• Style/Period.Period
• Title.Variant	• Date.Beginning	• Style/Period.Group
• Title.Translation	• Date.Completion	• Style/Period.School
• Title.Series	• Date.Alteration	• Style/Period.Dynasty
• Title.Larger Entity	• Date.Restoration	• Style/Period.Movement
<b>Measurements</b>	<b>Location</b>	<b>Culture</b>
• Measurements.Dimensions	• Location.Current Site	Subject
• Measurements.Format	• Location.Former Site	Relation
• Measurements.Resolution	• Location.Creation Site	• Relation.Identity
<b>Material</b>	• Location.Discovery Site	• Relation.Type
• Material.Medium	• Location.Current Repository	Description
• Material.Support	• Location.Former Repository	Source
<b>Technique</b>	ID Number	Rights
Creator	• ID Number.Current Repository	
• Creator.Role	• ID Number.Former Repository	
• Creator.Attribution	• ID Number.Current Accession	
• Creator.Personal name	• ID Number.Former Accession	
• Creator.Corporate name		

Source: Composite based on VRA Core 3.0 ([www.loc.gov/standards/vracore/VRACore3\\_Element\\_Description.pdf](http://www.loc.gov/standards/vracore/VRACore3_Element_Description.pdf)). Emphases reflect elements for describing visual resources when compared with Dublin Core Element Set

an institution. VRA Core 3.0 consists of a single element set combining two separate element sets for *works* and *images* from version 2.0, which was released in 1997. Similar to Dublin Core's structure, VRA Core 3.0 is "flat." All seventeen elements in VRA Core 3.0 are optional, repeatable, and have no predetermined order of precedence. In VRA Core 3.0, an element may be modified by a *qualifier*. Half of the elements reflect specialized needs for describing visual resources (as emphasized using boldface letters in table 2-1-6).

VRA Core 3.0 has been used in many digital collections. Its flat structure also makes it a good template to use in image collection management software such as CONTENTdm. However, the current version (VRA Core 4.0) included drastic structural changes which will be discussed in the following section.

In summary, the flat structure has the following features:

- *All elements are equal.* There are no superordinate or subordinate relationships between elements in the schema.

- ▶ *An element can be further refined.* An element refinement (e.g., *available*) shares the meaning of a particular element (e.g., *date*) but with narrower semantics. In implementation they are usually used as “qualifiers” (e.g., *date.available*). In DC standards, a refinement only refines one “parent” element (DCMI Grammatical Principles 2003).

## 2.1.3 Nested Structure

Major standards for data structures developed by the LAM communities have used nested structures to organize elements. Although nested structures resemble a hierarchical layout, the relationships between an element and its sub-element(s) are not always semantically hierarchical. The word “nested” is used to characterize the arrangement of elements.

### 2.1.3.1 VRA Core 4.0

In the 2.1.2.2, VRA Core 3.0 was introduced as a flatly structured element set. The current version, VRA Core 4.0 ([www.loc.gov/standards/vracore/](http://www.loc.gov/standards/vracore/)) was released in 2007. The *qualifiers* in 3.0 were converted to *sub-elements* and *attributes* in version 4.0. The majority of the *elements* in version 4.0 remained essentially the same with those in version 3.0, but the structure changed drastically (figure 2-1-5).



The entire VRA Core 4.0 structure has departed from Dublin Core and moved towards consistency with other standards published by the LAM communities prior to 2007. These include CDWA Lite, MODS, and the 2002 version of EAD (see chapter 10 for details). In principle, VRA Core 4.0 continues to follow the one-to-one principle. How the description sets are linked to form a single record is a local database implementation issue. VRA Core 4.0 also supports one-to-many relationships through the *relation* element, for example, part and whole relationships for complex works or relationships between a single work and its multiple images.

In version 4.0, a record can be created for any one of the three types of visual resources:

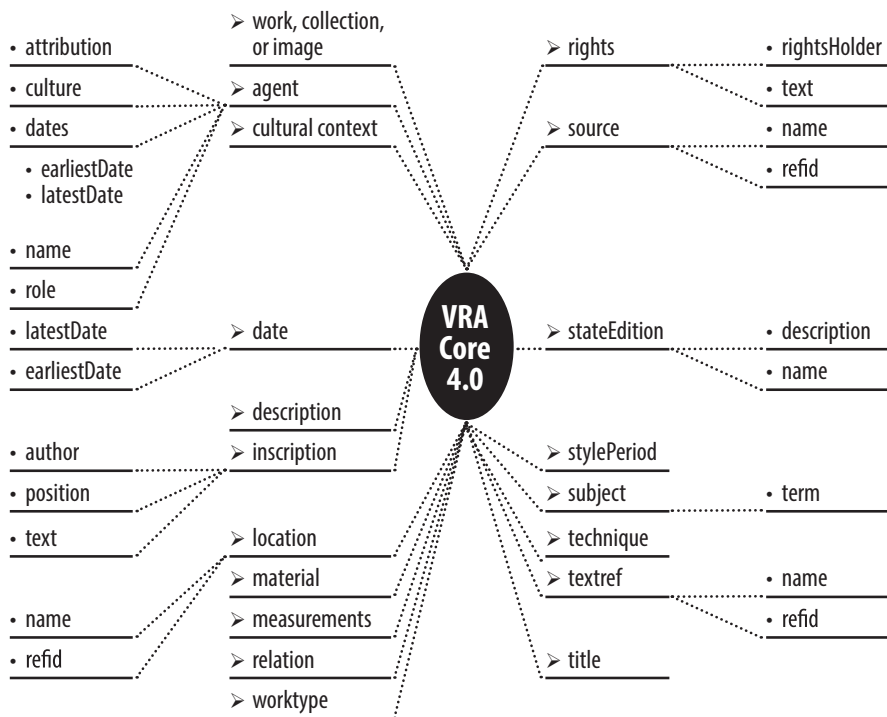
- ▶ *Work.* A unique entity such as an object or event.
- ▶ *Image.* A visual representation of a work in either whole or part.
- ▶ *Collection.* An aggregate of work or image records.



The record type *collection* was added in version 4.0. Another significant change is that the *creator* element was renamed to *agent*, which is a composite element containing several sub-elements: *name*, *role*, *culture*, *dates*, and *attribution* (refer to figure 2-1-5). Meanwhile, *culture*, which existed as a single, freestanding element in 3.0, is now placed in two different elements: as a sub-element under *agent* and as a freestanding element *cultural context* in which the work was created. One more noticeable change in version 4.0 is the restructuring of the *ID number* element. An ID associated with a repository is now a sub-element (i.e., *refid*) of *location*.

