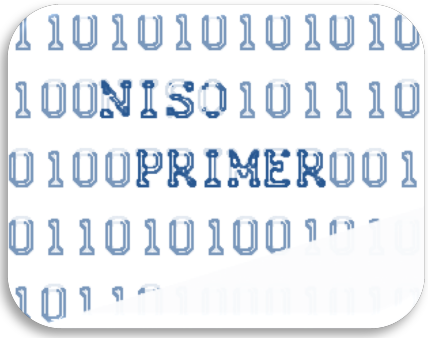
UNDERSTANDING METADATA

WHAT IS METADATA, AND WHAT IS IT FOR?

By Jenn Riley



A Primer Publication of the
National Information Standards Organization

About the NISO Primer Series

This NISO Primer Series is a four-part series of documents that provide introductory guidance to users of data (the other three documents cover research data). This primer discusses the latest developments in metadata and the new tools, best practices, and resources now available

For current information on the status of this publication contact the NISO office or visit the NISO website (www.niso.org).

Published by

National Information Standards Organization (NISO)
3600 Clipper Mill Road
Suite 302
Baltimore, MD 21211
www.niso.org

This publication is copyright © National Information Standards Organization (NISO), 2017. NISO is making this work available to the community under a Creative Commons Attribution-NonCommercial 4.0 International license. You are free to:

- Share — copy and redistribute the material in any medium or format
- Adapt — remix, transform, and build upon the material

Under the following terms:

- Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
- NonCommercial — You may not use the material for commercial purposes.

The complete license terms and conditions may be found here: http://creativecommons.org/licenses/by-nc/4.0/legalcode

All rights reserved under International and Pan-American Copyright Conventions. For noncommercial purposes only, this publication may be reproduced or transmitted in any form or by any means without prior permission in writing from the publisher, provided it is reproduced accurately, the source of the material is identified, and the NISO copyright status is acknowledged. All inquiries regarding translations into other languages or commercial reproduction or distribution should be addressed to:
NISO, 3600 Clipper Mill Road, Suite 302, Baltimore, MD 21211.

ISBN: 978-1-937522-72-8

Contents

Introduction

Consider how retailers store information about their products and their customers; employers about their employees and their operations; organizations about events they manage; research institutions about trends and notable people in their area; libraries, archives, and museums about the materials in their care; governments about their citizens, their allies, and their enemies—this is all metadata. Metadata, the information we create, store, and share to describe things, allows us to interact with these things to obtain the knowledge we need. The classic definition is literal, based on the etymology of the word itself—metadata is "data about data." With this broad definition, one might expect that metadata could be found everywhere, and in fact it is. Indeed, in 2013, *metadata* became a household term in the United States through heavy media coverage of the National Security Agency's collection of information on domestic telephone calls, including time and location initiated, duration, and number dialed.

Metadata in Everyday Life

Metadata is pervasive in information systems, and comes in many forms. The core features of most software packages we use every day are metadata-driven. People listen to music through Spotify; post photos on Instagram; locate video on YouTube; manage finances through Quicken; connect with others via email, text, and social media; and store lengthy contact lists on their mobile devices. All of this content comes with metadata—information about the item's creation, name, topic, features, and the like. Metadata is key to the functionality of the systems holding the content, enabling users to find items of interest, record essential information about them, and share that information with others.

Web pages often have metadata embedded in them. The links from one Web page to others and records of user behavior—selecting individual pages to view from among lists of search results, for example—are types of metadata as well. Web search engines build up vast indexes that use page text and its attendant metadata to provide relevant search results to users. Google goes even further. In 2012, it launched a "Knowledge Graph" with 3.5 billion "facts": metadata about 500 million people, places, and things, and the relationships between them.[1, 2] The Knowledge Graph and other structured metadata stored by Google are used to enhance search results and provide other value-added features such as sports scores, integration of search results with maps, and the knowledge cards that appear on the search results screen providing details on notable people and places.

Wikipedia, a free, crowdsourced online encyclopedia, both uses and generates metadata. The Wikimedia Foundation's Wikidata project is an open and collaboratively edited knowledge base similar to Google's Knowledge Graph. It stores factual information about topics in structured forms that can be pulled into Wikipedia articles or other information systems. The DBpedia project does the reverse, mining metadata from Wikipedia infoboxes, categories, images, geospatial information, and links to generate an open resource of structured metadata that can be reused in countless ways.

Metadata is vital for business transactions, too. Retailers must track many details about the products they carry, including price, source, inventory quantity, and descriptive information. This

[1] https://googleblog.blogspot.ca/2012/05/introducing-knowledge-graph-things-not.html
[2] http://searchengineland.com/google-launches-knowledge-graph-121585

is even more essential for online businesses; since online shoppers are unable to view items in person, they expect to be able to search by criteria such as keyword and object type, or to use facets to narrow a wide spectrum of products to a more manageable number. Businesses routinely store metadata about searches and transactions, which enables them to analyze sales trends, predict future demand, pay sales taxes owed to governments, and more. This same metadata allows businesses to provide a more personalized shopping experience, with features such as purchase history, address books for multiple shipping locations, and product recommendations. Manufacturers use metadata to track designs, parts, and materials, and to manage their research programs. The travel industry similarly relies on metadata about passengers, patrons, and bookings and about resources such as flights and hotel rooms. News media use metadata to track events, coverage, and published content. All businesses use metadata for human-resource functions such as hiring, payroll, and performance management.

Metadata in Action: Amazon and its Affiliates

Amazon.com has a worldwide online retail presence covering many different categories of goods. Metadata drives many parts of the company's operations. The metadata starts with the publishers of books or providers of other types of goods, as part of their own inventory systems. These suppliers send this metadata to Amazon, which integrates it with similar kinds of information from thousands of other providers to build its own website and sell products to users. Amazon collects metadata on sales and further uses it to provide customers with recommendations and optimize its relationships with suppliers. Amazon also makes the metadata about the products it brokers available to affiliate sites that build their own services on top of it, increasing sales through Amazon and driving business to the original supplier.

Metadata is at the core of social media platforms as well. Facebook users create metadata when managing friend lists, posting statuses or adding descriptions to media, "liking" friends' statuses, re-sharing already posted content, and contributing original media. By tracking these activities, Facebook analyzes trending topics and promotes sponsored posts that generate revenue. Pinterest users create boards that categorize and describe items of interest, with these categorizations and descriptions adding value to the links and serving as metadata for them. Pinterest then uses this socially generated metadata to build a search index and recommendations for content of interest to its members. Instagram users provide captions to images they upload and share, and follow other users' and business's accounts. Instagram uses this interaction data to improve its advertising. Twitter users organize the people they follow into lists, post text and media, use hashtags to comment on tweets and connect them to others, retweet others' content with or without commentary, and "favorite" tweets, driving features such as Twitter's trending-topics list. The depth of data about society represented by the content in Twitter and its metadata led to a

2010 agreement for the United States Library of Congress to archive this valuable material for research.[3]

These examples illustrate the somewhat fuzzy boundary between metadata and the information it describes. This distinction is irrelevant in many situations, as metadata is often created, stored, and acted upon largely as though it is data. Indeed, the distinction between metadata and data is in actuality solely one of semantics.

One feature the examples above share is that the metadata is all structured to some degree. The metadata is collected so that it can fulfill a useful purpose, and sorted into known categories. It is this notion of structure that turns raw information into actionable metadata. Specific elements are collected and stored in such a way as to show them in either administrative or public-facing interfaces with explanatory labels. *Properties* or *elements* are common terms for these labels, though the names vary by user community.

[3] https://blogs.loc.gov/loc/2010/04/how-tweet-it-is-library-acquires-entire-twitter-archive/

Metadata in the Cultural Heritage World

The cultural heritage world—libraries, archives, and museums—has a long history of creating and sharing robust, structured metadata. For libraries, this takes the form of the library catalog, which has evolved over centuries. Early library catalogs were merely large inventory books, which then were replaced by catalog cards in drawers. With computerization, libraries first moved to dedicated search terminals, and then in the Internet era to today's Web-based resource discovery systems. Libraries take a "bibliographic" approach to metadata, which is rooted in their traditional strength in describing books. Bibliographic metadata focuses on detailed descriptions of individual items that allow users to locate these items. Archives use "finding aids," descriptive inventories of collections, along with historical information necessary for understanding the material. For archives, metadata helps users locate groups of related items that arise from the regular work of an individual (which are called papers) or organization (which are called records), and that contain material best understood in the context of that grouping. Museums track detailed information about their acquisitions, exhibits, and loans. Museum curators use metadata as a way to interpret collections, conveying to visitors artifacts' historical and societal significance, as well as describing the relationship of objects to one another.

Cultural heritage metadata focuses heavily on descriptive information. For books, whether they are print or electronic, title, author, publication, and subject details predominate. For musical, film, and art works, title, creator, genre, and performance information are typically recorded. For archival papers and records, details of their creation and relationships among them are most important. Information about the creators of these works and their lives is also commonly recorded as metadata in cultural heritage organizations.

Types of Metadata

This type of metadata—information about the content of a resource that aids in finding or understanding it—is referred to as descriptive metadata. The cultural heritage community distinguishes descriptive metadata from other types. Administrative metadata is an umbrella term referring to the information needed to manage a resource or that relates to its creation. Within the administrative metadata sphere is technical metadata, information about digital files necessary to decode and render them, such as file type; preservation metadata supporting the long-term management and future migration or emulation of digital files, for example, a checksum or hash; and rights metadata, such as a Creative Commons license, which details the intellectual property rights attached to the content. Descriptive and administrative metadata are considered distinct from structural metadata, which describes the relationships of parts of resources to one another; examples include pages in a sequence, a table of contents with pointers to the beginnings of milestone sections, and connecting different resolutions or bit depth representations of identical content.

Types of Metadata

Descriptive metadata	For finding or understanding a resource
Administrative metadata - Technical metadata - Preservation metadata - Rights metadata	 - For decoding and rendering files - Long-term management of files - Intellectual property rights attached to content
Structural metadata	Relationships of parts of resources to one another
Markup languages	Integrates metadata and flags for other structural or semantic features within content

A final category of metadata is *markup languages.* These languages mix metadata and content together, a practice only sometimes used with other forms of metadata. Flags inserted in the content denote notable features. For a textual resource, this might mean marking structural elements such as paragraphs; flagging words with semantic information—that the word is a place name or a certain part of speech, for example; or providing formatting information, such as italics.

