Chapter 3

Metadata Sources

Metadata is generally known as 'data about data.' In the context of a library, the catalogue records, whether physical cards or electronic form, are a form of metadata. However, in today's world the term 'metadata' is invariably linked to the electronic form of data that is used for describing any content. [23]

According to NISO [16]:

*Metadata is structured information that describes, explains, locates, or otherwise makes it easier to retrieve, use, or manage an information resource. Metadata is often called data about data or information about information.*

In the last decade, we have seen the emergence of some new information resources, which are mostly the product of information explosion on the web. The major characteristics of these resources is that they are web-friendly. Many of these resources are useful for scholarly purposes, even though they are sometimes not structured enough for systematic indexing. Some of these sources do follow some structured formats but they are suited for specific types of applications, such as OAI-PMH Data Provider and RSS/Atom feeds. These resources can be additional sources of metadata for scholarly resources nonetheless.
3.1 Purpose of Metadata

Metadata facilitates access to the document it describes. In case of text documents, metadata may contain information about physical extent of the document, name of the author, date of creation, and a short summary or abstract. In the case of a digital image, the metadata may include size of the image, the colour depth, the resolution, date of creation, etc.

3.1.1 Types of Metadata

Metadata is broadly defined into three types [16]:

**Descriptive metadata** describes a resource for purposes such as discovery and identification. It includes elements such as author, title, abstract, and keywords.

**Structural metadata** describes the structure of documents. For example, how the components of a digital object is organized.

**Administrative metadata** provides technical information about documents, such as date of creation, file type, and location of the document as well as access control information. There may be several types of administrative data; two major types are:

- Rights management metadata
- Archival and preservation metadata

This study restricted to descriptive metadata, which is used for the discovery of relevant content in a collection. Some of the functions of descriptive metadata are:
3.1.2 Resource Discovery

Primary role of metadata is resource discovery, which is achieved by [16]:

- allowing resources to be found by relevant criteria (author, title, etc.);
- identifying resources by providing unique identifier to each document, which helps de-duplication;
- bringing similar resources together for certain fields (faceting);
- distinguishing discrete resources by using some taxonomy; and
- giving location information of the content.

3.1.3 Interoperability

Interoperability is the metadata exchange framework which allows heterogeneous systems, data structures, and interfaces to work together with minimal loss of content and functionality. Achieving interoperability mandates sharing certain principles between participating systems such as use of a well-defined metadata schema, use of standard search protocol, and crosswalks between metadata schemes.

There are two major approaches to interoperability, which are:

**Federated search:** Using a common search protocol (such as Z39.50), the same query is broadcast to multiple collections to get a unified search result.

**Metadata harvesting:** This is an alternative approach that promotes exposing the metadata of a collection in such a way that the search service provider can harvest the metadata to index in the local collection to provide a unified search facility to its users.
3.1.4 Digital Identification

Most metadata schemes include some elements to uniquely identify the work or object it describes. Some of the examples include Control Number (used in MARC21), Uniform Resource Locator (URL) and Digital Object Identifier (DOI). Persistent universal identifiers (e.g., DOI) are better than object locations (e.g., URL) as unique identifier since the location is prone to change, rendering the metadata record invalid.

3.1.5 Archiving and Preservation

The data formats of digital objects change over time due to various reasons. Hence, the metadata which describe its archival and preservation elements (such as lineage of a digital object and its physical attributes) is a key to ensure access to the data in future collections. Thus metadata also helps in archival and preservation of the content it describes.

3.2 Metadata Standards

Metadata standards continue to evolve as different communities of practice design their own metadata schema to serve their unique needs. Although there are some generic metadata schema available for use, such as Dublin Core, the same resource is often described in multiple schema for multiple purposes as well as to serve multiple user groups. For example, a technical report could have a MARC metadata set in a library's online catalogue as well as a set of Dublin Core elements in a digital archive.

Some of the commonly used metadata standards are described below.
3.2.1 Dublin Core

The Dublin Core Metadata Element Set (in short ‘Dublin Core’) provides generic 15 metadata elements through which most resources can be described and cataloged. It can be used to describe physical resources such as books, digital materials such as video, sound, image, or text files, and composite media like web pages. The Dublin Core metadata records are encoded in various forms of XML, including Resource Description Framework (RDF). [20]

The Dublin Core standard was conceived in 1995 in a workshop held in Dublin, Ohio. The basic 15 elements defined in the Dublin Core Metadata Element Set (DCMES) is given below. Each element is optional and is repeatable, and may appear in any order.

1. Title
2. Creator
3. Subject
4. Description
5. Publisher
6. Contributor
7. Date
8. Type
9. Format
10. Identifier
11. Source
12. Language
13. Relation

### 3.2.2 Qualified Dublin Core

The Qualified Dublin Core enhances the original core 15 elements by using a qualifier to add some further levels of detail. The qualifiers enable two types of refinement in Dublin Core:

**Element Refinement:** The qualifiers for element refinement makes the meaning of an element narrower or more specific. For example, the ‘date’ element may include ‘issued’ or ‘modified’ as attributes for refinement.

**Encoding Scheme:** These qualifiers identify the schemes in use (such as controlled vocabularies and formal notations) that help in the interpretation of the data in the element. For example, the ‘subject’ element may include the encoding scheme as ‘DDC’ (Dewey Decimal Classification) or ‘MeSH’ (Medical Subject Heading).

### 3.2.3 MARC

The MAchine-Readable Cataloguing (MARC) is deemed as the oldest metadata standard, originally conceived in 1960s at the US Library of Congress. It was designed to ease the creation physical card catalogues using computers. Subsequently, with computerisation of library catalogues, MARC became not
only the most commonly used metadata format in library catalogues, but in other bibliographic databases as well.

MARC defines a platform-independent schema for metadata representation for both physical and digital objects. However, the field designation may vary based upon the application of MARC standard, which could be for bibliographic information, authority information, holdings information or communities information. MARC records are composed of three sections: the record structure, the content designation, and the data content of the record.