The VRA Data Standards Committee developed two versions of the XML Schema for the VRA Core 4.0 metadata element set: one is the unrestricted version that specifies the basic structure of the schema, and the other is the restricted version that extends the unrestricted schema by adding controlled *type* lists and date formats. Examples used in this chapter are based on the restricted version.

**FIGURE 2-1-5**  
**VRA Core 4.0 elements and sub-elements**



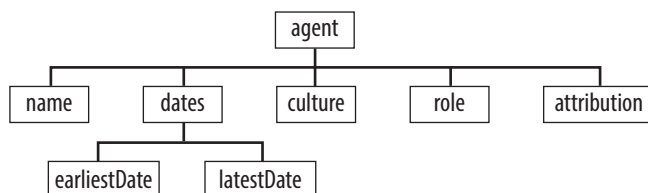
Source: Composite based on VRA Core 4.0 Element Outline 2007 ([loc.gov/standards/vracore/VRA\\_Core4\\_Outline.pdf](http://loc.gov/standards/vracore/VRA_Core4_Outline.pdf)).

VRA Core 4.0 is an example of nested structure, which contains the following features as demonstrated through the *agent* cluster:

1. Elements may have sub-elements; a sub-element may also contain other sub-elements (which may be referred to as sub-sub-elements). In the following example from VRA 4.0, the sub-element *dates* has two sub-sub-elements: *earliestDate* and *latestDate* of *agent* element (figure 2-1-6).

**FIGURE 2-1-6**  
**VRA Core 4.0 *agent* element and sub-elements**

agent  
 . name  
 . dates  
 .. earliestDate  
 .. latestDate  
 . culture  
 . role  
 . attribution



Both elements and sub-elements can have attributes. In the following example, the attribute *type* can be found in the *name*, *dates*, and *title* elements. The values of the *type* attribute to be used with each element are predefined by VRA Core 4.0 in the restricted version of the schema (table 2-1-7, lower box).

**TABLE 2-1-7**  
**Example of VRA Core *agent* element in a description**

In the Element Set	Applied to a Description (in XML format)
<b>agent</b>	<code>&lt;agent&gt;</code>
. name	<code>&lt;name type="personal"&gt;Wright, Frank Lloyd &lt;/name&gt;</code>
attribute: type	<code>&lt;dates type="life"&gt;</code>
. dates	<code>&lt;earliestDate&gt;1867&lt;/earliestDate&gt;</code>
attribute: type	<code>&lt;latestDate&gt;1959&lt;/latestDate&gt;</code>
.. earliestDate	<code>&lt;/dates&gt;</code>
.. latestDate	<code>&lt;culture&gt;American&lt;/culture&gt;</code>
. culture	<code>&lt;role&gt;architect&lt;/role&gt;</code>
. role	<code>&lt;/agent&gt;</code>
<b>title</b>	<code>&lt;title type="brandName"&gt;Fallingwater&lt;/title&gt;</code>
attribute: type	

Note: Data values defined for the **agent name** *type* attribute are: "personal", "corporate", "family", "other"; Data values for the **agent dates** *type* attribute are: "life", "activity", "other". For a complete list of VRA Core 4.0 Restricted Schema Type Values (2007) please see [http://www.loc.gov/standards/vracore/VRA\\_Core4\\_Restricted\\_schema\\_type\\_values.pdf](http://www.loc.gov/standards/vracore/VRA_Core4_Restricted_schema_type_values.pdf)

2. There are also global attributes (e.g., *source*, *href*, and *xml:lang*) that are optional and may be added to any element or sub-element as needed.
3. The same sub-elements may occur under different elements. For example, *earliestDate* and *latestDate* in VRA Core 4.0 are used not only as the sub-sub-elements of the sub-element *dates* of element *agent*, but also as sub-elements of *date* of a work.

As the above example indicates (right column of table 2-1-7), when applying these elements to a metadata description, the values (e.g., a person's name or birth date) should always go to the lowest applicable level of the elements. There is no direct value put under the *agent* element, because it has sub-elements.

One may find the study of VRA Core 4.0 helpful in understanding many other metadata standards in the LAM fields, such as MODS, EAD, and PBCore. These standards are introduced in chapter 10.

---

## 2.2 Application Profiles

### 2.2.1 The Concept of Application Profile

In spite that general-purpose metadata standards such as Dublin Core capture resource attributes that are common across resource types and domains, they are too limited to meet specialized user requirements and local needs within a particular community or within a particular project. The data structure, data values, and data contents requirements provided in a particular metadata element set often need to be modified to fit specialized and/or local requirements.

A typical approach to accommodating specialized requirements is to build *application profiles*. In section 2.1 we introduced DCMES and VRA Core, which can be considered “namespace schemas” (i.e., sets of data elements as defined by their maintainers). “Application profile (AP) schemas” present a different category where the elements may be from one or more element sets, thus allowing a given application to meet its functional requirements by using metadata elements from several element sets, including locally defined sets (DCMI Glossary 2005).

The meaning and practice of application profiling are explained in the following definitions:

- ▶ A profile outlines the extent to which an existing schema would be applied and provides guidelines for its application in the environment in question. The concept is based on the idea that metadata standards are necessarily localized and optimized for specific contents (Johnston 2003).

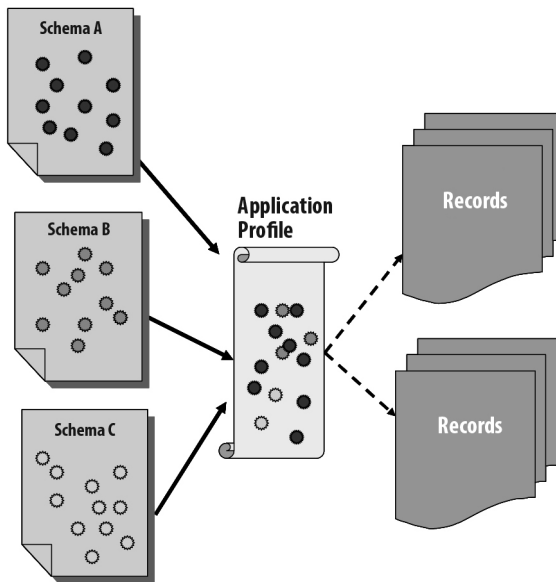
- ▶ An application profile is an assemblage of metadata elements selected from one or more metadata schemas and combined in a compound schema (Duval et al. 2002).

## 2.2.2 Examples of APs Consisting of Elements Drawn from Other Schemas

APs usually consist of metadata elements drawn from one or more metadata namespace schemas, combined into a compound schema by implementers, and optimized for a particular local application (Heery and Patel 2000; Duval et al. 2002). Figure 2-2-1 illustrates an AP consisting of elements drawn from one or more schemas. The AP can then be implemented by one or more different user communities in their metadata creation processes. The use of APs ensures a similar basic structure with common elements, while allowing for varying degrees of depth and detail for different user communities.