These various categories of metadata support different use cases in information systems. Discovery is perhaps the most common, with structured metadata allowing users to search for or browse to find resources or information of interest. Many metadata properties are useful to display to users to aid in identification or understanding of a resource. Interoperability, the effective exchange of content between systems, relies on metadata describing that content so that the systems involved can effectively profile incoming material and match it to their internal

structures. Metadata supports digital-object management by providing the information needed to render digital content appropriately or deliver the appropriate version to match a user need. Preservation is achieved through creating metadata that allows the verification of the integrity of content after transfer and at other notable points, and signaling when preservation actions such as a format migration or an integrity check should be undertaken. Finally, metadata supports navigation within parts of items, for example, from one page or section to the next, and among different versions of objects, such as varying resolutions of photographic images.

Metadata Type	Example Properties	Primary Uses
Descriptive metadata	Title Author Subject Genre Publication date	Discovery Display Interoperability
Technical metadata	File type File size Creation date/time Compression scheme	Interoperability Digital object management Preservation
Preservation metadata	Checksum Preservation event	Interoperability Digital object management Preservation
Rights metadata	Copyright status License terms Rights holder	Interoperability Digital object management
Structural metadata	Sequence Place in hierarchy	Navigation
Markup languages	Paragraph Heading List Name Date	Navigation Interoperability

How is Metadata Stored and Shared?

Relational Databases

Metadata can be found in a variety of forms and encodings. In traditional information systems design, it might be stored as fields in relational database tables. A collection of metadata in this context is known as a *record*. Effective design in this model is based on appropriate normalization of database tables to maximize storage efficiency, balanced with optimization for query performance. Most types of metadata can be stored in this way. In this scenario, metadata would typically be loaded in a batch through custom processes or entered manually through purpose-built user interfaces, each controlled with custom programming. Today, software systems that use this metadata model and wish to share their metadata with others commonly do so through Application Programming Interfaces (APIs), publishing specification documents that external software developers can use to build tools that query the system and retrieve metadata of interest.

XML

In the 2000s, XML (eXtensible Markup Language) emerged as a commonly used encoding, transfer, and occasional internal system storage mechanism for metadata. Metadata in XML exists as sets of files, called XML documents. XML defines elements, tags that signify that the values inside them have a certain meaning. Elements can also have other elements inside them, and it is from this feature that XML documents gain their structure. An XML document is a tree that begins with a single root element. Other elements and values then branch out from this original root, building a nested structure that contributes to the meaning of the metadata values in the document. XML elements can take attributes, which typically also have their own values. An XML attribute and its value refine the meaning of the element in which they appear. XML supports multilingualism of metadata by providing a predefined attribute to indicate the language in which an element's value appears. As with relational databases, an XML document describing a defined thing is known as a metadata record. See below for an example of such a record.

A Simple XML Record Example

```
<?xml version="1.0" encoding="UTF-8"?>
<work type="play">
  <workName>The Tempest</workName>
  <writtenBy>
      <playwright>
         <playwrightName>William Shakespeare</playwrightName>
         <bornInPlace>Stratford Upon Avon</bornInPlace>
      </playwright>
   </writtenBy>
</work>
```

`<work>`, `<playwright>`, and `<bornInPlace>` are examples of XML elements
Stratford Upon Avon is an example of an element's value
`type="play"` is an example of at attribute name (type) and value (play)

Specialized languages (most notably XPath, XSLT, and XQuery) exist for transforming and querying XML documents, as do XML processing toolkits for the major programming languages. Effective XML design centers around good choices in balancing the use of elements and attributes, and attention to document size to promote adequate query performance. Metadata stored as XML in systems is often loaded directly from external sources, or in other cases might be generated through software user interfaces or mapped en masse from other data sources. In many cases, XML data is ingested into a system that renders it into other forms for storage and indexing, though native XML databases do exist. The use of XML is not limited to descriptive metadata; many different types of metadata can be stored in XML documents.

Cultural heritage institutions have a long history of sharing metadata, dating back to the United States Library of Congress distributing catalog cards (primarily for books) to local libraries. In the early 2000s, the cultural heritage community entered a new phase of cooperation when it began to share XML-based metadata through the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH). With OAI-PMH, the types of material for which metadata was routinely shared by libraries, archives, and museums expanded greatly to include material such as photograph collections and pre-prints of research papers produced at universities. For a time, Google supported and fueled the use of OAI-PMH as part of its Sitemaps protocol, though this support was retired in 2008. While OAI-PMH is still in use by the institutional repository and digital collections communities due to its implementation in common software packages, such as DSpace, its limitations are well known. ResourceSync, a successor protocol that operates within the XML Sitemaps specification, has gained some traction, though more recently, Linked Data shows stronger promise for the future of sharing of metadata.

Linked Data and RDF

The concept of Linked Data was introduced by Tim Berners-Lee, who is widely regarded as the

inventor of the World Wide Web, in 2006. Implementation of this idea involves organizations publishing their structured data on the Web, explicitly naming entities in this data so they can be referenced by others, and linking to others' data to build a worldwide information network. This work has become the most successful practical step towards implementing the "Semantic Web," a vision for a worldwide network of actionable data. In recent years, the World Wide Web Consortium (W3C) has taken a leadership role in expanding the original vision for the Semantic Web to become an initiative for "Building the Web of Data," including Linked Data as a key part of the plan.

Linked Data—Design Issues
Tim Berners-Lee, 2006
http://www.w3.org/DesignIssues/LinkedData.html

1. Use URIs as names for things.
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL).
4. Include links to other URIs, so that they can discover more things.

Linked Data in operation relies heavily on RDF (Resource Description Framework) standards. RDF is a set of W3C specifications designed for metadata on the Semantic Web. Whereas XML models information as a tree, RDF models it as a graph, with small bits of information each connected to other small bits of information. No one entity or piece of data has primary importance in a graph; the network of information can be accessed equally at any point. As such, the concept of a metadata record, the sum total of information known about a single entity or a defined set of data elements intended to travel together, as used in relational databases and XML, does not fit well in the RDF model. An RDF graph is best viewed as a whole, or as a simplified subset used in a given context for a given purpose. Graphs are made up of individual triples, where a subject (the entity the triple is about) is connected to an object (the entity it's related to) with a predicate (a descriptor of a relationship). A triple might represent content such as: "*The Tempest*" (subject) "was written by" (predicate) "William Shakespeare" (object).

In RDF, all subjects and some objects are modeled as classes, which describe types of resources. RDF conventions dictate that class names start with a capital letter and each subsequent word in the label for the class also starts with a capital letter, with no spaces between words. There are endless possibilities for RDF classes; some examples include Person, Book, Painting, Building, Event, and PhilosophicalIdea. RDF predicates are modeled as properties, which describe relationships. Virtually any relationship concept can be expressed as an RDF property, though relationships exemplifying descriptive metadata are most common. Property names start with a lower case letter; each subsequent word in the label for the property starts with an upper case

letter, with no spaces between words. Examples include createdBy, memberOf, successorTo, occurredAtTime, sameAs, and familyName.

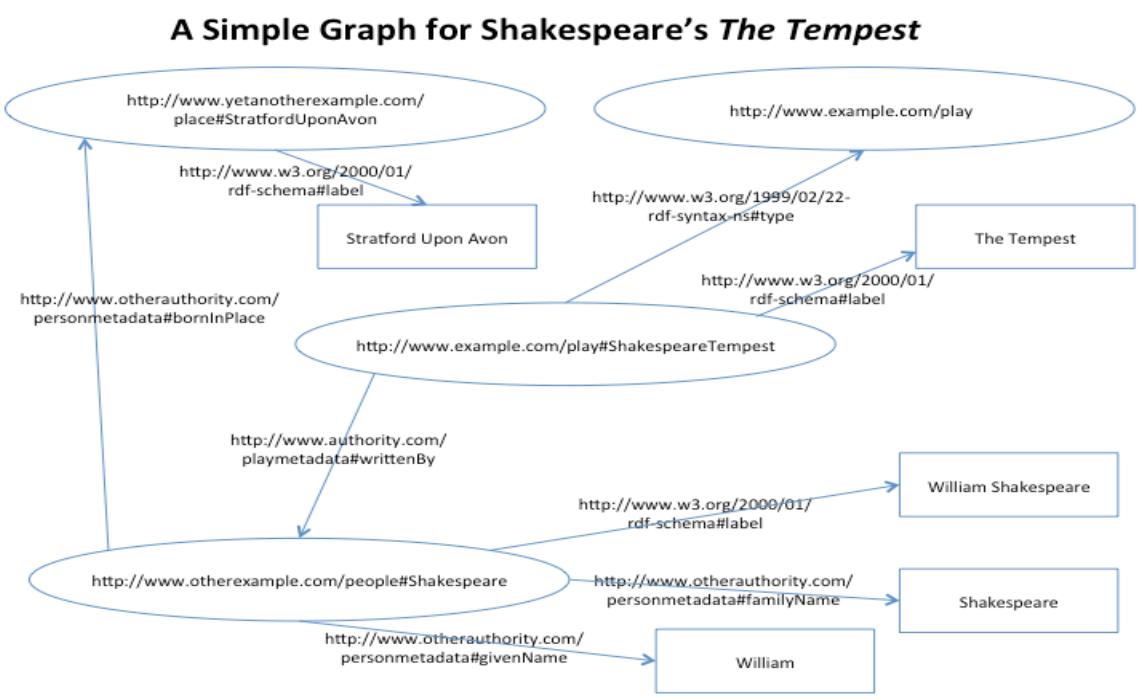
RDF properties can be defined with a domain, which indicates the subject of a triple is a member of a specified class. Similarly, defining a range of a property indicates that the object of a triple is a member of a specified class. Domains and ranges serve a dual purpose: first, to guide implementers as to how a particular property should be used, and second, to allow processing tools to derive new RDF relationships out of the connections implied by these definitions. For example, if the property createdBy is defined with a domain of Book and a range of Person, a system encountering this triple can assume the subject is of class Book and the object is of class Person, even if there are no known explicit RDF triples making those claims.