The default MARC record structure is based on the ISO 2709 standard, also known as ANSI/NISO Z39.2. There is also an XML framework for representation of the MARC data known as MARCXML, designed by the Library of Congress. Apart from representing MARC data, the MARCXML framework has also proven useful providing rich metadata support for OAI-compliant repositories and in packaging metadata with electronic resources.

3.2.4 METS

The Metadata Encoding and Transmission Standard (METS) schema is a standard for encoding descriptive, administrative, and structural metadata regarding objects within a digital library. The standard is maintained by the Library of Congress, and is being developed as an initiative of the Digital Library Federation.

METS is used as a content packaging standard due to its flexibility in packaging different kinds of content in variety of ways. According to NISO [16]:

"METS is an XML Schema for creating XML document instances that express the structure of digital library objects, the associated descriptive and administrative metadata, and the names and locations of the files that comprise the digital object."

1http://www.loc.gov/standards/marcxml/
2http://www.loc.gov/standards/mets/
METS Profiles are intended to describe a class of METS documents that helps in creation and packaging of METS documents conforming to a particular profile. Following METS Profiles are currently in use: [24]

- Musical Score
- Print Material (books, pamphlets, etc.)
- Music Manuscript (score or sketches)
- Recorded Event (audio or video)
- PDF Document
- Bibliographic Record
- Photograph
- Compact Disc
- Collection

3.2.5 MODS

The Metadata Object Description Schema (MODS) is an XML-based bibliographic description schema developed by the US Library of Congress as a compromise between the complexity of the MARC format used by libraries and the extreme simplicity of Dublin Core metadata.

MODS can be implemented in multiple scenarios. It is “intended to be able to carry selected data from existing MARC21 records, as well as to enable the creation of original resource description records.” According to NISO [16]:

“Although the MODS standard can stand on its own, it may also complement other metadata formats. Because of its flexibility and

http://www.loc.gov/standards/mods/
use of XML, MODS may potentially be used as a Z39.50 Next Generation specified format, an extension schema to METS, a metadata set for harvesting, and for creating original resource metadata records in an XML syntax."

### 3.2.6 EAD

The Encoded Archival Description (EAD) is a standard for describing collections held by archives and special collections. EAD originated in 1993, at the University of California, Berkeley and is maintained by the Library of Congress in partnership with the Society of American Archivists. [21]

The EAD uses XML format having a tag set of 146 elements which can be used to describe both a collection as a whole as well as a detailed multi-level inventory of the collection. Many descriptive EAD elements have been mapped to content standards (such as MARC or Dublin Core), increasing its flexibility and interoperability.

### 3.2.7 TEI

Text Encoding Initiative (TEI) is a standard to represent electronic text, such as novels, plays, and poetry. Although mostly used in humanities and social sciences, TEI guidelines is now a widely-used standard for online text materials used in teaching and research.

According to NISO [16]:

"It is assumed that TEI-encoded texts are electronic versions of printed texts. Therefore the TEI Header can be used to record bibliographic information about both the electronic version of the text and about the non-electronic source version. The basic bibliographic information is similar to that recorded in library cataloging and can be mapped to and from MARC."
CHAPTER 3. METADATA SOURCES

3.2.8 PREMIS

PREservation Metadata: Implementation Strategies (PREMIS) is an international working group for developing metadata for use in digital preservation. Sponsored by OCLC and RLG from 2003 to 2005, this group produced the PREMIS Data Dictionary for Preservation Metadata report. PREMIS Data Dictionary only defines the metadata needed for preservation functions being agnostic to the descriptive metadata of objects. The Library of Congress maintains a schema for representing PREMIS in XML.4

When people refer to PREMIS, they usually refer to the PREMIS Data Dictionary for Preservation Metadata, which defines the preservation metadata elements. The preservation functions of repositories differ on level of implementation, but it will generally include actions to ensure long-term viability of digital objects. It also ensures that the objects can be rendered in future as well as ensure the integrity of digital objects in the repository and support version tracking. (Caplan 2009) [3]

3.2.9 PRISM

PRISM (Publishing Requirements for Industry Standard Metadata) is a publishing industry specification by IDEAlliance5 that defines an XML metadata vocabulary for managing, post-processing, multi-purposing and aggregating publishing content for magazine and journal publishing.

The PRISM Working Group released PRISM 1.0 in 2001 and PRISM 2.0 in 2008. According to the FAQ,6 PRISM describes many components of print, online, mobile, and multimedia content including the following:

- Who created, contributed to, and owns the rights to the content?

---

4http://www.loc.gov/standards/premis/schemas.html
5http://www.idealliance.org/
6http://www.prismstandard.org/faq/
CHAPTER 3. METADATA SOURCES

- What locations, organizations, topics, people, and/or events it covers, the media it contains, and under what conditions it may be reproduced?
- When it was published or withdrawn?
- Where it can be republished and the original platform on which it appeared?
- How it can be reused?

PRISM allows its specifications to be encapsulated into other metadata carriers. The most popular carrier of PRISM metadata has been the RSS format, which is used for content syndication in XML. PRISM incorporates Dublin Core elements for descriptive elements in addition to the core PRISM elements.

3.2.10 Metadata Crosswalk

It is obvious that a single metadata schema will not be adequate for different communities and collections. Hence, several communities of practice develop their own metadata standard to serve their unique needs. While different metadata schema solves many problems within communities and collections, it creates problems for aggregation services. Aggregation services constantly need to find out crosswalk strategies for integrating disparate metadata schema into their unified collection.

A crosswalk is the mapping of elements, semantics, and syntax from one metadata schema to those of another. The mapping of a metadata schema having fewer elements (less granularity) to those having more elements (more granularity) is invariably troublesome. The crosswalk strategies prepares a common ground for various metadata schema to interoperate, often with a compromise formula.
CHAPTER 3. METADATA SOURCES

3.3 OAI Data Providers

The Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) is an interface that allows a collection or repository to expose the metadata of the items hosted by the collection. Using the same protocol external agents or applications can collect the metadata to build value-added services, such as a subject gateway. The term 'open' in the context of OAI protocol indicates openness of the method of metadata exchange; it does not impose open access requirement on the content itself. Also, the term 'open' is not equivalent to 'free' within the scope of the OAI-PMH.