FIGURE 2-2-1

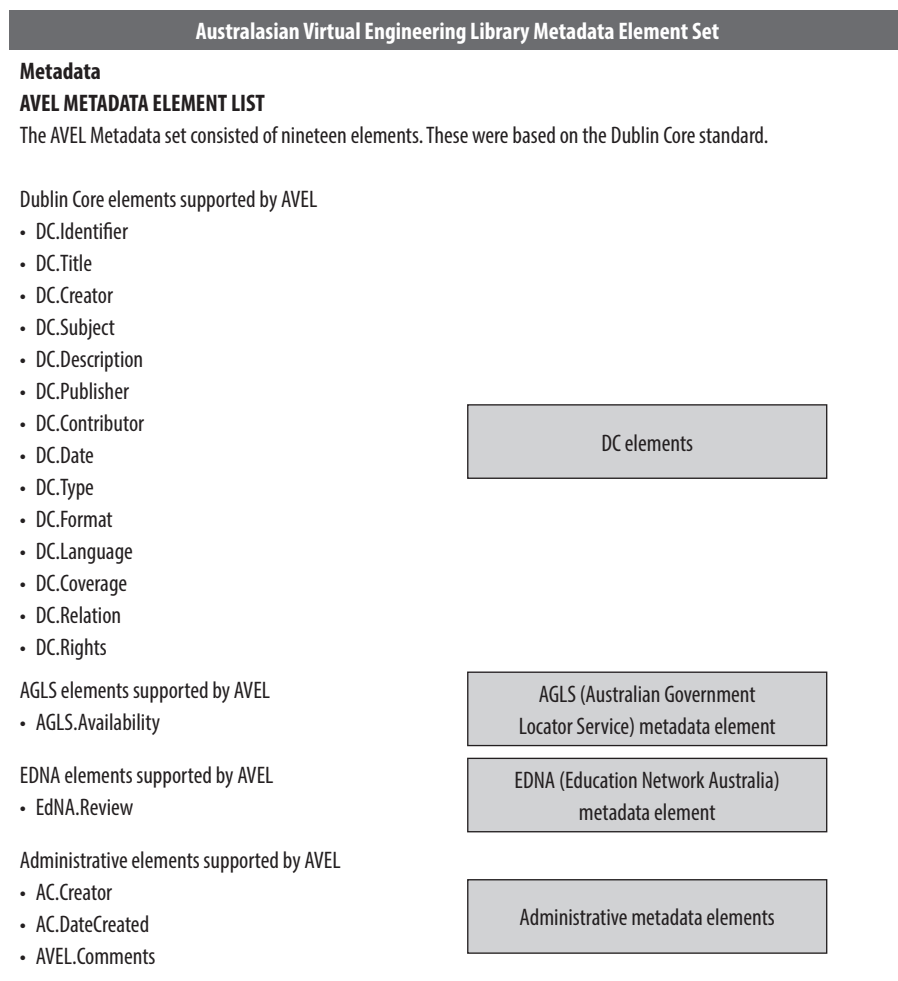
**Illustration of an application profile consisting of metadata elements and refinements drawn from one or more schemas**



For example, the Australasian Virtual Engineering Library developed the *AVEL Metadata Set* ([avel.library.uq.edu.au/technical.html](http://avel.library.uq.edu.au/technical.html)) based on Dublin Core. It consists of nineteen elements. In addition to supporting fourteen DC elements (excluding the *dc.source* element), it also supports one AGLS (Australian Government Locator Service) metadata element: *AGLS.Availability*; one EDNA (Education Network Australia) element: *EdNA.Review*; and three Administrative elements: *AC.Creator*, *AC.DateCreated*, and *AVEL.Comments* (figure 2-2-2).

FIGURE 2-2-2

### AVEL metadata element list



Source: Composite based on AVEL (<http://avel.library.uq.edu.au/technical.html>).

FIGURE 2-2-3

### Portion of the metadata elements within the NSDL\_DC metadata framework (NSDL 2007) updated 2013

Date	Recommended	A point or period of time associated with an event within the lifecycle of the resource. Employ W3CDF encoding scheme that looks like YYYY-MM-DD.	<dc:date>...</dc:date>
Created	Recommended	A refinement of the Date element	<dc:created>...</dc:created>
Available	Optional	A refinement of the Date element	<dc:available>...</dc:available>
dateAccepted	Optional	A refinement of the Date element	<dc:dateAccepted>...</dc:dateAccepted>
dateCopyrighted	Optional	A refinement of the Date element	<dc:dateCopyrighted>...</dc:dateCopyrighted>
dateSubmitted	Optional	A refinement of the Date element	<dc:dateSubmitted>...</dc:dateSubmitted>
Issued	Optional	A refinement of the Date element	<dc:issued>...</dc:issued>
Modified	Optional	A refinement of the Date element	<dc:modified>...</dc:modified>
Valid	Optional	A refinement of the Date element	<dc:valid>...</dc:valid>
Interactivity Type	Recommended if applicable	The type of interactions supported by a resource (active, expositive, mixed, undefined)	<ieee:interactivityType>...</ieee:interactivityType>
Interactivity Level	Recommended if applicable	The level of interaction between a resource and end user, that is the degree to which a learner can influence the behavior of the resource (very high, high, medium, low, very low)	<ieee:interactivityLevel>...</ieee:interactivityLevel>
Typical Learning Time	Optional	The typical amount of time for a particular education level to interact with the resource.	<ieee:typicalLearningTime>...</ieee:typicalLearningTime>
Format	Optional	Physical medium and/or file/MIME format	<dc:format>...</dc:format>
Extent	Optional	The size or duration of the resource.	<dc:extent>...</dc:extent>

Source: NSDL\_DC Metadata Guidelines ([https://wiki.ucar.edu/display/nsdl/docs/nsdl\\_dc](https://wiki.ucar.edu/display/nsdl/docs/nsdl_dc)).

Another example is the AP for the National Science Digital Library (NSDL), a metadata repository for educational resources in the Science, Technology, Engineering, and Mathematics (STEM) fields available on the web. NSDL recommends that the metadata descriptions for collection projects use metadata elements from DC along with three additional elements from IEEE Learning Object Metadata (LOM). Figure 2-2-3 shows a section from the *NSDL\_DC Metadata Guidelines* for: (1) Elements, (2) Recommended Usage, (3) Simple definitions / Notes, and (4) Sample XML tags. The elements from DCMES have a “dc” prefix, the terms from *DCMI Metadata Terms* (to be discussed in section 2.4.2) have a “dct” prefix, and elements from IEEE Learning Object Metadata (LOM) have an “ieec” prefix (NSDL 2007).