Subjects, predicates, and sometimes objects are represented by Uniform Resource Identifiers (URIs) or International Resource Identifiers (IRIs; URIs that allow use of non-ASCII characters) in RDF. This structured naming of classes and properties allows them to be referred to by other triples. These additional triples could make additional factual statements about resources, for example, stating that this book assigned an author in one triple was also written in this year or is of the type "fiction." This structure is also used to connect classes and properties to other classes or properties, for example, that a Book is a type of CreativeWork, or that a property defining the concept of authorship of a work created by one institution or group is semantically identical to another property for the same purpose created by a different institution or group. There is no expectation in RDF that the subject, predicate, and object in a triple all be defined by the same community; indeed, the power of Linked Data is to connect both data and vocabularies from multiple sources. It is these features that allow the RDF graph to grow using worldwide input. Objects can also be free-text strings, known in RDF as *literals.* RDF standards allow literals to be marked as being in a specific language or as conforming to a defined data type.

RDF Schema (RDFS) is the core RDF technology used for creating RDF languages. RDFS is used to formally define classes and properties, datatypes for objects, domains and ranges for properties, and hierarchical relationships between classes and subclasses, or properties and subproperties.

It is Linked Data best practice for URIs to be *dereferenceable*. This means that URIs should be actionable via HTTP so that both humans and machines can see useful information such as labels, definitions, and relationships to other resources when visiting the HTTP URI for an RDF-encoded concept. The process of content negotiation is used to provide human users with Web pages and software applications with raw data when they visit the same URI. This allows RDF-aware software applications to make use of classes and properties with which they are unfamiliar, and makes RDF-encoded Linked Data a powerful tool for connecting information from multiple sources.

An RDF Graph Example



RDF data can be shared in a variety of ways. It is highly compatible and commonly used with the microformats approach that embeds structured metadata inside HTML or XHTML using the itemprop attribute, for sharing with search engines or other tools that process data from websites. Alternatively, large RDF triplestores (software optimized for storage and retrieval of RDF data) often provide endpoints for remote searching using the RDF query language SPARQL through an API. This allows data from these triplestores to be used in third-party applications.

Additional RDF syntaxes are in wide use as Linked Data. RDF/XML is a serialization of a bounded RDF graph using the structure of XML. RDF/XML is a particularly verbose encoding of RDF, which reduces its human readability. While RDF/XML has been around longer than other RDF serializations, it is less popular with software developers. RDFa (Resource Description Framework in Attributes), like microformats, embeds RDF within HTML through the use of HTML attributes that have no impact on the rendering of a page through a Web browser, and serve instead to provide metadata about the content to systems reading the page. Turtle (Terse RDF Triple Language) is a far more compact RDF serialization that reuses some structure from the SPARQL query language. Turtle is a textual representation of an RDF graph that allows multiple predicates of the same subject to be presented in a compact way. N-Triples is another text-based RDF serialization. It is a subset of Turtle, limited to more structured and predictable syntax. JSON-LD (JavaScript Object Notation for Linked Data) is an RDF serialization that

builds on the software developer-friendly JSON format, which uses simple key-value pairs to record information. It is commonly used in simple Web services that provide short snippets of data in response to queries.

RDF Graph: RDF/XML Serialization

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:ex="http://www.example.com/"
   xmlns:play="http://www.authority.com/playmetadata#"
   xmlns:person="http://www.otherauthority.com/personmetadata#">
  <ex:play rdf:about="http://www.example.com/play#ShakespeareTempest">
    <rdfs:label>The Tempest</rdfs:label>
    <play:writtenBy>
      <rdf:Description
rdf:about="http://www.otherexample.com/people#Shakespeare">
        <rdfs:label>William Shakespeare</rdfs:label>
        <person:givenName>William</person:givenName>
        <person:familyName>Shakespeare</person:familyName>
        <person:bornInPlace>
          <rdf:Description
rdf:about="http://www.yetanotherexample.com/place#StratfordUponAvon">
             <rdfs:label>Stratford Upon Avon</rdfs:label>
          </rdf:Description>
        </person:bornInPlace>
      </rdf:Description>
    </play:writtenBy>
  </ex:play>
</rdf:RDF>
```

RDF Graph: Turtle Serialization

```
@prefix person: <http://www.otherauthority.com/personmetadata#> .
@prefix play: <http://www.authority.com/playmetadata#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

<http://www.example.com/play#ShakespeareTempest> a
<http://www.example.com/play> ;
    rdfs:label "The Tempest" ;
    play:writtenBy <http://www.otherexample.com/people#Shakespeare> .

<http://www.otherexample.com/people#Shakespeare> rdfs:label "William
Shakespeare" ;
    person:bornInPlace
<http://www.yetanotherexample.com/place#StratfordUponAvon> ;
    person:familyName "Shakespeare" ;
    person:givenName "William" .

<http://www.yetanotherexample.com/place#StratfordUponAvon> rdfs:label
"Stratford Upon Avon" .
```

RDF Graph – N-Triples Serialization

```
<http://www.example.com/play#ShakespeareTempest>
<http://www.w3.org/2000/01/rdf-schema#label> "The Tempest" .
<http://www.example.com/play#ShakespeareTempest>
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://www.example.com/play> .
<http://www.otherexample.com/people#Shakespeare>
<http://www.otherauthority.com/personmetadata#familyName> "Shakespeare"
.
<http://www.otherexample.com/people#Shakespeare>
<http://www.otherauthority.com/personmetadata#bornInPlace>
<http://www.yetanotherexample.com/place#StratfordUponAvon> .
<http://www.yetanotherexample.com/place#StratfordUponAvon>
<http://www.w3.org/2000/01/rdf-schema#label> "Stratford Upon Avon" .
<http://www.otherexample.com/people#Shakespeare>
<http://www.otherauthority.com/personmetadata#givenName> "William" .
<http://www.otherexample.com/people#Shakespeare>
<http://www.w3.org/2000/01/rdf-schema#label> "William Shakespeare" .
<http://www.example.com/play#ShakespeareTempest>
<http://www.authority.com/playmetadata#writtenBy>
<http://www.otherexample.com/people#Shakespeare> .
```

RDF Graph – JSON-LD Serialization

```
{
  "@context": {
    "person": "http://www.otherauthority.com/personmetadata#",
    "play": "http://www.authority.com/playmetadata#",
    "rdfs": "http://www.w3.org/2000/01/rdf-schema#",
  },
  "@graph": [
    {
      "@id":
"http://www.yetanotherexample.com/place#StratfordUponAvon",
      "rdfs:label": "Stratford Upon Avon"
    },
    {
      "@id": "http://www.otherexample.com/people#Shakespeare",
      "person:bornInPlace": {
        "@id":
"http://www.yetanotherexample.com/place#StratfordUponAvon"
      },
      "person:familyName": "Shakespeare",
      "person:givenName": "William",
      "rdfs:label": "William Shakespeare"
    },
    {
      "@id": "http://www.example.com/play#ShakespeareTempest",
      "@type": "http://www.example.com/play",
      "play:writtenBy": {
        "@id": "http://www.otherexample.com/people#Shakespeare"
      },
      "rdfs:label": "The Tempest"
    }
  ]
}
```

An additional method for storing and sharing metadata is embedding it in a digital file itself. Virtually all file format specifications include a metadata area, often primarily for technical metadata about the file, used for decoding and rendering. Software that creates these files, such as that in digital cameras, generates and embeds this metadata inside the file. Yet many well-known image, media, and document file formats, such as JPEG, WAV, PDF, and Microsoft Office files, also provide for the recording of descriptive metadata. Tools that allow users to create and edit these files typically also allow editing of this descriptive metadata. While embedded metadata must be extracted into external systems for indexing, it has the advantage of keeping the metadata with the file it describes, ensuring the file is understandable in new contexts.

Standardizing Metadata

Metadata is only useful if it is understandable to the software applications and people that use it. To aid in this understanding, organizations frequently predefine metadata sets to meet certain needs, and publish these definitions for system designers (and sometimes end users) to consult. XML metadata vocabularies are known as *schemas*, *element sets*, or sometimes *formats*. An XML Schema defines the elements that make up a valid document in that format, along with the attributes each element can take, in what order they can appear, and how many times they can appear. XML Schemas can be formally standardized through organizations such as the International Organization for Standardization (ISO), the National Information Standards Organization (NISO), or the World Wide Web Consortium (W3C). Industry- or community-leading bodies such as the Library of Congress can also often serve as organizational homes and maintenance organizations for XML-based metadata standards, endorsing them for use in their target communities. XML Document Type Definitions (DTDs) are an older technology than XML Schema, and are currently in only limited use.

The situation is somewhat different with documenting and sharing RDF metadata specifications. RDF languages are typically known as *vocabularies*, which refer to definitions of both classes and properties. Formal standardization is not the norm for most RDF vocabularies; instead, communities tend to build vocabularies that are useful to them, then promote their use through a combination of documentation and open sharing of data that uses these vocabularies.

Both XML and RDF use the concept of *namespaces* to indicate which vocabulary a given element, attribute, class, or property a given term comes from. In both, a *namespace prefix* is used to stand in for a URI or IRI to streamline the syntax of the metadata. To fully process the metadata, the namespace prefix must be extended to the full URI or IRI and added to the specific element, class, or property being used.

Expanding Namespaces in RDF

The namespace prefix `rdf:` is shorthand for...	...this URI.
`rdf:`	http://www.w3.org/1999/02/22-rdf-syntax-ns #
Therefore anything that follows `rdf:`...	...gets added after the URI when the data is processed.
`rdf:type`	http://www.w3.org/1999/02/22-rdf-syntax-ns#type

In XML, a Schema has a default namespace that provides a "home base" for the elements and attributes defined by the Schema. All elements in an XML document without an explicitly declared namespace are defined as being in the default namespace. Elements and attributes from other namespaces can be formally brought into an XML Schema as desired and therefore become a part of the XML language defined by that Schema. While there are mechanisms for XML documents allowing arbitrary elements from other namespaces to be used, this feature is rarely employed as there are few mechanisms built into XML technologies to allow processors to make sense of unfamiliar vocabularies.

In RDF, namespaces are a fundamental part of the architecture. Whereas in XML the default namespace is assumed and elements from other namespaces are the exception, in RDF the assumption is that a class or property from any namespace can be used at any point. None is given priority over another. Indeed, it is quite normal for the subject, predicate, and object of a triple to be URIs from different namespaces. The very design of RDF promotes mixing and matching vocabularies in the graph, and the goal of Linked Data is to connect as many data points from as many different sources as possible. The best practice of allowing URIs or IRIs to be dereferenced to discover additional information about a subject, object, or predicate is designed to assist software applications in processing data in previously unknown namespaces.

Controlled vocabularies

In addition to standardizing syntax, metadata designers often wish to standardize metadata through control of the actual values used. One way in which this is done is through the use of controlled vocabularies. A controlled vocabulary is a predetermined list of terms on a certain topic or of a certain type. These lists typically identify one preferred word or phrase for a given concept, and sometimes provide mappings from other terms for the concept to the preferred one. They also frequently define (often hierarchical) relationships among terms.

Controlled vocabularies typically have in scope only a single language, though large communities occasionally work to link existing controlled vocabularies in different languages or define new controlled vocabularies to bring together terms from different languages. Controlled vocabularies can be exceedingly simple, using only a dozen or so terms, or much more robust, including as many as tens of thousands. Examples of controlled vocabularies include Internet MIME types, Spotify genres, the Book Industry Standards and Communications (BISAC) vocabulary, and Library of Congress Subject Headings (LCSH). Those developed and maintained by the cultural heritage community, such as LCSH, tend to be among the most robust controlled vocabularies in common use.

In XML implementations, the term selected from a controlled vocabulary typically appears as the value of an element, and an attribute in this case would exist to indicate the vocabulary from which the term is selected. In RDF, the controlled term would be referred to in the metadata with a URI or IRI rather than a textual string as the object of a triple. This URI or IRI would then be

dereferenceable to provide further information about the term, the vocabulary it comes from, and its relationship to other terms.

Content standards

A second method for standardizing the values that appear in metadata is the use of content standards, which are sets of guidelines that dictate how textual values in metadata should be structured. They are common as formal guidelines documents in the cultural heritage community. In other communities, where they can be known as style guides, they tend to be briefer and more informal.

Content standards typically cover topics such as where the information to be recorded should be found; what punctuation, capitalization, and abbreviations should be used; and how to make decisions about which information to record. They sometimes define and dictate the use of small controlled vocabularies as well. Examples of content standards include the Wikipedia Manual of Style guidelines for Infoboxes; Describing Archives: A Content Standard (DACS) and Rules for Archival Description (RAD) for archives; and the library community's Anglo-American Cataloging Rules, second edition (AACR2) and its successor, Resource Description and Access (RDA). For XML-based metadata, content standards control the values entered for XML elements. In RDF, content standards apply to literal values of objects of triples.

Notable Metadata Languages: Examples in Broad Use

Schema.org

On the open Web, Schema.org is among the most visible metadata vocabularies. Launched by the major search engines in 2011, Schema.org is an RDF vocabulary that allows creators to mark up semantics within the text of Web pages, enhancing the ability of systems to do interesting things with this content. The vocabulary is managed through a community governance process. As of 2016, the Schema.org home page claims that it is used on more than 10 million websites.

Schema.org defines nearly 600 "types" (which are defined as RDF classes) and over 800 properties. It also promotes the use of extensions to expand the vocabulary for use by specialized communities. Some of the most used Schema.org types are for data related to creative works; embedded objects; events; organizations; people, places and businesses, products and transactions; reviews and ratings; and actions. Its scope is largely descriptive. Each of these high-level categories includes progressively detailed subcategories, for example:

CreativeWork->Article->ScholarlyArticle

or

Place->CivicStructure->GovernmentBuilding->CityHall.

Each Schema.org class has a number of defined properties. For example, the class Article is associated with the property wordCount, and for the class CivicStructure, the property openingHours can be used.

Schema.org Example: Extract from North Carolina State Libraries' Home Page, Microformat Syntax Inside HTML