3.3.1 Definition and Concepts

The Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) provides an application-independent interoperability framework for metadata exchange. In the OAI-PMH framework, there are two classes of participants:

Data Providers are the owners of the metadata and in most cases maintain the content as well ('repository' function). They expose their metadata to Service Providers (or harvesters) through OAI-PMH.

Service Providers harvest the metadata from Data Providers using the OAI-PMH as a basis for building value-added services. The Service Provider, in turn, may expose its harvested metadata using the same OAI protocol. Therefore Service Providers can play the additional role of Data Provider as well.

The Metadata

In the context of OAI-PMH, there are three entities related to the metadata made accessible by the protocol, which are:
**Resource:** A resource is the digital object that metadata describes. The resource can be physical (books), digital, or a web resource (URL).

**Item:** An item is a component of a repository from which metadata about a resource (a Record) can be generated. Examples of items are MARC records and database records.

**Record:** A record is a metadata record in a specific metadata format (e.g., Dublin Core), encoded in XML.

**Unique Identifier**

A unique identifier, as the name suggests, uniquely identifies an item within a repository or collection. Effectively, all possible metadata records available from a single item, if the collection supports multiple metadata formats, share the same unique identifier. The unique identifier is also used in the OAI-PMH requests for extracting metadata from a specific item.

The format of the unique identifier must correspond to a Uniform Resource Identifier (URI) syntax. For example, the DSpace software promotes the use of Handle system as the URI scheme for the unique identifier.

**Metadata Record**

A record is metadata expressed in a single format, encoded in XML. When repository supports multiple metadata formats, the metadataPrefix argument of the OAI-PMH request identifies the metadata format being requested. The metadata records encoded within the <record> element of the response, each of which have two mandatory elements [9]:

- **header** element contains the unique identifier of the item, the datestamp, and the set(s) the item is part of. It may also contain withdrawal or deletion details of the item, if applicable.
The metadata element is a single manifestation (e.g., in Dublin Core format) of the metadata from an item.

Each metadata record may also contain an <about> element with following parts:

- rights statements: information about community-defined rights statements.
- provenance statements: specify whether the record is a harvested record and if so from which repository, and when.

### 3.3.2 Metadata Schema

The protocol requires that all OAI Data Providers offer unqualified Dublin Core metadata (encoded in XML) as a lowest common denominator, which has been the key to the wide success of the protocol. However, repositories are encouraged to offer metadata in additional metadata schema as suitable for its community, which a harvester can request and make use of. The protocol allows a harvester to obtain a list of metadata schema the repository supports.

> "It is certainly clear that almost any metadata scheme can be "downgraded" into unqualified Dublin Core (in a non-reversible fashion). This is a very powerful method of enabling communities with common metadata to work together while still ensuring some minimal level of interoperability for very broad federated search or other services based on unqualified Dublin Core." [12]

### 3.3.3 OAI Metadata Problems

As mentioned above, simple (unqualified) Dublin Core is the minimum requirement for implementing OAI protocol. In practice, however, the simple Dublin Core is too simple for many purposes.
CHAPTER 3. METADATA SOURCES

Some of the problems of unqualified Dublin Core (oai_dc) in OAI-PMH are:

- The identifier field in the oai_dc is used for providing the resource location (URL) as well as for the citation of the item.

- The date field is repeated for various types of dates, such as accession date, date of issue, etc.

- In many cases, the unqualified Dublin Core is generated by downgrading a richer metadata format, which result in incorrect mapping into wrong DC element. (Tennant 2004) [18]

- There are encoding issues in many fields creating problems for mapping. For example, the date field may contain 2004-01-17T21:48:33Z, or Jan 17, 2004, or simply 2004.

While working on a project on collecting metadata from OAI-compliant repositories in Dublin Core format, the National Science Digital Library of USA faced number of problems (Dushay 2003) [4]. Some of the problems are listed below:

**Missing Data**

In most of the records the data missing from the Dublin Core was format and type elements. It happens due to two major reasons: a) the entire collection consisted of materials in one format or of one type, and hence it was deemed unnecessary for the local purposes; and b) the metadata was prepared from an earlier source which did not include this information.

**Incorrect Data**

Among other problems, NSDL found that the creator names repeated in the language element, or the identifier for the metadata record repeated in the
Dublin Core identifier element. There were also many instances where misinterpretations of Dublin Core definitions resulting in wrong data fed into some elements.

**Inconsistent Data**

NSDL found various defaulted strings signifying missing or unknown data. Some examples are:

1. `<dc:description>unknown</dc:description>`
2. `<dc:description> — </dc:description>`
3. `<dc:description> ... </dc:description>`
4. `<dc:description>No abstract available. </dc:description>`
5. `<dc:source>No source: created in machine-readable format</dc:source>`

### 3.3.4 OAI Best Practices

In 2004, a group of OAI Data and Service Providers affiliated to the Digital Library Federation (DLF) and the National Science Digital Library (NSDL) as well as other interested individuals gathered to discuss on the issues on OAI implementations and concerns stemming from the harvesting of metadata from diverse collections. The result of that discussion is the best practices document for the OAI data provider implementations.

The OAI best practices for OAI implementations are available as a wiki [5], which says:

"These best practices are designed to provide data and service providers with the information needed to create resources that are consistent across repositories and thereby support the user in finding the variety of rich resources available and to aid in the implementation of OAI data provider services."
3.4 Z39.50 Targets

The most traditional source of metadata are the Z39.50 targets. Most of the legacy databases, especially library catalogues, support Z39.50 protocol for metadata searching. According to McCallum (2006), more than half of the search queries handled every day by the Library of Congress catalogue come via the Z39.50 protocol.

The protocol is sponsored by American National Standards Institute (ANSI) and US National Information Standards Organization (NISO); hence, the protocol is officially known as ANSI/NISO Z39.50. The standard is currently maintained by the US Library of Congress.