An AP may also provide additional documentation on how the terms used are constrained, encoded, or interpreted for specific purposes (Baker 2003). In this example, Column 2 of figure 2-2-3 indicates whether an element is optional, recommended, or recommended if applicable.

It is widely accepted as best practice to develop APs for a community that presents unique needs for extending or modifying an existing metadata schema, or for a specific implementation that requires specific definition of, say, repeatable and/or mandatory fields. Even in cases where no extension or modification of elements is necessary, the implementer still needs to develop an AP for the specific implementation requirements of the project (or products) in regards to cardinality and controlled values, such as in the NDSL example. (Chapter 4 will present steps for developing application profiles.)

### 2.2.3 Sources of Reusable Elements

As the number of metadata standards and application profiles grows, it will be beneficial to have a “shopping center” to check for metadata element sets and APs that have already been developed and to determine which elements are available for reuse in constructing an application profile. Such “shopping centers” do exist, and are called metadata registries. The *CORES Registry* ([cores.dsd.sztaki.hu/](http://cores.dsd.sztaki.hu/)) is one of the earliest registries of metadata element sets, application profiles, and encoding schemes, and includes a large number of activity reports that describe and comment on various metadata-related activities and initiatives. However, the number of metadata vocabularies has been very limited due to voluntary contribution.

*Linked Open Vocabularies* (LOV) ([lov.okfn.org/dataset/lov/](http://lov.okfn.org/dataset/lov/)) is a newer and larger registry that provides information on metadata vocabularies that are expressed with RDF Schema (RDFS) or Web Ontology Language (OWL), both of which are W3C defined languages. LOV has extended quickly since its initiation in March

2011. Note that the term “vocabularies” is used in this context to mean *property vocabularies*. It does not mean *controlled vocabularies* (which are referred to as “value vocabularies” in the RDF community). LOV offers search capabilities for existing elements at either the vocabulary level or element level. In other words, metadata terms (including classes, properties, value spaces, and other relevant artifacts) are searchable either by individual terms or by vocabulary as a whole. For example, a search for the property *audience* yields seventy-eight results (with varying granular levels) in thirty-seven vocabularies. In addition to searching and exploring the vocabulary content, a user can also find metrics about the use of vocabularies on the Semantic Web. As of the beginning of 2015, DCMES (with a prefix of “dc” or “dce”) had been referenced by 308 other metadata vocabularies ([lov.okfn.org/dataset/lov/vocabs/dce](http://lov.okfn.org/dataset/lov/vocabs/dce)). LOV visualizes relationships between vocabularies, such as how many vocabularies have referenced vocabulary A—as well as how and to which vocabularies vocabulary A made reference. These references can be classified as *relies on*, *extends*, *specializes*, or *generalizes*, among other types of relationships.

Currently, the LOV registry only includes metadata vocabularies in RDF serialization formats or OWL. Those expressed in XML but not in RDF format (e.g., VRA Core 4.0) are not included in LOV as of this writing. Thus, for completeness, other registries should also be consulted. The library community’s standards are contained in the *Open Metadata Registry* ([metadataregistry.org/](http://metadataregistry.org/)). This registry’s notable contents are MARC 21, ISBD (International Standard Bibliographic Description), and RDA. It registers many values defined in these metadata standards, for example, each value associated with the MARC 21 007 field (Physical Description Fixed Field-General Information) is registered there with a unique URI.

---

## 2.3 Ontologies as Metadata Vocabularies

### 2.3.1 Background

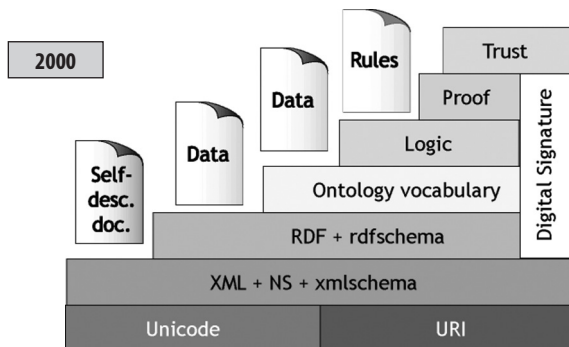
The LOV registry (see section 2.2.3) contained 475 metadata vocabularies as of the beginning of 2015, about half of which have the term “ontology” in the title (e.g., “Music Ontology,” “Copyright Ontology,” “EPrints Ontology”). Metadata vocabularies listed in chapter 1 and reviewed in the beginning of chapter 2 (DCMES, VRA Core) were not initially designed as ontologies, so XML schemas were developed for the implementation of these vocabularies. A transition from XML-schema-style vocabularies to RDF vocabularies gradually took place over the last decade. Meanwhile, hundreds of ontologies have been developed in the twenty-first century, some of which function as metadata vocabularies. This section



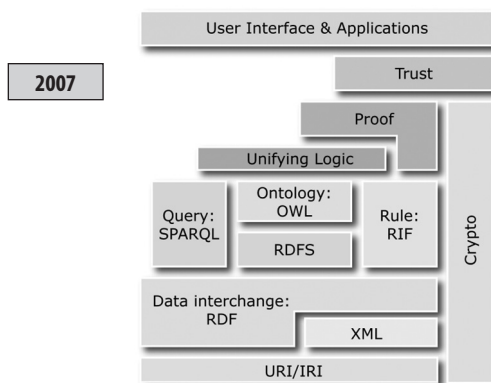
will focus on ontologies as metadata vocabularies, whereas the next section will introduce RDF vocabularies of metadata terms.

According to the W3C “Semantic Web Layer Cake,” a visual model representing the architecture and standards for common data formats on the web, RDF + RDF Schema and Web Ontology Language (OWL) occupy the layers above XML (figure 2-3-1). “The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries. It is a collaborative effort led by the W3C with participation from a large number of researchers and industrial partners. It is based on the Resource Description Framework (RDF)” (W3C, [www.w3.org/2001/sw/](http://www.w3.org/2001/sw/)). The fast growth of ontologies is largely related to the release of OWL in 2004 and OWL 2 in 2009.

**FIGURE 2-3-1**  
**The Semantic Web Layer Cake**



Source: Tim Berners-Lee. 2000. Semantic Web on XML. XML 2000, December 3–8, Washington D.C., <http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html>.



Source: W3C. W3C Semantic Web Activity. <http://www.w3.org/2001/sw/>. Latest layer cake diagram, <http://www.w3.org/2007/03/layerCake.png>.