```
<div id="f-address">
      <h2>Contact</h2>
      <h3 itemscope itemtype="http://schema.org/Library"
itemprop="name"><link itemprop="logo" h
         ref="//www.lib.ncsu.edu/website/images/logo_small.png"><a
href="//www.lib.ncsu.edu/" id="f-contact-hill" itemprop="url">D. H.
Hill Library</a></h3>
      <p>
```

```
            <span itemprop="address" itemscope
itemtype="http://schema.org/PostalAddress">
                  <span itemprop="streetAddress">2 Broughton
Drive</span> <br>
                  <span itemprop="postOfficeBoxNumber">Campus Box
7111</span> <br>
                  <span itemprop="addressLocality">Raleigh</span>,
                  <span itemprop="addressRegion">NC</span>
                  <span itemprop="postalCode">27695-7111</span> <br>
            </span>
            <span itemprop="telephone"><a href="tel:9195153364" id="f-
hill-phone" data-attr="phone">(919) 515-3364</a></span>
      </p>
</div>
```

Schema.org Example: Extract from Data Representing Nate Silver's *The Signal and the Noise* in OCLC WorldCat Linked Data, Turtle Syntax

```
@prefix schema: <http://schema.org/> .
@prefix void:  <http://rdfs.org/ns/void#> .

<http://experiment.worldcat.org/entity/work/data/1172581839#Person/silv
er_nate_1978>
        a                  schema:Person ;
        schema:birthDate   "1978" ;
        schema:name        "Silver, Nate, 1978-" .

<http://worldcat.org/entity/work/id/1172581839>
        a                       schema:CreativeWork , schema:Book ;
        void:inDataset          <http://purl.oclc.org/dataset/xwc> ;
        schema:about
<http://experiment.worldcat.org/entity/work/data/1172581839#Topic/knowl
edge_theory_of> ,
<http://experiment.worldcat.org/entity/work/data/1172581839#Topic/bayes
ian_statistical_decision_theory> ,
<http://experiment.worldcat.org/entity/work/data/1172581839#Topic/forec
asting_history> , <http://id.loc.gov/authorities/subjects/sh85050485> ,
<http://id.loc.gov/authorities/subjects/sh85072732> ,
<http://id.worldcat.org/fast/988194> ;
        schema:creator          <http://viaf.org/viaf/256089470> ,
<http://experiment.worldcat.org/entity/work/data/1172581839#Person/silv
er_nate_1978> ,
<http://experiment.worldcat.org/entity/work/data/1172581839#Person/silv
er_nate> , <http://id.loc.gov/authorities/names/n2012043409> ;
        schema:description     "Silver built an innovative system for
predicting baseball performance, predicted the 2008 election within a
hair's breadth, and became a national sensation as a blogger. Drawing
on his own groundbreaking work, Silver examines the world of
prediction."@en ;
        schema:genre           "History" , "Nonfiction" ;
        schema:name            "The signal and the noise : why so many
predictions fail-- but some don't"@en ;
```

The focus of Schema.org on the semantics of the text within Web pages helps take metadata processing on the Internet to a higher level. In the early days of the Web, HTML markup provided basic information on the title, author, and subject of a page, to assist search engines in retrieving that page in response to a user's query. More recently, search engines are focused on accumulating knowledge in addition to merely indexing pages, as evidenced by Google's Knowledge Graph initiative. Schema.org is designed to further this use case, as it promotes the encoding of small but vital bits of knowledge within Web pages; for example, a Schema.org description can note that a certain building is located at certain geographic coordinates. It uses the existing technologies driving the Web to encode the building blocks of human knowledge in a structured and machine readable way.

While the Schema.org vocabulary was originally designed to mark up semantic content in Web pages, it has become ubiquitous enough to serve as a backbone for metadata shared in various ways on the Semantic Web, including through bulk downloads and information stored in triplestores and then made available for querying by external systems via SPACQL. The classes and properties defined in Schema.org can be found in any number of Linked Data applications. One notable example is OCLC, the not-for-profit library cooperative that is working to make library data more integrated into the open Web.

OCLC's strategy is to expose as Linked Data the valuable metadata created by libraries over the course of their history. The cooperative's Linked Data implementation used Schema.org as a core part of the vocabulary for its initial April 2014 launch of 197 million open bibliographic descriptions of books and other creative works.[4, 5] In OCLC's shared metadata, the Schema.org vocabulary is supplemented by classes and properties selected from other RDF vocabularies in common use and some classes and properties OCLC has newly defined to meet library needs. Here again, this metadata is designed to make connections that grow the Linked Data graph. For example, OCLC's definitions of new classes for the types of resources that appear in library collections, such as newspapers or musical scores, include triples that define these classes as subclasses of Schema.org's CreativeWork. In this way, any software traversing the Linked Data graph will infer that resources such as newspapers or musical scores can be understood as creative works, even if no metadata provider has explicitly made this connection.

Web Ontology Language (OWL)

An early, foundational tool for the Semantic Web is OWL (Web Ontology Language). It exists in two forms—as RDF/XML (mandatory for all OWL2 tools to support) and as a functional style syntax. While there are several ways for vocabulary designers to formally document RDF classes and properties, OWL is among the most used. It is designed to represent formal semantics in a machine-readable way, and to promote machine reasoning on shared RDF data. As such, OWL implementers can use the language for core Semantic Web tasks, such as defining individuals as

[4] https://www.oclc.org/en-US/news/releases/2014/201414dublin.html
[5] https://www.oclc.org/worldcat/data-strategy.en.html

members of a class, documenting two classes or individuals as equivalent, describing how two classes interact or complement each other, or indicating when membership in one class excludes that in another ("disjointness").

OWL exists in several forms of varying complexity; one is OWL Lite, which most implementers find sufficient for their needs. RDF vocabulary designers will frequently take the time to create a full OWL ontology document in RDF/XML that uses the owl:Ontology feature to provide basic descriptive and administrative information about the ontology itself. This information can include the ontology's name, designers, and relationship to other ontologies; the document then proceeds to define classes, properties, and their relationships using the OWL language along with mechanisms from RDF and RDFS.

OWL Example: Extract from Pizza Ontology used for Training with the Ontology Software *Protege*, RDF/XML Syntax