3.4.1 Brief History of Z39.50

According to the Z39.50 primer released by NISO [15]:

"Z39.50 grew out of the Linked Systems Project (LSP), an initiative in the 1980s to standardize searching of the major bibliographic databases of OCLC, the Library of Congress, the Washington (Western) Library Network (WLN), and the Research Libraries Information Network of the Research Libraries Group (RLG). Working in parallel to the LSP initiative was the standardization effort around an information retrieval protocol for library applications under the auspices of NISO. The protocol developed from the LSP moved to NISO and was further developed into the Z39.50 information retrieval standard, approved as a NISO standard in 1988."

The early implementers of Z39.50 worked on enhancing and expanding the protocol features. NISO released a revised versions of Z39.50 in 1992, 1995, and then in 2003, which is the current version of the standard.
3.4.2 Definition and Concepts

According to the ANSI/NISO Z39.50-2003 standard document [14]:

"The Information Retrieval service definition describes an activity between two applications: an initiating application, the client, and a responding application, the server. The server is associated with one or more databases. Communication between the client and server is carried out by the Z39.50 protocol."

Z39.50 standardizes the communication between the client and the server independent of the operating systems, search platforms, databases, and programming languages. It standardizes the procedures and features for searching and retrieving information. This standard is mostly intended for large information systems and services such as library catalogues, content repositories, and union catalogues.

3.4.3 Z39.50 Processes

The Z39.50-2003 standard provides nine operation types: Init, Search, Present, Delete, Scan, Sort, Resource-report, Extended-services, and Duplicate Detection. Each request that initiates an operation is called an initiating request and a response that ends an operation is called a terminating response. From the client perspective, every operation begins when it issues the initiating request, and ends when it receives the terminating response.

The major operations are described below [14]:

**Init** allows the client to invoke a session with the server (known as Z-association).

Once the Z-association is established the clients may proceed with other operations.
Search is invoked by the client to search the database. A search request specifies one or more databases and a query term. It may also include the attributes of those search terms (e.g., author or title etc.). The server responds with a result set.

Present is initiated by the client when the user wants to display records listed in the result set. The Present request may include choices about data elements and format (e.g., MARCXML) for transferring the record from the server to the client.

Delete allows the client to instruct the server to delete one or more result sets that have been created in the server during the Z-association.

Scan provides the client the ability to scan lists of terms for a certain letter or keyword available from one or more databases.

Sort allows the client to request the server to sort and order a result set according to client-supplied criteria.

Resource-report allows the client to request from the server details of resource usage pertaining to a completed operation or to the Z-association.

Extended-services defines a set of tasks or operations that the client may request the server to perform, such as: saving a result set for later use, executing search queries on a periodic schedule, exporting the records in a result set, ordering documents, and requesting printing.

Duplicate Detection allows the client to request the server to analyse one or more result sets in terms of potential duplicates and to construct a new result set according to client-specified criteria for detecting, retaining, grouping, and ordering the records including duplicates.

In a typical Z39.50 Search transaction, the Z39.50 client is responsible for translating the query text into a standardized representation before passing it to a server. The server executes the search against one or more databases
depending upon the request. Once the result set is created, the client can then ask for records from the result set or request additional processing of the result set such as Sort. Upon receipt of the records, the client display records to the user in certain format depending upon the setup.

3.4.4 Other transactions

Based on the requirements of Z39.50 implementers and users, following are the enhancements that are available in the Z39.50-2003:

**Explain** allows the client to obtain information about the implementation of a Z39.50 server including the databases available for searching. The client can also request information about restrictions on the use of the server, hours of operation and availability and other important information which it can use.

**Segmentation** provides for the transfer of records in parts when the entire record exceeds the negotiated transfer size; this is especially critical for databases which disseminate image and other digital objects.

**Proximity Searching** a query type that enables a client to specify proximity search, in which two or more matching term must occur within a specified distance.

**New Record Syntaxes** defines several new record syntaxes that a Z39.50 server can use to package records for transfer to the client including:

- *Simple Unstructured Text Record Syntax (SUTRS):* intended for formatted textual data suitable for display by the client with little or no processing.
- *OPAC Record Syntax:* includes holdings and circulation data.
- *Generic Record Syntax (GRS):* a general-purpose format for records of varying complexity.
3.5 SRU Servers

Search/Retrieval via URL (SRU) is a standard search protocol that uses the web technologies such as Hypertext Transfer Protocol (HTTP) and Simple Object Access Protocol (SOAP) to transfer the messages between the client and server. Being highly influenced by the Z39.50 protocol, much of the functionality of SRU is derived from the older protocol. However, only the most useful features of Z39.50 were brought into SRU, which were simplified to take advantage of the ubiquitous web technologies.

3.5.1 SRU Operations

There are following three important SRU operations which are conspicuous by being very similar to its Z39.50 counterpart:

**searchRetrieve** The searchRetrieve operation is the main operation in SRU, which combine *Search* and *Present* operations of Z39.50. It allows the client to submit a search and retrieve request for matching records from the server. Each request must include the SRU version of client and the query string in CQL format. The CQL query syntax explained in next subsection.

**Explain** Similar to its counterpart in the Z39.50 protocol, the Explain operation allows the client to request information about the facilities available at an SRU server. The information from Explain response can be used by the client to self-configure and provide appropriate services to the user.

**Scan** Similar to Z39.50 Scan, the SRU Scan operation allows the client to request a range of indexed terms for a given term. The server responds with an ordered list of terms which may also include the number of hits for each term, if supported by the server. This can be useful since through Scan some of the terms are selected for subsequent searching.
3.5.2 CQL

SRU uses a textual query format which is at once intuitive and powerful. This format is known as Contextual Query Language (CQL), formerly known as Common Query Language. The CQL provides query building on commonly used access points, such as title, author, subject and further refinements on fields such as personal author, uniform title, geographical subject. Examples of CQL are:

\[
dc.title \text{ any fish}
\]
\[
dc.title \text{ any fish or dc.creator any sanderson}
\]
\[
dc.title \text{ any fish sortBy dc.date/sort.ascending}
\]
\[
> \text{ dc = "info:srw/context-sets/1/dc-v1.1" dc.title any fish}
\]

Unlike the query syntax commonly used with Z39.50, CQL is intended to be human readable and writable, while maintaining flexibility of the Z39.50 query language. According to the CQL Specification [10]:

"CQL tries to combine simplicity and intuitiveness of expression for simple, every day queries, with the richness of more expressive languages to accommodate complex concepts when necessary."