Ontologies have existed in the fields of computer science and information science for several decades, but there has been a steady increasing interest in using ontologies to model and standardize the ways that different pieces of data relate to one another (Lowe 2007). By definition, an ontology is a formal model that allows knowledge to be represented for a specific domain. An ontology describes (a) the types of things that exist (classes), (b) the relationships between them (properties), and (c) the logical ways those classes and properties can be used together (axioms) (Gruber 1993; ISO 25964-2:2013). There are two general categories of ontologies. “Reference ontologies” are rich, axiomatic theories whose focus is to clarify the intended meanings of terms used in specific domains. “Application ontologies,” also called “lightweight” ontologies, provide a minimal terminological structure to fit the needs of a specific community. (Borge, Guarino, and Masolo 1996; Menzel 2003). The ontologies that are designed and functioning as metadata vocabularies fall into the application ontology category. For this reason, some vocabularies that have employed the ontology structure are self-described as “schemas,” not “ontologies.” Freebase (<https://www.freebase.com/>, from 2007 to 2015-03 and proceeded to [www.wikidata.org](http://www.wikidata.org)) is a community-curated open collection of structured data for millions of well-known people, places, and things. This key resource for Google’s knowledge graphs considers the property vocabularies that are used for description to be “schemas” rather than “ontologies” (as does Schema.org). No matter what they are named or what encoding languages they use, ontologies as metadata vocabularies represent a modular metadata structure. An important feature of these vocabularies is their evolving and extensible nature. Unlike the metadata element sets that have been published as standards and have remained stable for years, these application ontologies or schemas are always evolving. The terms within them can have many different statuses, including “testing,” “unstable,” “archaic,” and “stable,” but the namespace URI of an ontology remains the same even as the vocabulary matures.

It is beyond the focus of this book to get into the details of ontology design and issues. This section will only use one of the most-widely used application ontologies, Friend of a Friend (FOAF), and Schema.org to explain the structure of an ontology that is also used as a metadata vocabulary. In this book, we will follow the convention of ontologies in which names of ontological classes (also referred as “types”) start with a capitalized letter and names of properties start with a lowercase letter.

## 2.3.2 Modular Structure

A module is defined as “a self-contained component (unit or item) that is used in combination with other components” (WordNet, [wordnet.princeton.edu/](http://wordnet.princeton.edu/)). Decomposing a large software program into modules is a common approach in software engineering in order to lower the cost and allow modules to be designed and revised independently. Module decomposition aims to accomplish the specific goal that each module’s structure should be simple enough that it can be understood fully. It should be possible to change the implementation of one module without the knowledge of the implementation of other modules and without affecting the behavior of other modules (Parnas, Clements, and Weiss. 1984). These principles are also applicable to metadata vocabularies where a very large number of elements are involved, but in a modularized structure. Rather than creating an element set for describing an object as a whole, modular metadata structures focus on self-contained components of an object (e.g., information about a person is managed in a different module from the one about an object). These components can exist independently and also be assembled together to form a set of descriptions for a resource being described. Presently many ontologies are expressed using OWL. OWL ontologies are not only capable of describing simple relationships, they can also be designed to describe all kinds of complex relationships between and among individuals of an ontological class.

## 2.3.3 Friend of a Friend (FOAF)

Friend of a Friend (FOAF) is designed as a machine-understandable ontology that describes persons, their activities, and their relationships to other people and objects. The central idea of FOAF is linking networks of information with networks of people. FOAF has been evolving gradually since its creation in the mid-2000s, and has been widely accepted by members of the Internet community, especially in relation to the development of the Semantic Web. FOAF Vocabulary Specification version 0.99 was released in January 2014 ([xmlns.com/foaf/spec/](http://xmlns.com/foaf/spec/)). Main FOAF terms can be grouped in three broad categories:

- ▶ *Core*. These classes and properties describe characteristics of people and social groups that are independent of time and technology.
- ▶ *Social Web*. In addition to the FOAF core terms, there are a number of terms for use when describing Internet accounts, address books, and other web-based activities.
- ▶ *Linked Data utilities*. A set of terms that are useful to the web community.

The Core terms includes these classes, or “types”: Agent (its sub-classes are Group, Organization, Person), Document (its sub-classes are Image, PersonalProfileDocument) and Project. There are more than ten properties in all. An agent instance could be a person, group, software, or physical artifact. Its sub-types consider that:

- ▶ A *Person* instance could be alive, dead, real, or imaginary.
- ▶ *Organization* corresponds to social institutions such as companies, societies, and so on; the concept is intentionally quite broad.
- ▶ *Group* represents a collection of individual agents, informal and ad-hoc groups, long-lived communities, organizational groups within a workplace, and others.

Although it looks like a taxonomy, with three sub-classes being listed under *Agent*, an ontology allows the use of constraints to clarify the relationships between classes much more powerfully than a taxonomy does. For example, *Person* is disjointed from *Organization*, which means the same agent instance cannot be listed as an instance of both classes. *Group*, however, is not disjointed from any class. Rather, the *Group* class represents a collection of individual agents. It may itself play the role of an *Agent*, meaning that it can perform actions (`xmlns.com/foaf/spec/`). Ontological classes all possess a specific set of properties; this is also a fundamental difference from taxonomies in terms of the basic components.

Now let’s use FOAF to describe a *Person* instance. Using FOAF, anyone can create a profile for his/her personal information, work experience and educational background, and (more importantly) the friends (or people) one knows. The latter is used most creatively to link people together in a networked society.

FIGURE 2-3-2

### A FOAF file for a *Person* instance with selected properties describing personal information, work place and educational background, plus another person one knows

```

- <rdf:RDF>
- <foaf:PersonalProfileDocument rdf:about="http://marciazeng.slis.kent.edu/marciaZeng.rdf">
  <foaf:maker rdf:resource="#me"/>
  <foaf:primaryTopic rdf:resource="#me"/>
- <foaf:PersonalProfileDocument>
- <foaf:Person rdf:ID="MarciaZeng">
  <foaf:name>Marcia Lei Zeng</foaf:name>
  <foaf:title>Dr.</foaf:title>
  <foaf:givenname>Marcia Lei</foaf:givenname>
  <foaf:family_name>Zeng</foaf:family_name>
  <foaf:nick>Marcia</foaf:nick>
  <foaf:homepage rdf:resource="http://marciazeng.slis.kent.edu"/>
  <foaf:workplaceHomepage rdf:resource="http://www2.kent.edu/slis/index.cfm"/>
  <foaf:schoolHomepage rdf:resource="http://www.ischool.pitt.edu"/>
- <foaf:knows>
- <foaf:Person>
  <foaf:name>Jian Qin</foaf:name>
  <rdfs:seeAlso rdf:resource="http://my.ischool.syr.edu/People/jqin"/>
  </foaf:Person>
</foaf:knows>
</foaf:Person>
</rdf:RDF>

```

The example in figure 2-3-2 shows various properties that are associated with *Person* class, including *name*, *title*, *givenName*, *familyName*, *nick*, *homepage*, *workplaceHomepage*, *schoolHomepage*, and *knows*. The property *knows* connects persons simply but effectively. Some of the properties are inherited from upper classes. For example, any *Thing* can have a *name*.