```
<owl:Class rdf:about="#Pizza">
  <rdfs:label xml:lang="en">Pizza</rdfs:label>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasBase"/>
      <owl:someValuesFrom rdf:resource="#PizzaBase"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#Food"/>
  <owl:disjointWith rdf:resource="#PizzaTopping"/>
</owl:Class>
<owl:ObjectProperty rdf:about="#hasTopping">
  <rdf:type
  rdf:resource="http://www.w3.org/2002/07/owl#InverseFunctionalProperty
  "/>
  <rdfs:comment xml:lang="en">
    Note that hasTopping is inverse functional because isToppingOf is
    functional
  </rdfs:comment>
  <rdfs:domain rdf:resource="#Pizza"/>
  <rdfs:subPropertyOf rdf:resource="#hasIngredient"/>
  <rdfs:range rdf:resource="#PizzaTopping"/>
  <owl:inverseOf rdf:resource="#isToppingOf"/>
</owl:ObjectProperty>
<owl:Class rdf:about="#PizzaBase">
  <rdfs:label xml:lang="pt">BaseDaPizza</rdfs:label>
  <rdfs:subClassOf rdf:resource="#Food"/>
  <owl:disjointWith rdf:resource="#PizzaTopping"/>
  <owl:disjointWith rdf:resource="#Pizza"/>
</owl:Class>
```

Simple Knowledge Organization System (SKOS)

Simple Knowledge Organization System (SKOS) is another early, foundational Semantic Web RDF vocabulary. Its purpose is to encode taxonomies, thesauri, subject heading lists,

classification schemes, and other forms of knowledge organization systems. The core of the SKOS vocabulary is the class "Concept," which is used for entries within the knowledge organization system being represented. Concepts can be organized into groups or schemes. They have labels, which can be represented as preferred or alternate labels, and notations, which are typically codes. SKOS allows for textual definitions, notes on the scope of a concept, and examples. It also provides properties for relationships between terms, such as broader, narrower, and related; and administrative information about the concept, such as editorial notes, a history of the entry for the term in the knowledge organization system, and change history. While most of SKOS is useful primarily in the narrow use case of encoding a formal knowledge organization scheme, several properties used for connecting related concepts to one another are widely used in other Linked Data applications. The SKOS properties broadMatch, closeMatch, and exactMatch, in particular, are commonly found.

SKOS Example: Extract from Entry "Symphonies" in Library of Congress Classification, N-Triple Syntax

```
<http://id.loc.gov/authorities/classification/M1001>
<http://www.w3.org/2004/02/skos/core#prefLabel> "Symphonies" .
<http://id.loc.gov/authorities/classification/M1001>
<http://www.w3.org/2000/01/rdf-schema#label> "Music and Books on Music--
--Music--Instrumental music--Orchestra--Original compositions--
Symphonies" .
<http://id.loc.gov/authorities/classification/M1001>
<http://www.w3.org/2004/02/skos/core#altLabel> "Symphonies" .
<http://id.loc.gov/authorities/classification/M1001>
<http://www.w3.org/2004/02/skos/core#broader>
<http://id.loc.gov/authorities/classification/M1001-M1049> .
<http://id.loc.gov/authorities/classification/M1001>
<http://www.w3.org/2004/02/skos/core#notation> "M1001" .
<http://id.loc.gov/authorities/classification/M1001>
<http://www.w3.org/2004/02/skos/core#inScheme>
<http://id.loc.gov/authorities/classification>.
```

Dublin Core (DC)

The Dublin Core Metadata Element Set (DCMES) grew out of a 1995 meeting in Dublin, Ohio, that was focused on metadata for networked electronic information. Attendees were tasked with identifying a core set of features common to most types of digital information. In this first meeting, 13 core elements were defined, which soon grew to the 15 elements known as DCMES today. These are: contributor, coverage, creator, date, description, format, identifier, language, publisher, relation, rights, source, subject, title, and type. This set, also known as "simple Dublin Core," or Dublin Core (DC), is standardized as ISO 15836 and ANSI/NISO Z39.85, both called *The Dublin Core metadata element set*.

The simplicity of DC was intentional, and led to relatively wide adoption early in its life. DC elements were soon embedded in Web pages and used heavily by early search engines for

indexing. DC was selected as the base, required metadata format for descriptions shared via the OAI-PMH protocol. Still, many metadata implementers asked for more specificity than DC offered, and in response, the Dublin Core Metadata Initiative (DCMI), which maintains the vocabulary, expanded simple DC with "qualifiers" to provide additional refinement to the core elements. This expanded version is now known as DCTERMS.

DC and DCTERMS are defined as both XML and RDF vocabularies. However, DCTERMS is more tightly defined as RDF and takes advantage of RDF features such as formally defined domains and ranges for classes. More than most metadata vocabularies, Dublin Core straddles the XML and RDF divide, attempting to be useful to both communities. The initiative is going in a decidedly RDF direction, however. Like the early DC XML elements before them, the DCTERMS RDF classes and properties are now commonly seen in real-world Linked Data, though the DCTERMS representation of the original 15 Dublin Core properties are by far the most widely implemented.

The DCMI has developed a number of other specifications under the Dublin Core umbrella that go beyond the definition of specific metadata vocabularies. The DCMI Abstract Model attempts to find a middle ground between the record-based, prescriptive XML approach and the graph-based, open RDF approach. It describes a model for resources slightly more prescriptive than the RDF graph, provides for bounded "description sets" that bundle individual descriptions of features of resources, and formally defines the role of controlled vocabularies and syntax-based rules for resource description. The DCMI Abstract Model adds to but references RDF. It is currently not widely referred to outside of the DCMI community, and it remains to be seen if the significant additional structure it provides over the RDF model proves valuable to metadata practitioners.

Along the same lines, the Singapore Framework for Dublin Core Application Profiles (Singapore Framework) are specifications that document the use of metadata for a specific purpose. They describe that purpose, document what types of resources are likely to be represented by descriptions matching the profile, list the metadata vocabulary terms allowed, and specify the encoding syntax for conformant metadata. With the Singapore Framework, like the DCMI Abstract Model, the DCMI community has introduced a model that is more complex than some other alternatives, and time will tell if this complexity provides utility.

DC Example: Extract from Record for an Image in the Portal to Texas History, Provided by the University of North Texas, XML Syntax

```
<?xml version="1.0" encoding="UTF-8"?>
<oai_dc:dc xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
      <dc:title>
            Maxine Walker Perini as a Child with Pet Dog and Doll in
```

```
Doll Buggy
      </dc:title>
      <dc:description>Copy negative of young Maxine Walker Perini
wearing a coat and bow, holding a Boston terrier, and pushing a doll in
buggy. She is on the sidewalk in front of the stairs of a
house.</dc:description>
      <dc:subject>People - Individuals</dc:subject>
      <dc:subject>Social Life and Customs - Pets - Dogs</dc:subject>
      <dc:subject>dolls</dc:subject>
      <dc:subject>toys</dc:subject>
      <dc:subject>Perini, Maxine Walker</dc:subject>
      <dc:subject>pets</dc:subject>
      <dc:subject>children</dc:subject>
      <dc:coverage>United States</dc:coverage>
      <dc:coverage>
            New South, Populism, Progressivism, and the Great
Depression, 1877-1939
      </dc:coverage>
      <dc:type>Photograph</dc:type>
      <dc:format>1 photograph : negative, b&w ; 4 x 5 in.</dc:format>
      <dc:format>Image</dc:format>
      <dc:identifier>local-cont-no: 81-00423-7</dc:identifier>
      <dc:identifier>
            http://texashistory.unt.edu/ark:/67531/metapth50844/
      </dc:identifier>
      <dc:identifier>ark: ark:/67531/metapth50844</dc:identifier>
</oai_dc:dc>
```

Friend of a Friend (FOAF)

Another early RDF vocabulary on the open Web is Friend of a Friend (FOAF). This vocabulary provides for descriptive metadata on people and organizations, along with their attributes and relationships. FOAF classes include Person, Organization, Group, and Project, which make use of properties such as name, title, and member. The FOAF Core defines a small number of primary classes and properties designed to be a baseline for further refinement and specification. FOAF also includes a list of classes and properties related to how people and organizations interact with the social Web, including terms such as PersonalProfileDocument and accountName.

FOAF is primarily used in systems to identify people and organizations and provide basic information about them. The vocabulary does support some more advanced features, however, such as denoting the individuals who are members of a group or documenting personal interests.

FOAF Example: Extract from Entry for Nobel Laureate in Medicine Peter C. Doherty in ORCID, Turtle Syntax

```
<http://orcid.org/0000-0002-5028-3489>
      a       foaf:Person , prov:Person ;
      rdfs:label "Peter Charles Doherty" ;
      foaf:account <http://orcid.org/0000-0002-5028-3489/> ;
      foaf:based_near
              [ a       gn:Feature ;
                gn:countryCode "AU" ;
                gn:parentCountry <http://sws.geonames.org/2077456/>
              ] ;
      foaf:familyName "Doherty" ;
      foaf:givenName "Peter Charles" ;
      foaf:publications <http://orcid.org/0000-0002-5028-3489/> .
```

ONline Information eXchange (ONIX)

An advanced metadata format in the e-commerce sector is the publishing industry's ONline Information eXchange (ONIX) XML schema. ONIX consists of three formats—ONIX for Books, ONIX for Serials, and ONIX for Publications Licenses—though the format for books is by far the most widely used. It is maintained by EDItEUR, an international standards group in the print and digital book and serials sector, together with book industry organizations in the United States and the United Kingdom. ONIX 3.0, released in 2009, provides enhanced support for ebooks. Using the ONIX for Books XML document format, publishers can provide detailed metadata about their products to retailers. Feeds can include identifiers such as ISBN and barcodes, measurements, robust encoding of multiple forms of a title, multiple forms of names for authors and other contributors and their affiliations, marketing materials such as book jacket text, and supply details.

EXchangeable Image File Format (Exif)

The EXchangeable Image File Format (Exif) is somewhat of a misnomer, as it is not a file format, but rather a tag structure for embedded metadata within digital image files. It emerged from the Japanese digital camera industry and is currently supported by nearly all digital camera and smartphone manufacturers. Various software packages for editing and sharing images, including Photoshop and Flickr, support Exif,. However, many social media sites, such as Instagram and Facebook, strip Exif metadata from shared images, though they often read and store the geolocation information from the Exif metadata before doing so. The TIFF and JPEG file formats support embedded Exif, but JPEG2000, PNG, and GIF do not. Exif stores mostly technical metadata about the image, but there is also a supplementary section for metadata about embedded audio. The Exif specification includes metadata elements such as pixel dimensions, date and time taken, ISO setting, aperture, white balance, and information on the lens used.

Notable Metadata Languages: Examples from the Cultural Heritage Sector

MAchine Readable Cataloging (MARC)

The most widely used metadata language in the library community far predates XML and RDF technologies and indeed modern metadata formats in general. MAchine Readable Cataloging (MARC) first arose in 1968 out of a pilot project at the Library of Congress to experiment with distributing the information on catalog cards to libraries in machine readable form. Since then, it has become entrenched as the metadata format underlying online library catalogs and as the way libraries share records with each other. OCLC's WorldCat database, used for sharing records among libraries and allowing users to find holdings across multiple institutions, holds nearly 380 million MARC bibliographic records as of July 2016.[6]

MARC is standardized as ANSI/NISO Z39.2 *Information Interchange Format* and ISO 2709 *Information and documentation—Format for information exchange*. The ISO 2709 format is designed for maximum information-storage efficiency. The bulk of the record is made up of variable fields that can be any length. A directory at the beginning of the record points processors to where each field begins, allowing no empty bits to be stored. Field names are alphanumeric characters rather than language-based labels, and are known as tags. An ISO 2709 MARC field can have two single-character indicators that provide meta-information about the values in the field. Fields are further broken down into alphanumeric subfields that are used to encode various aspects of the information in the field.

MARC is actually a family of formats, with different implementations of ISO 2709 used in different countries. Most notable are the MARC21 formats maintained by the Library of Congress, which are in use in the United States, Canada, and much of the English-speaking world. MARC21 is composed of five formats: MARC21 Bibliographic, MARC21 Authority, MARC21 Holdings, MARC21 Classification, and MARC21 Community Information. The first two are in wide use, the third appears only in select systems, and the final two have even more

[6] https://www.oclc.org/worldcat/inside-worldcat.en.html

limited implementation. MARC21 uses only numeric field codes, making it slightly more restrictive than the parent ISO 2709 format.

The MARC21 bibliographic format is used for describing the items that libraries hold. It is made up of several hundred fields, though a much smaller core set is used most frequently. These include fields for various types of titles, authorship of works by people or groups, edition and publication information, physical description, series, notes, and subject and genre terms. The MARC21 Authority format is used for documenting controlled terms for people, corporate bodies, work titles, subjects, and genres. These controlled terms are then used as entries in the appropriate fields in MARC21 bibliographic records, to provide consistency in these records and aid discovery. The MARC Authority format includes fields for encoding the controlled heading, which often include additional metadata about the entity—occupation for a person, for example; address for a person or corporate body; or key for a musical work. It further includes fields for encoding alternate forms of a name for the entity and making notes that document why the particular form of the controlled heading was chosen.

MARC Example: Record for Nigella Lawson's *How to Eat* in British Library Online Catalog