3.6 RSS Feeds

RSS (most commonly translated as “Really Simple Syndication” but sometimes referred to as “Rich Site Summary”) is a family of web feed formats used to publish frequently updated works such as blog entries, news headlines, audio, and video in a standardized format. An RSS document, which is called a feed, a web feed, or a channel, includes full or summarized text, and other metadata such as publishing dates, creator, and the location (URL) of the content.
RSS is a method used by various types of websites to syndicate web-based content encoded in XML. The power of RSS comes from its simplicity of implementation. Any web-based information provider can prepare a simple XML document describing the contents along with its URL using the RSS specification. This XML document may then be released as an RSS feed to be consumed by various applications to get access to the content specified in the feed.

### 3.6.1 Brief History

Named as *RDF Site Summary*, the first version of RSS was created at Netscape in 1999 for use in their my.netscape.com portal. As RSS was being embraced by web publishers who wanted their feeds to be used on the Netscape and other RSS portals, Netscape dropped RSS support from their portal in April 2001 after it was acquired by AOL. However, the simplicity of RSS format caught the imagination of web publishing industry resulting in multiple groups developing and refining the RSS specifications. After various revisions and versions, in 2003 the version 2.0 of the RSS was frozen by the *Berkman Center for Internet and Society* of the Harvard University. [25]

### 3.6.2 Feed Elements

An RSS 2.0 document, encoded in XML, is encapsulated in the `<rss>` element, which must contain one `<channel>` element describing the publishers details and may contain multiple `<item>` elements listing the contents. Following sections describe the various elements available in an RSS feed. [26]

**Required channel elements**

**title:** This element is used for the name of the channel, usually same as the website title. For example, *‘D-Lib Magazine’*
CHAPTER 3. METADATA SOURCES

link: This element contains the URL of the website corresponding to the channel. For example, http://www.dlib.org/

description: It contains a phrase or a sentence describing the channel. For example, ‘An electronic publication with a primary focus on digital library research and development.’

Optional channel elements

language: Contains the language the channel. For example, en-us.

copyright: Contains the copyright notice for content in the channel.

managingEditor: Contains email address of the person responsible for editorial content.

webMaster: Contains email address of the person responsible for technical issues relating to the channel.

category: Contains one or more categories that the channel belongs to. It does not have to adhere to any controlled vocabulary.

generator: Contains name of the program used for generating the feed.

docs: Contains a URL that points to the documentation for the format used in the RSS file.

cloud: Contains a web-service for RSS clients to check for updates to the channel.
ttl: The time to live (ttl) is the number of minutes that indicates how long a channel can be cached (local storage) by RSS clients before refreshing from the source.

image: Contains the location of a GIF, JPEG or PNG image that can be displayed with the channel.

rating: Contains the PICS (Platform for Internet Content Selection) rating for the channel. These ratings indicate the suitable audience for the content, similar to the ratings used by movies.

textInput: Contains a text input box that can be displayed with the channel.

skipHours: Contains information for aggregators to know which hours they can skip.

skipDays: Contains information for aggregators to know which days they can skip.

Every RSS feed document is likely to contain several <item> elements for various content units. Following are the elements available for each <item> entry.

**Required item elements**

title: Contains the title of the item. For example, ‘Digital cultural collections in an age of reuse and remixes’.

link: Contains the URL of the item. For example, http://dx.doi.org/10.1045/september2010-editorial.

description: Contains the item synopsis or abstract.
Optional item elements

**author:** Contains email address of the author of the item.

**category:** Contains one or more categories the item may belong to.

**comments:** Contains the URL of a page for comments relating to the item. Mostly useful for news, magazines, and blogging sites.

**enclosure:** Describes a media object that is attached to the item. Used mostly for links to podcasts.

**guid:** Contains a unique identifier for the item: useful in archival.

**pubDate:** Contains the publication date of the item.

**source:** Contains name of the RSS channel the item came from; useful for external content.

### 3.6.3 Observations

By design RSS feeds are transient documents that provide information about latest publications. As new publications are added, old publications are removed from an RSS feed. Hence,

1. It is a useful resource to get quick access to metadata of latest publications.

2. It is not suitable for retrospective indexing of old publications.

### 3.7 Atom Feeds

The Atom format was developed as an alternative to the RSS format. However, the name Atom is used for a pair of standards: a) an XML-based web content and metadata syndication format, and b) an application-level protocol (known as AtomPub) for publishing and editing web resources.
3.7.1 Brief History

The RSS 2.0 specification is copyrighted by Harvard University and is frozen. Although the specification is available under a Creative Commons license, no significant changes can be made in this standard due to its need to remain backward compatible. Ben Trott, an advocate of the new format that became Atom, believed that RSS had limitations and flaws and it needed a fresh design. [19]

Atom 1.0 syndication format is specified in RFC 4287, which was published in December 2005.

3.7.2 Feed Elements

An Atom feed document, encoded in XML like RSS, is encapsulated within a <feed> element. The sub-elements of the root <feed> element are categorized into three types, required, recommended and optional as described below.