FOAF (prefix “foaf”) is one of the top five most-used metadata vocabularies found in the Linked Open Data (LOD) cloud (other most-used prefixes are “rdf,” “rdfs,” “dcterm,” and “owl”). More than 700 datasets (69.13%) in the LOD cloud have used FOAF, according to the report “State of the LOD Cloud 2014” (Schmachtenberg, Bizer, and Paulheim 2014).

### 2.3.4 Schema.org

In 2011, a group of search engines that included Bing, Google, Yahoo!, and Yandex released Schema.org ([schema.org/](http://schema.org/)), a collection of schemas for webmasters to use as the metadata vocabulary to markup their pages in ways that would be recognized by these search providers. The vocabulary refers to ontological classes as “item types.” The main item types under the broadest type, *Thing*, are listed below according to version 1.91:

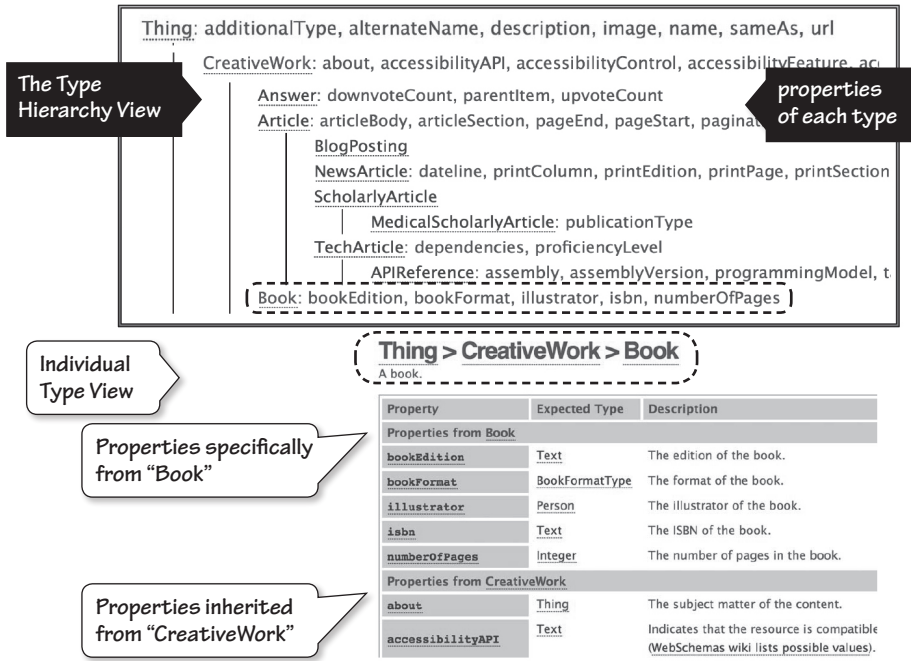
#### Thing

- ▶ *Creative works*
- ▶ *Event*
- ▶ *Intangible*
- ▶ *Health and medical types*
- ▶ *Organization*
- ▶ *Person*
- ▶ *Place*
- ▶ *Product*
- ▶ *Review*
- ▶ *Action*

Each item type has its own set of properties that can be used to describe the item. For example, *Thing* has a number of common properties: *name*, *alternativeName*, *description*, *url*, *image*, *potentialAction*, *sameAs*, and *additionalType*. This means all sub-classes of *Thing* automatically inherit these properties (figure 2-3-3).

*CreativeWork* as a sub-type of *Thing*, has more than thirty sub-types itself, including many that LAMs often deal with (*Article*, *Book*, *Dataset*, *Map*, *MediaObject*, *Movie*, *MusicRecording*, *Periodical*, *Photograph*, *Sculpture*, and *TVSeriesWebPage*). It also contains many more types, such as *Answer*, *Clip*, *Code*, *Comment*, *Diet*, *EmailMessage*, *Episode*, *ExercisePlan*, *MusicPlaylist*, *Question*, *Recipe*, *Review*, *Season*, *Series*, *SoftwareApplication*, *TVSeason*, *WebPage*, etc. The following example focuses on *Book*. In both the type hierarchy view and the individual type view, you can

**FIGURE 2-3-3** Explanation of the Type Hierarchy view and Individual Type view of “Book” type



Source: Compiled based on screenshots from Schema.org pages (<https://schema.org/docs/full.html> and <https://schema.org/Book>).

see that *Book* is the sub-type of *CreativeWork* and inherits the properties from its ancestors *CreativeWork* and *Thing* (*Thing* → *CreativeWork* → *Book*), in addition to possessing a few additional properties that are especially useful for the *Book* class.

The sub-types listed above represent just a small portion of this giant vocabulary. Because they are close to what LAMs have been describing—creative works—it is important to pay particular attention to them. If we further explore the other types, such as *Events*, *Action*, or *Intangible*, we will be even more surprised by the extent of its coverage (e.g., a type for *Volcano*). Schema.org is a vocabulary that has no boundaries, is updated often, and continues to grow.

The *Organization* class has a long list of sub-types—from *Airline* to *DryCleaningOrLaundry*, from *FinancialService* to *Locksmith*, from *GovernmentOffice* to *SportsTeam*—any type one can find on the web. Again, all these specific types of organizations inherit the properties of *Organization* (such as *address*, *aggregateRating*, *brand*, *contactPoint*, *faxNumber*) and may have additional ones of their own. For example, *MedicalClinic* has its own properties *availableService* and *medicalSpecialty*.

One frequently asked question is, “Why do search engines want to do this?” Aren’t there hundreds of specialized ontologies and element sets on the web already? Why don’t they incorporate/support other vocabularies such as FOAF, IPTC Core, Dublin Core, Music Ontology, etc.? Well, the answer from the Schema.org is that one of its goals was to create a single place where webmasters could go to figure out how to markup their content, with reasonable syntax and style consistency across types. Schema.org listed three types of users and the benefits they receive:

- First, for webmasters, schema.org provides webmasters with a single place to learn about markup, instead of having to graft together a schema from different sources, each with its own rules, conventions and learning curves. They only need to learn one thing rather than having to understand different, often overlapping vocabularies (Schema.org, [2013], <https://schema.org/docs/faq.html#0>). From the content point of view, Schema.org has been incorporating elements from various vocabularies and adding new types and properties collaboratively with other communities.
- Second, for search engines, the item types and properties that are most valuable to search engines are defined though Schema.org. This means search engines will get the structured information they need most to improve search.
- Third, for users, Schema.org considers that this will make it easier for webmasters to add markup, thus facilitating the search engines’ ability to see more of the markup they need to discover resources. The users will then end up with better search results and a better experience on the web (Schema.org, [2013]). By using Schema.org types with one’s custom search engine, a topical search engine can be created by a web site. For example, a topical search engine for movies will ensure that a search on *The Internship* will get results of the 2013 film and not potential career opportunities (Schema.org 2014).