```
FMT   BK
LDR         am a2200217ua 4500
001   011981326
008   981130s1998    enka   ||     001 ||eng
015   |a GB98Z0319 |2 bnb
020   |a 0701165766 : |c £25.00
040   |a StDuBDS |d Uk
08204 |a 641.5 |2 21
1001  |a Lawson, Nigella, |d 1960-
24510 |a How to eat : |b the pleasures and principles of good food / |c
      Nigella Lawson.
260   |a London : |b Chatto & Windus, |c 1998.
300   |a xviii,526p. : |b ill. (some col.) ; |c 24cm.
336   |a text |2 rdacontent
337   |a unmediated |2 rdamedia
338   |a volume |2 rdacarrier
500   |a Includes index.
650 0 |a Cooking.
85241 |a British Library |b HMNTS |j YK.1998.b.9105
SYS   011981326
```

Bibliographic Framework Initiative (BIBFRAME)

The Bibliographic Framework Initiative (BIBFRAME) is a project based at the Library of Congress that aims to design a new model for encoding and sharing of bibliographic information. It is structured according to Linked Data principles, to allow library data to operate more

effectively in 21st-century information environments and become part of the emerging "Web of Data." BIBFRAME is a formal RDF vocabulary. It is intended to eventually replace MARC21, and to keep many of MARC21's semantics, allowing significant proportions of existing data to be migrated forward. The large and comprehensive BIBFRAME model takes the approach of defining classes and properties in its own namespace for all features deemed important to its scope. It does, however, define a few of its highest-level classes as subclasses of well-known entities defined by other communities. BIBFRAME 2.0 was released in April 2016, but many design issues are still under discussion, and the model therefore should not yet be considered fully stable.

The BIBFRAME 2.0 model defines entities for Work (the conceptual essence of a resource), Instance (an individual, material embodiment of a Work), Item (an actual physical or electronic copy of an Instance), Agent (person or organization associated with a Work), and Event (an occurrence that is recorded in a work). Each of these is modeled as an RDF class, with subclasses defined for more specific concepts within these categories.

Additional classes and properties are defined in BIBFRAME for the types of bibliographic and authority data traditionally recorded by libraries and encoded in MARC. Metadata about titles, creators, topics, genres, form, language, production, edition, physical characteristics, identifiers, notes, relationships to other resources, and holdings in a specific library are all provided for in BIBFRAME.

BIBFRAME Example: Extract from Entry for Suzanne Pickett's *Hot Dogs for Thanksgiving* on BIBFRAME Website, RDF/XML Syntax

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:bf="http://bibframe.org/vocab/">
    <bf:Work
        rdf:about="http://bibframe.org/resources/sample-lc-2/731515">
        <rdf:type rdf:resource="http://bibframe.org/vocab/Text"/>
        <bf:authorizedAccessPoint>Pickett, Suzanne. Hot dogs for
Thanksgiving / by Suzanne Pickett.Hot dogs for
Thanksgiving</bf:authorizedAccessPoint>
        <bf:workTitle
rdf:resource="http://bibframe.org/resources/sample-lc-
2/731515title29"/>
        <bf:creator rdf:resource="http://bibframe.org/resources/sample-
lc-2/731515person30"/>
        <bf:language
rdf:resource="http://id.loc.gov/vocabulary/languages/eng"/>
        <bf:subject rdf:resource="http://bibframe.org/resources/sample-
lc-2/731515person32"/>
        <bf:subject rdf:resource="http://bibframe.org/resources/sample-
lc-2/731515topic33"/>
        <bf:subject rdf:resource="http://bibframe.org/resources/sample-
lc-2/731515topic34"/>
        <bf:subject
rdf:resource="http://id.loc.gov/vocabulary/geographicAreas/n-us-al"/>
```

```
        <bf:classificationLcc
rdf:resource="http://id.loc.gov/authorities/classification/CT275.P628"/
>
        <bf:authorizedAccessPoint xml:lang="x-bf-
hash">pickettsuzannehotdogsforthanksgivingengworktext</bf:authorizedAcc
essPoint>
    </bf:Work>
    <bf:Instance>
        <rdf:type rdf:resource="http://bibframe.org/vocab/Monograph"/>
        <bf:instanceTitle
rdf:resource="http://bibframe.org/resources/sample-lc-
2/731515title44"/>
        <bf:isbn13
rdf:resource="http://isbn.example.org/9781881320760"/>
        <bf:publication>
            <bf:Provider>
                <bf:providerName>
                    <bf:Organization>
                        <bf:label>Black Belt Press</bf:label>
                    </bf:Organization>
                </bf:providerName>
                <bf:providerPlace>
                    <bf:Place>
                        <bf:label>Montgomery, AL </bf:label>
                    </bf:Place>
                </bf:providerPlace>
                <bf:copyrightDate>c1998.</bf:copyrightDate>
            </bf:Provider>
        </bf:publication>
        <bf:extent>190 p. ;</bf:extent>
        <bf:dimensions>23 cm.</bf:dimensions>
        <bf:instanceOf
rdf:resource="http://bibframe.org/resources/sample-lc-2/731515"/>
    </bf:Instance>
        <bf:Annotation
        rdf:about="http://bibframe.org/resources/sample-lc-
2/731515annotation40">
        <bf:derivedFrom
rdf:resource="http://bibframe.org/resources/sample-lc-
2/731515.marcxml.xml"/>
        <bf:descriptionSource
rdf:resource="http://id.loc.gov/vocabulary/organizations/dlc"/>
        <bf:descriptionConventions
rdf:resource="http://id.loc.gov/vocabulary/descriptionConventions/aacr2
"/>
        <bf:generationProcess>DLC transform-tool:2015-07-23-
T17:01:00</bf:generationProcess>
        <bf:changeDate>1998-07-21T12:52</bf:changeDate>
        <bf:annotates
rdf:resource="http://bibframe.org/resources/sample-lc-2/731515"/>
    </bf:Annotation>
</rdf:RDF>
```

Metadata Object Description Schema (MODS)

The Library of Congress's Metadata Object Description Schema (MODS) is an XML Schema for bibliographic information of particular interest to libraries. Its elements and attributes are English-language tags, but they are heavily inspired by the tags in the MARC format. Indeed, MODS was conceived as a schema that could represent most of the semantics of MARC but be friendly to XML-based applications and environments.

The MODS XML Schema is organized into 20 top-level elements that group related parts of the bibliographic description. These are: titleInfo, name, typeOfResource, genre, originInfo, language, physicalDescription, abstract, tableOfContents, targetAudience, note, subject, classification, relatedItem, identifier, location, accessCondition, part, extension, and recordInfo. The extension element is a distinctive feature of MODS, which allows additional metadata in any XML namespace to be embedded in a MODS record. This facilitates some measure of extensibility—where additional elements are needed, they can be included within the same description. If a system that encounters these MODS records understands the namespace for elements used within the extension element, it can index and process that data.

MODS Example: Extract from Record for a Scholarly Article in Columbia University's Academic Commons Repository

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mods xmlns="http://www.loc.gov/mods/v3"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.loc.gov/mods/v3
http://www.loc.gov/standards/mods/v3/mods-3-4.xsd">
   <titleInfo>
      <title>You Can&#8217;t Think and Hit at the Same Time: Neural
Correlates of Baseball Pitch Classification</title>
   </titleInfo>
   <name type="personal" ID="jss2212">
      <namePart type="family">Sherwin</namePart>
      <namePart type="given">Jason Samuel</namePart>
      <role>
         <roleTerm type="text">author</roleTerm>
      </role>
      <affiliation>Columbia University. Biomedical
Engineering</affiliation>
   </name>
   <name type="personal" ID="jsm2112">
      <namePart type="family">Muraskin</namePart>
      <namePart type="given">Jordan Scott</namePart>
      <role>
         <roleTerm type="text">author</roleTerm>
      </role>
      <affiliation>Columbia University. Biomedical
Engineering</affiliation>
   </name>
   <name type="personal" ID="ps629">
      <namePart type="family">Sajda</namePart>
```