Required feed elements

Following are the required elements inside the <feed> element:

id: Contains a unique identifier for the feed using a universally unique and permanent URI. Usually, it is same as the publisher's website URL.
   For example, <id>http://www.dlib.org/</id>.

title: Contains the title for the feed, often same as the title of the publisher's website.
       For example, D-Lib Magazine.

updated: Indicates the last time the feed was modified in a significant way.
         For example, 2010-04-04T15:33:04Z
CHAPTER 3. METADATA SOURCES

Recommended feed elements

author: Contains author details of a feed. It is optional if all the <entry> elements have <author> defined for them. This element may contain three sub-elements, viz., <name>, <email>, and <uri> for name, email, and homepage of the author respectively. For example,

```xml
<author>
  <name>Laurence Lannom</name>
  <email>llannom@cnri.reston.va.us</email>
</author>
```

link: Contains URL of a related web page for the feed. The type of relation is defined by the rel attribute. Usually, this is used for a link back to the feed itself or the homepage of the publisher.

Optional feed elements

category: Contains name a category that the feed belongs to. A feed may have multiple category elements.

contributor: Contains name of the contributor to the feed. A feed may have multiple contributor elements.

generator: Contains name of the software used to generate the feed.

icon: Contains URL of a small image icon which provides iconic visual identification for the feed.

logo: Contains URL of a larger image which provides visual identification for the feed.

rights: Contains information about rights, e.g. copyrights, held in and over the feed.
**subtitle:** Contains a small description or subtitle for the feed.

Every Atom feed document is likely to contain several `<entry>` elements for various content units. Following are the sub-elements available for each `<entry>` element.

**Required entry elements**

**id:** Identifies the entry using a universally unique and permanent URI. For example, `<id>http://www.dlib.org/dlib/september10/</id>

**title:** Contains the title for the entry, often same as the title of the individual content feed. For example,

```
<title>RDA Vocabularies: Process, Outcome, Use</title>
```

**updated:** Contains the date and time the item was modified last time in a significant way. For example,

```
<updated>2009-06-03T21:41:13Z</updated>
```

**Recommended entry elements**

**author:** Contains name(s) of author for the item. Each `<entry>` element must contain at least one `<author>` element. Otherwise, at least one `<author>` element must be defined at `<feed>` level. This element may contain three sub-elements, viz., `<name>`, `<email>`, and `<uri>` for name, email, and homepage of the author respectively.

**content:** Contains the complete content of the item. It is mandatory if the `<link>` is empty.

**link:** Contains URL of a related web page for the item. The type of relation is defined by the `rel` attribute. For example,
CHAPTER 3. METADATA SOURCES

The `<link>` element is mandatory if the `<content>` element is empty.

**summary:** Contains a short summary, abstract, or excerpt of the entry.

**Optional entry elements**

**category:** Contains name a category that the `<entry>` may belong to. A entry may have multiple category elements.

For example, `<category term="technology"/>`.

**contributor:** Contains name of the contributor to the entry. An entry may have multiple `<contributor>` elements.

For example,

```xml
<contributor>
  <name>John Smith</name>
</contributor>
```

**published:** Contains the date and time of the initial creation or first availability of the entry.

For example, `<published>2010-1-13T09:51:08:00</published>`.

**source:** If an `<entry>` is copied from another feed, then the source feed's metadata is preserved in this element.

**rights:** Contains information about rights, e.g. copyrights, held in and over the entry.

**3.7.3 Comparison with RSS**

The following is a basic comparison of Atom 1.0 with RSS 2.0.
CHAPTER 3. METADATA SOURCES

Content model and payload

RSS 2.0 may contain either plain text or escaped HTML as a payload, without any content designation to indicate which of the two is provided. Escaped HTML (for example, the string ‘AT&T’ would be expressed as ‘AT&amp;amp;amp;T’) has been a source of difficulty for implementers. The RSS content model does not permit actual well-formed XML markup, which reduces the re-usability.

Atom 1.0 format not only allows for a broad variety of payload types but also provides a mechanism to specify the type of content being provided by the entry. The content must be explicitly labelled as one of the following:

- plain text
- escaped HTML
- well-formed XHTML markup
- some other XML vocabulary
- base64-encoded binary content
- link to external web content

The content designation guarantees that the recipients of the Atom feed will be able to process content appropriately.

Date/Time formats

For formatted timestamps for date and time of the feed, the RSS documents use the RFC 822 specification, which was written in 1982. The Atom working group chose to use timestamps formatted according to the rules specified by RFC 3339, which was specifically designed for Date and Time on the Internet in 2002 and is much easier to parse with modern applications.
Internationalization

While the RSS vocabulary has a mechanism to indicate the language of the feed, there is no way to specify a language for individual items or text elements.

Atom, on the other hand, uses the standard xml:lang attribute to make it possible to specify a language context for all the content related elements in a feed. Atom also allows links and unique identifiers to contain non-ASCII characters, providing better internationalization support.

Required Content

RSS requires title, link, and description elements at the feed level, but does not make any of these mandatory in the individual items in a feed.

Atom requires title, a unique identifier, and a last-updated timestamp both at the feed level as well as at the individual items level.

Full or Partial Content

Within an RSS document the <description> element is commonly used to contain either the full text of an entry or a synopsis, without any built-in way to signal what it contains.

Atom provides separate elements (<summary> and <content>) for these purposes.

Autodiscovery

RSS autodiscovery has been source of difficulty for implementers, since it often relies on unregistered (thus invalid) application/rss+xml MIME type, and has never been standardized.
Atom uses application/atom+xml MIME Type, which is registered with Internet Assigned Numbers Authority (IANA). Additionally, Atom feeds support auto-subscribe from the contents of the feed thanks to its self pointers.

**Extensibility**

RSS does not have its own XML namespace but may contain elements from other XML namespaces. There is no clear documentation to find out usage instructions for popular extensions, such as dc:creator and content:encoded.

Atom has its own XML namespace and may contain elements or attributes from other XML namespaces. There are specific guidelines in the standard (RFC 4287)\(^7\) on how to interpret extension elements. Additionally, Atom provides recommended extension points for using non-Atom vocabularies.

**URIs**

RSS does not specify the handling of relative URI references. Different RSS client software (i.e., feed readers) resort to various heuristics for their interpretation.

Atom specifies use of the XML's built-in xml:base attribute for the use of relative URI references.

The table 3.1 shows fields-level comparison between RSS 2.0 and Atom 1.0 [1].