WorldCat, the largest global catalog of library collections that covers bibliographic data of creative works, has already appended Schema.org descriptive markup to its catalog pages. It is the largest set of linked bibliographic data on the web. This exposure to search engines enables major search engines and other Web crawlers make use of WorldCat metadata in search indexes and other applications (OCLC 2012). It is also true that more and more LAM digital collections are considering and experimenting with publishing parallel datasets using Schema.org in order to expose the data that has been kept in local databases (i.e., silos, small or large) to search engines. This is another reason why this chapter introduces Schema.org.

## 2.4 RDF Vocabularies for Metadata Terms

Topics covered earlier in this chapter are the result of more than two decades' research and development (R&D) and therefore more or less well established. Newer concepts and developments that have emerged in the last few years characterize a gradual maturity of Semantic Web technologies and their impact on metadata R&D. Among these newer concepts and developments are two related topics that have significant implications for the LAM communities and beyond. One is the “RDF-ization” of metadata terms; the other is its impact, which is a revolutionary change of the metadata “record” concept. This section will discuss these two topics respectively after a brief explanation of RDF. The focus of the section requires us to revisit Dublin Core, but you will find out that today's Dublin Core has changed from an element set to an open-ended RDF vocabulary.

### 2.4.1 An Introduction to RDF (Resource Description Framework)

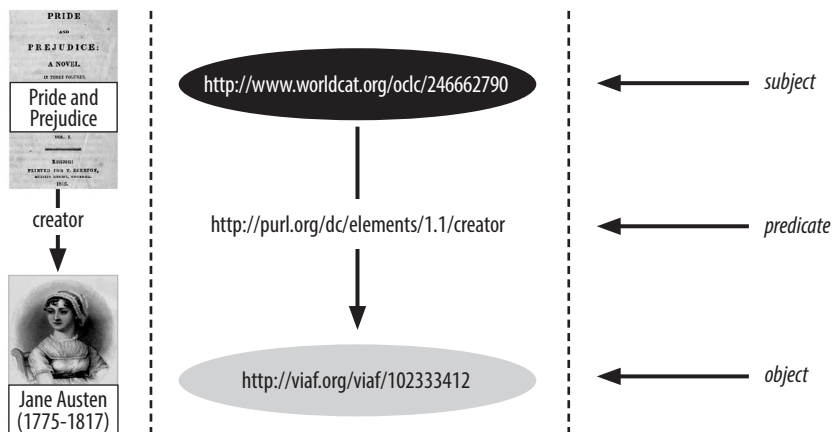
RDF (Resource Description Framework) is a framework for representing information on the web. “RDF has features that facilitate data merging even if the underlying schemas differ, and it specifically supports the evolution of schemas over time without requiring all the data consumers to be changed” (W3C 2014, [www.w3.org/RDF/](http://www.w3.org/RDF/)). It was originally created in 1999 as an XML-based standard for encoding metadata—data about data, particularly about web resources (see *Resource Description Framework (RDF) Model and Syntax Specification*. W3C Recommendation 22 February 1999.) With the development of the Semantic Web, RDF, as a model, has been playing an increasingly important role and has evolved in its own right as a standard for metadata description alone, especially since its updated specification in 2004. Now RDF has multiple syntaxes, including RDF/XML, RDFa, N-triples, Turtle, JSON-LD, along with others.

The greatest use of RDF is not restricted to encoding information about web resources; RDF is used to provide information about, and relations between, *things* in the real world: people, places, concepts, and the like. The RDF model is based on the principle of making logical statements about resources in the form of *subject-predicate-object* expressions (called *triples* in RDF terminology). An RDF graph can be visualized as a node and directed-arc diagram, in which each triple is represented as a node-arc-node link (W3C 2014). The RDF terminology for the various parts of the triple (middle of the figure) will be interpreted below in figure 2-4-1.



FIGURE 2-4-1

**Example of an RDF graph (center of the figure) that has two nodes (subject and object), with a predicate connecting them**



RDF is based on the idea of identifying things using web identifiers and describing resources in terms of simple properties and the values of the properties. Since the release of RDF 1.1 (W3C 2014), a more inclusive scheme than URI (Uniform Resource Identifiers) for identifiers is recommended. It is called IRI (Internationalized Resource Identifier). IRI was defined by the Internet Engineering Task Force (IETF) in 2005 to extend upon the existing URI scheme. URIs are limited to a subset of the ASCII character set. IRIs allow non-ASCII characters such as non-English characters in the Universal Character Set (Unicode) to be used in the IRI character string.

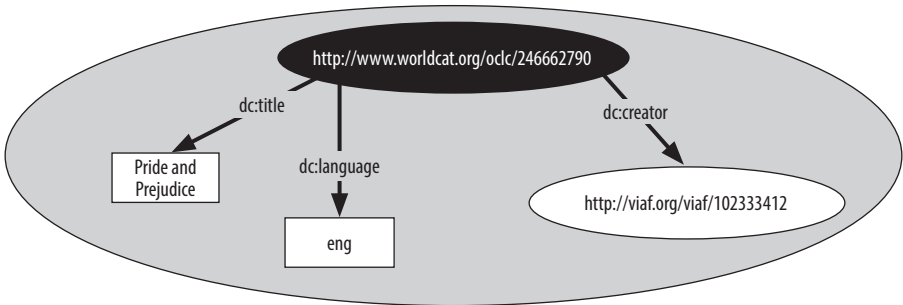
RDF uses the IRI as its basic mechanism for identifying *subjects*, *predicates*, and *objects* in statements, as seen from the example above (figure 2-4-1):

- ▶ The *subject* is a book (*Pride and Prejudice*), represented by an IRI given by WorldCat.
- ▶ The *predicate* uses “creator” as defined by Dublin Core, represented by an IRI.
- ▶ The *object* is author Jane Austen, represented by an IRI given by VIAF.