```
      <namePart type="given">Paul</namePart>
      <role>
         <roleTerm type="text">author</roleTerm>
      </role>
      <affiliation>Columbia University. Biomedical
Engineering</affiliation>
   </name>
   <name type="corporate">
      <namePart>Columbia University. Biomedical Engineering</namePart>
      <role>
         <roleTerm type="text">originator</roleTerm>
      </role>
   </name>
   <extension
xmlns:rioxxterms="http://docs.rioxx.net/schema/v1.0/rioxxterms/"
xmlns="http://www.loc.gov/mods/v3"
xsi:schemaLocation="http://docs.rioxx.net/schema/v1.0/rioxxterms/
http://docs.rioxx.net/schema/v1.0/rioxxterms.xsd">
      <rioxxterms:funder>National Institutes of Health
(U.S.)</rioxxterms:funder>
      <rioxxterms:projectid>R01-MH085092</rioxxterms:projectid>
   </extension>
   <typeOfResource>text</typeOfResource>
   <genre>Articles</genre>
   <originInfo>
      <dateIssued encoding="w3cdtf" keyDate="yes">2012</dateIssued>
   </originInfo>
   <language>
      <languageTerm type="text">English</languageTerm>
   </language>
   <abstract>Hitting a baseball is often described as the most
difficult thing to do in sports. A key aptitude of a good hitter is the
ability to determine which pitch is coming. This rapid decision
requires the batter to make a judgment in a fraction of a second based
largely on the trajectory and spin of the ball. When does this decision
occur relative to the ball&amp;#8217;s trajectory and is it possible to
identify neural correlates that represent how the decision evolves over
a split second? Using single-trial analysis of electroencephalography
(EEG) we address this question within the context of subjects
discriminating three types of pitches (fastball, curveball, slider)
based on pitch trajectories...</abstract>
   <subject>
      <topic>Biomedical engineering</topic>
   </subject>
   <subject>
      <topic>Neurosciences</topic>
   </subject>
   <subject>
      <topic>Bioinformatics</topic>
   </subject>
   <relatedItem type="host">
      <titleInfo>
         <title>Frontiers in Neuroscience</title>
      </titleInfo>
      <part>
         <detail type="volume">
            <number>6</number>
```