### 3.7.4 Observations

Following are the observations on RSS and Atom feeds:

- Both formats provide structural data through XML.

\(^7\)http://www.ietf.org/rfc/rfc4287.txt
Table 3.1: Atom 1.0 vs RSS 2.0 Comparison table

<table>
<thead>
<tr>
<th></th>
<th>RSS 2.0</th>
<th>Atom 1.0</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>rss</td>
<td>-</td>
<td>feed</td>
<td>Vestigial in RSS</td>
</tr>
<tr>
<td>channel</td>
<td>feed</td>
<td>title</td>
<td></td>
</tr>
<tr>
<td>title</td>
<td>-</td>
<td>link</td>
<td>Atom defines an extensible family of rel values</td>
</tr>
<tr>
<td>link</td>
<td>subtitle</td>
<td>language</td>
<td>Atom uses standard xml:lang attribute</td>
</tr>
<tr>
<td>description</td>
<td>-</td>
<td>copyright</td>
<td></td>
</tr>
<tr>
<td>language</td>
<td>-</td>
<td>webMaster</td>
<td></td>
</tr>
<tr>
<td>copyright</td>
<td>rights</td>
<td>managingEditor</td>
<td>author or contributor</td>
</tr>
<tr>
<td>webMaster</td>
<td>-</td>
<td>managingEditor</td>
<td>author or contributor</td>
</tr>
<tr>
<td>pubDate</td>
<td>published (in entry)</td>
<td>generator</td>
<td></td>
</tr>
<tr>
<td>lastBuildDate</td>
<td>updated</td>
<td>generator</td>
<td></td>
</tr>
<tr>
<td>(in channel)</td>
<td></td>
<td>generator</td>
<td></td>
</tr>
<tr>
<td>category</td>
<td>category</td>
<td>generator</td>
<td></td>
</tr>
<tr>
<td>generator</td>
<td>generator</td>
<td>generator</td>
<td></td>
</tr>
<tr>
<td>docs</td>
<td>-</td>
<td>cloud</td>
<td></td>
</tr>
<tr>
<td>cloud</td>
<td>-</td>
<td>ttl</td>
<td>&lt;ttl&gt; is problematic. prefer HTTP 1.1 cache control</td>
</tr>
<tr>
<td>ttl</td>
<td>-</td>
<td>image</td>
<td>Atom recommends 2:1 aspect ratio</td>
</tr>
<tr>
<td>image</td>
<td>logo</td>
<td>-</td>
<td>As in favicon.ico</td>
</tr>
<tr>
<td>-</td>
<td>icon</td>
<td>rating</td>
<td></td>
</tr>
<tr>
<td>rating</td>
<td>-</td>
<td>textInput</td>
<td></td>
</tr>
<tr>
<td>textInput</td>
<td>-</td>
<td>skipHours</td>
<td></td>
</tr>
<tr>
<td>skipHours</td>
<td>-</td>
<td>skipDays</td>
<td></td>
</tr>
<tr>
<td>skipDays</td>
<td>-</td>
<td>item</td>
<td></td>
</tr>
<tr>
<td>item</td>
<td>entry</td>
<td>author</td>
<td></td>
</tr>
<tr>
<td>author</td>
<td>author</td>
<td>-</td>
<td>contributor</td>
</tr>
<tr>
<td>-</td>
<td>contributor</td>
<td>description</td>
<td>summary and/or content depending on whether full content is provided</td>
</tr>
<tr>
<td>description</td>
<td>summary and/or content</td>
<td>comments</td>
<td></td>
</tr>
<tr>
<td>comments</td>
<td>-</td>
<td>enclosure</td>
<td>rel=&quot;enclosure&quot; on &lt;link&gt; in Atom</td>
</tr>
<tr>
<td>enclosure</td>
<td>-</td>
<td>guid</td>
<td></td>
</tr>
<tr>
<td>guid</td>
<td>id</td>
<td>source</td>
<td>rel=&quot;via&quot; on &lt;link&gt; in Atom</td>
</tr>
<tr>
<td>source</td>
<td>-</td>
<td>-</td>
<td>Container for feed-level meta-data to support aggregation</td>
</tr>
</tbody>
</table>
CHAPTER 3. METADATA SOURCES

- There are multiple versions of both RSS and Atom XML schema, which require multiple parsing rules.

- Most feeds provide only mandatory fields in spite of detailed provisions in the standard.

- Does not specify if it is an article, review, or report.

- The description field is often used as 'catch all' field for myriad types of information.

3.8 Email Alerts

The advent of electronic mail (or email) predates Internet and it became ubiquitous with proliferation of Internet. The use of this medium allowed the publishers to shorten the time to reach its users, and with cheap Internet costs, it saved lot of money as well. Many journals, print or online, directly or indirectly (through aggregators), provide email alerts to subscribers when a new issue is published. These alerts usually contain the title of each article published in the issue along with the author names, sometimes accompanied with a link to the online version of the article.

These email alerts are important source for the current awareness services provided by libraries. Similar to RSS/Atom feeds, any user who wants to be kept updated about the new publications can subscribe to the email alerts service, usually through a web-based form. The advantage of email alert over RSS/Atom feeds is that the user does not require any separate application, since it gets delivered to the regular email service as soon as the content is ready.

However, unlike RSS/Atom feed, the email alerts does not conform to any fixed format. There is no standard governing email alerts. Hence, publishers may include any content in any format they seem fit for their need.
The Fig. 3.1 shows the screenshot of a sample email alert.