*The objects* in RDF triples may be either constant values represented by character strings (called *literals*) or URIs/IRIs (called *non-literals*), in order to represent different kinds of values. In figure 2-4-2, the nodes that are IRIs are shown as ellipses (e.g., the book IRI and creator IRI), and nodes that are literals are

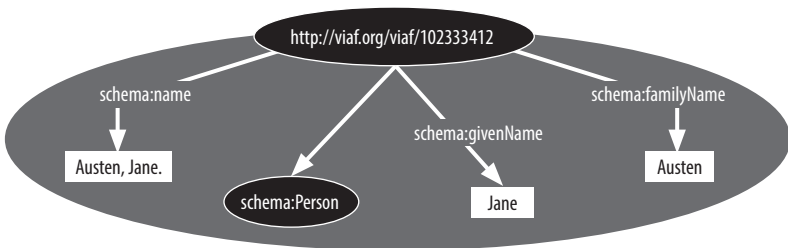
shown as boxes (e.g., title and language values in the figure). A group of triples form an RDF graph for a specific book (here its IRI is <http://www.worldcat.org/oclc/246662790>) and is expressed in the graph below (figure 2-4-2). The triples tell us about the book’s title, language, and creator.

**FIGURE 2-4-2** An illustration of a group of triples in an RDF graph for a book



You may have noticed that the value for the creator of the book has an IRI: <http://viaf.org/viaf/102333412>. When an *object* in a triple has its value in the form of an IRI, it means that it is a unique resource, and can be a *subject* itself, and hence has its own triples. In this case, as a *subject*, <http://viaf.org/viaf/102333412> has a group of triples, as illustrated in the next figure. The graph tells us that this is a person, has name “Austen, Jane,” given name “Jane,” and family name “Austen.” The *predicates* are the properties defined in Schema.org (figure 2-4-3).

**FIGURE 2-4-3** An illustration of a group of triples in an RDF graph for a person



Through a search query using SPARQL language, the two graphs can be connected, from:

<http://www.worldcat.org/oclc/246662790>—dc:creator → <http://viaf.org/viaf/102333412>  
to:

<http://viaf.org/viaf/102333412>— schema:name → Austen, Jane. (figure 2-4-4).

This illustration only gives a very limited number of triples to explain some of the concepts related to RDF. For a full set of metadata descriptions expressed with various RDF syntaxes, consult <http://www.worldcat.org/oclc/246662790> and check “Linked Data” section at the end of the web page provided by WorldCat.

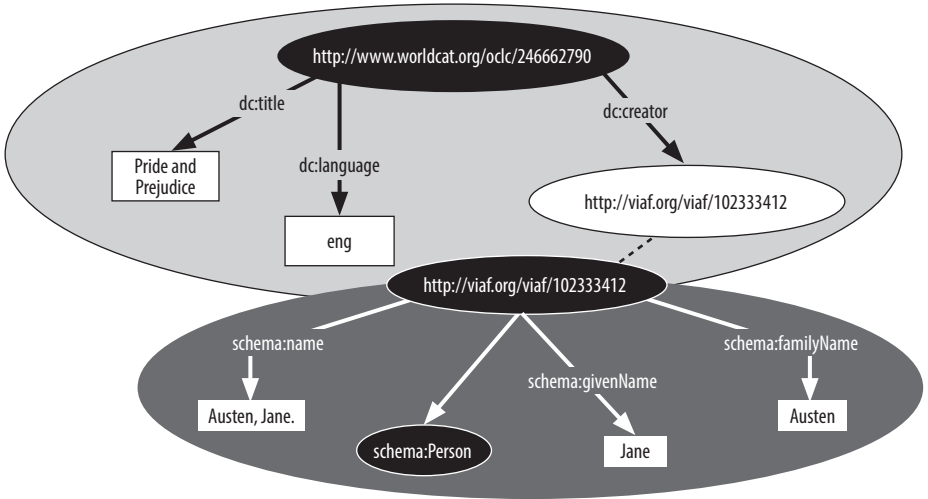
An RDF document is a document that encodes an RDF graph or RDF dataset in a concrete RDF syntax (or, in formal terms, *serialization format*). RDF documents enable the exchange of RDF graphs and RDF datasets between systems (W3C 2014). One of the syntaxes is RDF/XML (XML syntax for RDF). For the graph in figure 2-4-3 about Jane Austen (as a person), the triples using RDF/XML syntax will look like this (note: the first row lists the namespaces of *rdf* and *schema*):

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:schema="http://schema.org">
  <rdf:Description rdf:about="http://viaf.org/viaf/102333412">
    <rdf:type rdf:resource="http://schema.org/Person"/>
    <schema:name>Austen, Jane.</schema:name>
    <schema:familyName>Austen</schema:familyName>
    <schema:givenName>Jane</schema:givenName>
  </rdf:Description>
</rdf:RDF>
```

When converted to the Turtle syntax using a convertor, EasyRdf ([www.easyrdf.org/converter](http://www.easyrdf.org/converter)), the triples will look like this:

```
@prefix schema: <http://schema.org/> .
<http://viaf.org/viaf/102333412>
  a schema:Person ;
  schema:name "Austen, Jane." ;
  schema:familyName "Austen" ;
  schema:givenName "Jane" .
```

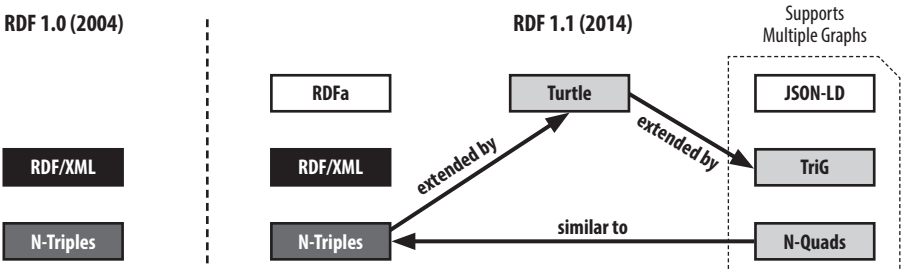
**FIGURE 2-4-4**  
**Illustration of two connected graphs**



In chapter 3, sections 3.7 and 3.9 will present more examples of RDF documents and machine-generated visualization results.

RDF serialization formats (i.e., syntaxes) have been extended significantly during the past ten years (figure 2-4-5). In addition to RDF/XML (demonstrated in the above Jane Austen example), there are more: RDFa (for HTML and XML embedding), the Turtle family of RDF languages (Turtle, TriG, and N-Quads), and JSON-LD (JSON-based RDF syntax) (W3C 2014; Manola, Miller, and McBride 2014).

**FIGURE 2-4-5**  
**RDF serialization formats at a glance**



Source: Generated based on “What’s New in RDF 1.1,” W3C First Public Working Draft (December 17, 2013) <http://www.w3.org/TR/2013/WD-rdf11-new-20131217/>.