```
         </detail>
         <detail type="issue">
            <number>177</number>
         </detail>
         <extent unit="page">
            </extent>
         <date>2012-12-19</date>
      </part>
      <identifier
type="doi">http://dx.doi.org/10.3389/fnins.2012.00177</identifier>
      <identifier type="issn">1662-453X</identifier>
   </relatedItem>
   <identifier type="CDRS
doi">http://dx.doi.org/10.7916/D8GQ6VVF</identifier>
   <location>
      <physicalLocation authority="marcorg">NNC</physicalLocation>
   </location>
</mods>
```

CIDOC Conceptual Reference Model (CIDOC CRM)

The museum community also has a strong history of developing and using formal metadata standards. The International Council on Museums's International Committee for Documentation, known as CIDOC, has developed a Conceptual Reference Model (CRM) that serves as a base ontology for concepts needed in museum documentation and cultural heritage metadata. The model is standardized as ISO 21127 *Information and documentation – A reference ontology for the interchange of cultural heritage information*.

The CIDOC CRM is defined as a textual but formal reference model, with an RDFS representation as a derivative of the formal model. As such, it is primarily intended as a core set of terms for the discussion of concepts among multiple communities and among different metadata formats. CIDOC CRM-compatible systems might encode data in any number of RDF vocabularies or XML schemas.

A distinctive feature of the CIDOC CRM is its focus not only on describing cultural heritage objects, but also on the acts and events related to their creation and lifespan. The entities defined by the ontology are therefore wide-ranging, including classes for features such as places, time spans, events, actors (people who take some relevant action), physical things, and information objects.

Categories for the Description of Works of Art (CDWA)

The Categories for the Description of Works of Art describes itself as a conceptual framework for information about artworks, writ large. It is intended as a basis for designing art information systems, and is used primarily by the art museum community. CDWA defines approximately 540 data elements and relationships among them; a smaller number are defined as the core minimum

required for the meaningful description of a work. As a conceptual framework, CDWA prescribes no specific information encoding.

The design of CDWA focuses on the creation of high-quality and actionable information. It promotes the use of separate display and indexing data for many categories, to allow more effective searching while at the same time showing coherent information to users. It also promotes the use of authoritative sources for the details recorded in a CDWA description, at times even suggesting specific sources to use in compiling information in a given category.

CDWA defines categories for basic facts such as title and artist; the artistic content of the work, such as its style and period, materials and techniques, and subject matter; physical attributes such as inscriptions and measurements; curatorial details such as current location, collecting history, and exhibition and loan history; and relationships to other artworks and textual references.

Visual Resources Association Core (VRA Core)

The visual resources community supports disciplines, such as art history, that rely upon discovering and using reproductions of works of art and architecture. A professional organization for that community, the Visual Resources Association (VRA), has developed the VRA Core metadata vocabulary for recording information about these works of art and specific representations of them. The VRA Core is represented as two different XML Schemas, a restricted version that strictly enforces the use of certain pre-specified values in the "type" attribute on many VRA Core XML elements, and an unrestricted version that allows free text to be used for these values. While the VRA Core is maintained by the visual resources community, the official XML Schemas and documentation are hosted at the Library of Congress.

A distinctive feature of the VRA Core is its separation of metadata about the work of art itself and metadata about images of those works. The same VRA Core elements can be used to describe either a work or an image, but the values for these elements will likely be different; for example, they will typically have different creation dates or creators. Works and images in VRA Core can also be grouped into collections. Like CDWA, VRA Core allows for recording values for indexing in addition to values intended for display. As a metadata schema focused on works of art, VRA Core contains elements for material, technique, inscription, measurements, cultural context, style/period, and work type, in addition to more generic elements for cultural material, such as agent (the creator of a work), title, date, and rights.

Encoded Archival Description (EAD)

Encoded Archival Description (EAD) is the primary metadata vocabulary in use in the English-speaking archival community. Like the VRA Core, it is developed and maintained by domain experts, in this case the Society of American Archivists, and the Library of Congress hosts EAD's public documentation. As archives tend to deal with collections of material (often called fonds), they do not typically describe individual physical or digital items. Instead, they practice multi-level description, meaning that a collection of resources with a shared origin is described as

a whole, then (optionally) smaller subsets are further described in more detail. These hierarchical descriptions are presented along with historical information about the collection, its creator, and its context in a document known as a finding aid.

EAD is an XML markup language for finding aids rather than a true descriptive metadata standard. However, its tags are still largely semantic in nature rather than format-focused. Most EAD elements are for archival concepts such as a biography of the creator of a set of records; scope and content notes for the material being described; archival components such as series, subseries, or file; or storage containers such as boxes and folders. Each of these elements can be used at any level in the multi-level description. EAD also provides elements to denote textual features such as lists or paragraphs, and, when desired, mark up important words or phrases that are notable, including names or dates.

Notable Metadata Languages: Other Examples

Data Documentation Initiative (DDI)

The Data Documentation Initiative (DDI) metadata standard is a large element set designed for describing data in the social, behavioral, and economic sciences. It is structured with features that are useful at any phase of the research lifecycle, including study conception, data collection, data normalization and analysis, sharing, and archiving. DDI has become a key standard in its target disciplines, and is growing even more important as the research community increasingly values data management planning and the open dissemination and long-term stewardship of research data.

DDI is canonically defined as a number of XML schemas, separated into modules. DDI includes metadata about research studies, data collection methods, questions and responses, variables, links to raw or processed data sets, and the relationships among studies and data sets. The XML version of DDI is implemented in many software packages that deal with social science data. An RDF representation of DDI for sharing data sets and metadata about them as Linked Data is in development.

The DDI Alliance, which manages the element set, has also defined several controlled vocabularies for use with DDI metadata. These vocabularies cover areas of particular interest to the social sciences, such as unit of analysis, data type, and mode of collection.

PREservation Metadata: Implementation Strategies (PREMIS)

The major preservation metadata standard in the digital archiving realm is PREservation Metadata: Implementation Strategies (PREMIS). It is intended to describe properties of digital content necessary to support the digital preservation process, track preservation actions taken, and record information about the actors responsible. PREMIS is maintained at the Library of Congress by a volunteer editorial board, which operates on public input. It is defined as an XML schema, but an RDF definition in OWL has been released as well.

PREMIS defines five entities that are described or acted upon in digital preservation systems: Objects, Environments, Events, Agents, and Rights. Objects are information units upon which preservation actions are taken, and include intellectual entities, representations, files, and bitstreams, all of which have defined relationships to one another. Environments are hardware or software in which these types of content reside. Actions notable in a digital preservation environment are known as Events. Agents are people, organizations, or software, and, finally, Rights are statements that assert some type of permission over content. Each of these entities takes a variety of properties relevant to their purpose; for example, Object properties include objectIdentifier, preservationLevel, fixity, size, and format; and Event properties include eventIdentifier, eventType, and eventDateTime.

Text Encoding Initiative (TEI)

The Text Encoding Initiative (TEI) is a markup language for machine-readable texts of all types, including prose, verse, texts, transcripts of spoken-word performances, dictionaries, and manuscripts and other primary sources. It is an extraordinarily large markup language, and as such is not designed to be used as a whole in any given implementation. Instead, its source XML definitions can be rendered as XML DTDs, XML schemas, or RELAX NG schemas, by picking which modules (groups of related elements intended for a specific purpose) should be included for a specific project. Implementers can go even farther by excluding or renaming specific elements in a module, or adding elements or attributes of their own. These implementation-level customizations are recorded in a documentation format called "One Document Does it All" (ODD), which itself is a variety of the TEI.

TEI modules cover a wide range of textual elements. Basic structural features are commonly used, and include paragraphs, headings, verses, lines, named speakers, stage directions, and quotations. Semantic elements such as names, numbers, and dates can be similarly marked up. Extra- or non-textual features such as tables and embedded graphics may be included, as well as metadata about the text. TEI's multi-lingual support is strong, allowing signaling of the language used in any part of a text, supplying alternate language versions of passages of a document, allowing multiple character sets (through Unicode support), and defining document-specific glyphs. In addition to supporting the marking up of textual features, TEI provides for electronic scholarly editions of texts through its "Critical Apparatus" module. The elements and attributes in this module support encoding variations between multiple sources of a text, and variant readings of a (typically manuscript) text.

Music Encoding Initiative (MEI)

The Music Encoding Initiative (MEI) is an encoding format for musical notation closely based on the TEI. It is an XML language, and borrows from the TEI community its organization into modules, methods for customization for specific projects, and use of ODD.

MEI supports several of the most commonly used forms of musical notation, including common Western music notation (the form with which most readers will be familiar), mensural and

neumatic notation, and tablature for guitar and lute. In addition to a header for metadata about the musical score being described, MEI has features for all symbols needed in each of the supported notation formats, including elements for scores, parts, staffs, key signatures, clefs, measures, bar lines, notes, and chords. Like TEI, MEI also supports the encoding of analytical and editorial structures.

How is Metadata Generated?

In the metadata traditions for the cultural heritage sector, where descriptive metadata has primarily been created to assist users with finding printed books, journals, manuscripts, or cultural objects of interest, there were historically few choices other than to have humans create that metadata. Features such as title, author, and publication date were generally transcribed by hand after examining a physical item. For other pieces of metadata, such as background information about an author or performance or publication history, an expert would do targeted research and then record the results. Value-added and interpretive information, such as summaries or subjects, were similarly developed and supplied by experts.

At first, technology simply allowed institutions to share this hand-crafted metadata to reduce duplication of effort. Purpose-built metadata entry systems then started to emerge, and later, widely accessible tools such as spreadsheets began to be used to speed the metadata creation process. More recently, metadata creation interfaces have become increasingly sophisticated, with user-friendly designs that don't necessarily present data entry screens that replicate the underlying data structures. Today, metadata is typically created through intermediate steps rather than using XML or RDF directly. One exception to this principle is marking up content with metadata, for example, encoding a text in TEI.

Technological advances have also demonstrated reliable methods for creating metadata through automated processes, especially as born-digital information has become the norm. Technical metadata in particular is an example of this: most file formats include at least some embedded technical information designed to assist software with interpreting the content. There is also a history of software using system-level information to add extra administrative information to digital files, such as date created or the ID of the user logged into the system at the time the file was generated.

The Web and greater integration of software systems have also made it easier to effectively share metadata. With increasing transmission of digital content between organizations and customers, interoperability of metadata is essential. Everyone benefits when metadata can be transferred effectively; it reduces duplication of effort. Amazon's collection of metadata about products, for example titles and authors for books that come through the ONIX supply chain, is a high-profile example of the benefits of integration that leverages existing technologies and reduces the need for downstream metadata creation.

Lately, processes have emerged to analyze digital content and automatically generate metadata about it. Automated transcription of speech from audio and video is a relatively mature technology, especially for recordings captured in controlled environments with dedicated sound systems. Facial recognition technology for video and still images is improving quickly. For textual resources, latent semantic analysis and topic modeling allow for semi-supervised generation of topics relevant to the analyzed texts. Part-of-speech and named-entity recognition technologies are frequently used in research environments. Automated image annotation, using algorithms to identify objects in photographs, is a burgeoning research area. A similar research community, called music information retrieval, focuses on signal processing for audio files. Work from this community is integral to playlist generation in online music streaming services, but it also endeavors to perform tasks such as automated genre classification for musical recordings. Real progress and possibilities for high-quality data generated through programmatic means are emerging. These and other successes in automated metadata generation have started to bridge communities, allowed them to share their data more effectively, and are beginning to make each more open to metadata creation methods used by others.

Future Directions

The emergence of the Linked Data movement has had a profound impact on the way metadata is used in the commercial, information, and cultural heritage sectors. The technology driving Linked Data systems has made it easier to share information, and this ease has fueled buy-in for the movement's underlying focus on data openness and interconnectedness.

Linked Data thinking has influenced how metadata and the systems that use it are viewed as well. The focus of Linked Data on the graph as a whole rather than on bounded sets of data defined by their source has led many to value connections among data sets and become less invested in maintaining distinctions among them. The Semantic Web's open-world assumption posits that a lack of information must lead to an uncertain conclusion (a closed-world assumption, on the other hand, offers more certainty but relies on all relevant information already being known). This stance has led to a desire to more heavily link data from multiple sources, building value and new knowledge from these connections. Metadata creation in this context becomes less about filling in fields on data-entry forms than about making links between pre-existing things. The future may very well continue this trend towards deeper connections and larger knowledge graphs.

The vast amount of information being generated, processed, and transferred in the 21st century, along with technological advances, has led to an increasing reliance on automated means of creating and maintaining metadata. Systems are simply becoming more intelligent in the way they generate and process information. Tasks once done by humans, such as looking up the form of name an author prefers or normalizing a textual date to a standardized form, can now be assigned to the system, saving valuable human effort for cases in which it is truly necessary. Software tools will likely be increasingly expected to display more information to users, and let them decide what is valuable and useful, rather than predetermine what a person should see. To handle this significant increase in the volume of data a user might need to interact with, we will likely see better-designed user interfaces that clearly show where conflicting information exists and are transparent as to the sources of the information being displayed.

Finally, the emerging culture of openness and interconnectedness of metadata is leading to a redefined definition of "authoritative" or "good" metadata. The Web has opened up new opportunities for previously marginalized voices to share their knowledge. Informed enthusiast communities exist online for nearly every topic, and these individuals can frequently provide far better metadata than organizations that are tasked with managing content but that lack this subject expertise. This process is sometimes known as "crowdsourcing" or "trusting the wisdom of the crowd." Intelligent systems can combine this user-generated metadata with metadata from more traditional sources in a way that is sensible to users. To facilitate this deeper integration of metadata, we can likely expect interfaces of systems designed for laypeople to become incredibly streamlined and user friendly, building on activities this community is already motivated to do.

It's also likely that more advanced rewards structures will emerge to compensate users for their work.

We can likely expect a lengthy transition period to this more open and linked model. There are real and valid reasons for certain systems to rely on more controlled, traditionally authoritative approaches to metadata, or for an organization to desire more human oversight over the management of this critical resource. These will almost certainly always be a part of the overall metadata landscape. Yet the benefits of sharing metadata and adopting related vocabularies from others are compelling. It is enticing to imagine that this culture of openness will lead to different metadata communities learning from each other and working more closely together in the future.

Appendix: Resources

Metadata Standards and Vocabularies

More metadata standards, with links to specifications and analysis of their utility on several axes, can be found at http://jennriley.com/metadatamap/

BIBFRAME http://www.loc.gov/bibframe/

CDWA http://www.getty.edu/research/publications/electronic_publications/cdwa/

CIDOC CRM http://www.cidoc-crm.org/

DCMI Abstract Model http://dublincore.org/documents/abstract-model/

DDI http://www.ddialliance.org/

Dublin Core http://dublincore.org/

EAD http://www.loc.gov/ead/

Exif http://www.cipa.jp/std/documents/e/DC-008-2012_E.pdf

FOAF http://xmlns.com/foaf/spec/

MARC http://www.loc.gov/marc/

MEI http://music-encoding.org/

MODS http://www.loc.gov/standards/mods/

ONIX for Books http://www.editeur.org/93/Release-3.0-Downloads/

OWL http://www.w3.org/TR/owl-features/; http://www.w3.org/TR/2012/REC-owl2-primer-20121211/

PREMIS http://www.loc.gov/standards/premis/

Schema.org http://schema.org/

Singapore Framework for Dublin Core Application Profiles
http://dublincore.org/documents/2008/01/14/singapore-framework/

SKOS http://www.w3.org/2004/02/skos/

TEI http://www.tei-c.org/

VRA Core http://www.loc.gov/standards/vracore/

XML Standards and Resources

OAI-PMH https://www.openarchives.org/pmh/

ResourceSync http://www.openarchives.org/rs/1.0/resourcesync

W3C XML Activity http://www.w3.org/XML/

XPath http://www.w3.org/TR/xpath/

XSLT http://www.w3.org/TR/xslt20/

XQuery http://www.w3.org/TR/xquery/

RDF Standards and Resources

RDF Concepts http://www.w3.org/TR/rdf11-concepts/

RDF Primer http://www.w3.org/TR/rdf11-primer/

RDFS http://www.w3.org/TR/rdf-schema/

SPARQL http://www.w3.org/TR/sparql11-query/

W3C RDF Activity http://www.w3.org/RDF/

RDF Serializations

JSON-LD http://www.w3.org/TR/json-ld/

Microformats http://microformats.org/

N-Triples http://www.w3.org/TR/n-triples/

RDF/XML http://www.w3.org/TR/rdf-syntax-grammar/

RDFa http://www.w3.org/TR/xhtml-rdfa-primer/

Turtle http://www.w3.org/TR/turtle/

Linked Data Resources

Datahub (data registry including many Linked Data sources) http://datahub.io/

Linked Data Cloud (visualization of sites sharing Linked Data) http://lod-cloud.net/

Linked Data—Design Issues (Tim Berners-Lee).
http://www.w3.org/DesignIssues/LinkedData.html

Linked Data: Evolving the Web into a Global Data Space (handbook of best practices for Linked Data implementations) http://linkeddatabook.com/book

LOD Stats (site for discovery of nearly 10,000 Linked Data datasets) http://stats.lod2.eu/

OCLC Linked Data Strategy http://www.oclc.org/data.en.html

OCLC WorldCat Linked Data Vocabulary https://www.oclc.org/developer/develop/LinkedData/worldcat-vocabulary.en.html

W3C Data Activity http://www.w3.org/2013/data/

Projects to Watch

BIBFRAME http://www.loc.gov/bibframe/
An emerging Linked Data-friendly vocabulary focusing on bibliographic and authority information for use by the library community. Developed and managed by the Library of Congress.

DBpedia http://www.dbpedia.org
A community-based project that extracts structured data from the unstructured text in Wikipedia articles, and exposes that data for other uses.

Digital Public Library of America http://dp.la
A portal to discover content from American libraries, archives, and museums, and a platform that

makes data about this content available for open reuse.

Europeana http://www.europeana.eu
A portal to discover content from European libraries, archives, and museums, and a platform that makes data about this content available for open reuse.

LC Linked Data Service http://id.loc.gov
A service from the Library of Congress providing Linked Data access to many controlled vocabularies maintained at the Library. The service is available for human users through a Web user interface, and to software applications through content negotiation.

Linked Data for Libraries (LD4L), LD4L Labs, and Linked Data for Production (LD4P) http://www.ld4l.org/
A suite of collaborative projects of the Cornell University Library, the Harvard Library Innovation Lab, Stanford University Libraries, and others to create practical Linked Data implementations integrating data from multiple library sources, using pre-existing ontologies and software, and as evolutions of library technical services workflows.

RDF Data Shapes Working Group https://www.w3.org/2014/data-shapes/charter
A W3C working group to define methods for defining structural constraints and validation methods for RDF graphs.

Schema Bib Extend Community Group (SchemaBibEx) https://www.w3.org/community/schemabibex/
A community of the World Wide Web Consortium seeking to extend schema.org with bibliographic information.

Wikidata http://www.wikidata.org
A project from the Wikimedia foundation to record structured data about things, concepts, and events, Wikidata is open to contributions by anyone. It provides raw data to other Wikimedia projects such as Wikipedia, and makes it available for use in external applications.