The Spring 2010 issue of Issues in Science & Technology Librarianship is now available at http://www.istl.org/

CONTENTS

Board Accepted Articles

Five Voices, Two Perspectives: Integrating Student Librarians into a Science and Engineering Library
by Eugene Barsky, Aletela Greenwood, Samantha Sinanan, Lindsay Tripp, and Lindsay Willson, University of British Columbia

Refereed Articles

Collection Assessment in Response to Changing Curricula: An Analysis of the Biotechnology Resources at the University of Colorado at Boulder
by Gabrielle Wiersma, University of Colorado at Boulder

Browsing of E-Journals by Engineering Faculty
by Denise Beaubien Bennett and Amy G. Buhler, University of Florida

Sequencing Genetics Information: Integrating Data into Information Literacy for Undergraduate Biology Students
by Don MacMillan, University of Calgary

Book Reviews

Scientific Writing and Communication: Papers, Proposals, and Presentations
by Mindy Thuna, University of Toronto Mississauga

The Extreme Searcher's Internet Handbook
by Margaret Henderson, Virginia Commonwealth University

Human Information Retrieval
by David Hook, MDA

Figure 3.1: Sample Email Alert
CHAPTER 3. METADATA SOURCES

3.8.1 Observations

Following are some of the issues to consider while dealing with email alerts as a metadata source:

- Although email alerts provide information about the latest issue, the emails could be archived. However, collecting emails on a regular basis could be a concern due to email clients and archive formats are being platform specific.

- The format of emails in an archive vary depending upon the email client. For example, Unix uses mbox and Microsoft uses PST format which are not compatible with each other.

- Many mail clients do not accept rich HTML-encoded emails. In cases where email alert is available in rich HTML format, the mail clients may convert it into simple text.

- In general, email alerts are always publisher defined format which require building custom heuristics and parsing rules for each publisher.

3.9 Online Table of Contents

The Internet and web technologies provide a perfect platform for publishers to provide latest content to its subscribers. Many journal publishers have started publishing online version of their print journals, which are commonly known as electronic journal or e-journal. The online medium provides significant cost savings to publishers to publish e-journals not only in printing and binding cost but also in the delivery and supply chain.

Most of the e-journals publishers, commercial or open access, make the table of contents freely available over the Internet. These pages contain the link to each article along with title, author names and in many cases it provides abstract
CHAPTER 3. METADATA SOURCES

of the article as well. In case of e-journals, the link, if followed, provides more information about the article, such as, author affiliation, abstract, and bibliography citations. Hence, these table of contents can be a rich source of metadata.

See Fig. 3.2 for the screenshot of a sample table of contents.
## Table of Contents

**SEPTEMBER/OCTOBER 2010**

Volume 16, Number 9/10

* Continuing publication of *D-Lib Magazine* is made possible by the *D-Lib Alliance*.

---

### Editorial

**Repositories and One More Thing**

By Laurence Lynam, Corporation for National Research Initiatives

---

### Articles

**Designing and Implementing Second Generation Digital Preservation Services: A Scalable Model for the Stanford Digital Repository**

Article by Tom Cramer and Katherine Kott, *Stanford University Libraries*

*Abstract:* This paper describes the Stanford Digital Repository (SDR), a large scale, digital preservation system for scholarly materials. It examines the lessons-learned through over five years of development and operational experience. Building on the knowledge gained, the paper goes on to outline a new repository design and service framework, SDR 2.0, that will address some of the challenges that have emerged. Changes in the environment such as staffing levels and collaborative opportunities are also described. Finally, the paper includes observations on the general state of the preservation and repository communities, and the emergence of a new generation of systems and strategies in this space.

**A Checklist and a Case for Documenting PREMIS-METS Decisions in a METS Profile**

Article by Sally Vermaaten, *Statistics New Zealand*

*Abstract:* Shared metadata practices foster preservation and interoperability in several ways. They facilitate inter-repository exchange, the development of reusable metadata tools, and repository self-assessments and audits. Despite the benefits of shared practices, there has been little convergence on best practices for a widely used metadata strategy, PREMIS in METS. This paper proposes documenting PREMIS-METS decisions in METS profiles as a beneficial internal practice and an efficient way of sharing and comparing metadata strategies, thereby facilitating best practices. The paper then introduces a tool to help implementers document PREMIS-METS decisions in a METS profile. This tool is a checklist of 13 key PREMIS-METS issues that a repository should consider documenting in their METS profiles. Each of the 13 issues is illustrated with examples from METS profiles currently registered with the Library of Congress.

**Representation and Recognition of Subject Repositories**

Article by Jessica Adamick and Rebecca Reznik-Zellen, *University of Massachusetts Amherst*

*Abstract:* Subject repositories are under-studied and under-represented in library science literature.

---

---

Figure 3.2: Sample Table of Contents Page
3.9.1 Observations

Following are some of the observations on using Table of Contents (ToC) as a source of metadata:

- Many journals provide retrospective data of all issues in ToC format online.
- Screen-scraping techniques can be used to harvest metadata elements from ToC on a regular basis.
- Web-based ToC can be parsed by studying the HTML formatting. The journals using same software, such as Open Journal System (OJS), are likely to have similar formatting. It is still required to create parsing rules for each journal site.
- Metadata information available from ToCs are non-standard. Hence, heuristics are useful in understanding the fields structure. However, ToCs generated by client side scripting, such as JavaScript, will be difficult to parse.
- ToCs are available HTML format, which is not strict like XML (e.g., closing a tag not mandatory in some cases). Multiple techniques are used in HTML-parsing depending upon the strictness.

3.10 Conclusion

As we have seen above, there are number of information resources where we can find information about scholarly publications. Most of these resources provide different levels of metadata about their publications, which could be harvested to be included in a local index.

Sometimes the same publication may be available through multiple resources. For example, an e-journal may provide RSS feed as well as OAI-compliant in-
CHAPTER 3. METADATA SOURCES

There will be a need to define a unique identifier, so that the metadata from different resources can be merged without any loss of data. Following are some of the options to be considered:

- OAI-compliant sources provide a unique identifier for each metadata record, which is mandatory for protocol compliance. It can be used without any conversion.

- Use the Digital Object Identifier (DOI), if available. However, most journals do not provide DOI.

- Use the resource location (URL) as the identifier. In a Dublin Core record the URL is provided in the identifier field.

- In cases of online ToC that does not have URL for each title, some other scheme need to be used, such as combination of author name, date, and journal abbreviation.

3.11 Bibliography


CHAPTER 3. METADATA SOURCES


