

# Purposes

Benjamin Riesenber

**M**APs serve a range of purposes and offer benefits for metadata creators, managers, and consumers. They can provide a detailed model for reference, a resource that exists outside of the collections, institutions, and technical platforms within which metadata exists. This resource aids the metadata design and creation processes and enables the accurate interpretation of metadata following creation, allowing reuse in new environments. MAPs may be repurposed and reused in whole or in part, saving time and effort and improving metadata quality and consistency across collections, institutions, and platforms. These benefits are discussed in more detail below, along with recommendations for designing and implementing MAPs that best realize them.

## Metadata Implementation

### Benefits during Metadata Design

MAP implementation provides benefits from the beginning of the metadata life cycle. The components of a MAP can provide a clear checklist for completion during the metadata design process. Through the process of defining entities, properties, and values, the many decisions required in order to create a complete metadata model are partitioned into more manageable sets. We can work through a logical series of questions built around the structure of the MAP itself. For example:

- What kind of entities will be described? Do these need to be specifically defined for our application? If so, where are our resource types defined in existing vocabularies, and which definitions

### Vocabularies

Entities and properties for use in a MAP may be drawn from a diverse array of sources. These are often referred to here simply as vocabularies, but they may also be called schemas, ontologies, taxonomies, or thesauruses. These sources themselves can be modeled, described, and documented in many ways. They will be considered in more detail in the chapters on MAP components and sources.

best meet the needs of our application?

- What kind of information do we need to provide about our resources—for discovery, administration, reuse, or other purposes? What properties will we need in order to do this? Can existing elements meet all of our needs?
- How should the values for each property be structured? What values should be required for all resources? What are the local needs for searching and display of these values? When specifying constraints on values, will we prioritize local needs, ease of data reuse, or other criteria?

Because MAPs can define or constrain entities, properties, and values, working through a MAP design process can help ensure that metadata planning is comprehensive. A draft MAP can be shared with stakeholders for feedback prior to metadata creation to better ensure that resource description will meet the needs of users.

### MAP REUSE

Publishing a MAP by making it openly accessible online or sharing it with institutional stakeholders

	Date
<b>Instructions</b>	Enter dates using the following format: YYYY-MM-DD
<b>Examples</b>	For March 12, 1976, enter: 1976-03-12
	For December 1, 2020, enter: 2020-12-01

**Figure 2.1**

Data-entry instructions and examples in a human-readable MAP

and engaging with comments and feedback helps move our work toward shared best practices and build consensus with regard to metadata modeling. The fact that MAPs are reusable, either in part or as a whole, is also beneficial. Use or adaptation of an existing MAP for new work can help institutions improve descriptive consistency. This reuse also further propagates best practices and can save time and effort during the metadata design process.

## Benefits during Metadata Creation

If they provide a detailed model and clear guide for generating resource descriptions, MAP implementations will yield significant benefits during metadata creation, whether they are designed for use by humans or for machine processing. Some of these benefits are described below, and a more detailed look at human- and machine-readable MAP presentations is provided in chapter 5.

### HUMAN-READABLE MAPS

Human-readable MAPs are a way to present guidance to metadata creators such as catalogers and other specialists as they work, providing clear and precise rules, answering questions, and resulting in more consistently formed metadata values.

For example, in a human-readable MAP, the specification that values for a date property be formatted in a certain way can be supplemented with example values (see figure 2.1).

By providing clear guidance and examples, human-readable MAPs improve workflows for metadata creation by anticipating and answering questions that come up during the process. If such MAPs provide clear guidance, if they are fully integrated into metadata-creation workflows and used by the staff doing the work, they improve the quality and consistency of metadata values, assisting users by improving the functionality of metadata in discovery systems.

The implementation of a human-readable MAP also provides an opportunity to receive feedback from staff creating resource descriptions. Any differences between what the MAP requires and what can be

generated using local tools for metadata creation will be readily apparent to staff doing the work.

### MACHINE-READABLE MAPS

Human-readable MAPs provide information to catalogers and metadata specialists who are creating metadata, so the conformance of values with constraints depends on the extent to which guidance is understood and adhered to during descriptive work. In contrast, a machine-readable MAP may interact directly with an application used for data entry. Instead of or in addition to providing instructions for the entry of a date value, for example, this approach may generate a data-entry form that does not accept a date value unless it is entered in the YYYY-MM-DD format.

The Sinopia Linked Data Editor offers examples of the ways in which a machine-readable MAP can pass information to a data-entry form.<sup>1</sup> To create resource descriptions with the Editor, a user must first select a MAP, referred to as a resource template within the Sinopia platform. These resource templates encode MAP information in machine-readable form, identifying a single entity type that the template may be used to describe and enumerating properties that may be used to describe it. The resource template also defines constraints on values for each property, such as:

- whether a value is required
- whether multiple values can be entered
- whether the data-entry form should be set up for inputting a literal value (textual or numeric string) or an IRI
- which vocabulary encoding schemes (controlled vocabularies) should be queried for users to select an IRI from

Because linked-data bibliographic descriptions constructed using data models such as BIBFRAME and RDA/RDF require description of multiple conceptual entities to catalog a single item in hand, such as a print monograph, multiple resource templates are

## Vocabulary Encoding Schemes

Some readers may be unfamiliar with the term *vocabulary encoding scheme* and instead accustomed to using the term *controlled vocabulary* to refer to a controlled list of values that may be selected from for use in metadata instances. The Library of Congress Name Authority File is a well-known example. In non-RDF implementations, it is common to specify that textual values should be entered as they are found in a vocabulary encoding scheme (VES). In RDF metadata instances, an IRI will often be retrieved from a VES for use as a value.

## Syntax Encoding Schemes

A syntax encoding scheme (SES) provides specific rules for formatting textual or numeric values (called literals in RDF implementations). The XML Schema datatypes, which include technical specifications for formatting a variety of values such as numbers and dates, are well-known examples.

used to describe a single resource.

Machine-readable MAPs also provide a path back to the sources of MAP components. A Sinopia resource template, for example, includes IRIs for all entities and properties, and each of these can be dereferenced to access further information about source vocabularies.

## Interoperability

Because MAPs can provide a complete and detailed specification of a metadata model, they are essential in supporting interoperability. The Dublin Core Metadata Initiative glossary definition for *interoperability* is “the ability of two or more systems or components to exchange descriptive data about things, and to interpret the descriptive data that has been exchanged in a way that is consistent with the interpretation of the creator of the data.”<sup>2</sup> Following this two-part definition, we can consider interoperable metadata as that which meets two distinct requirements:

- *Semantic interoperability*: The meanings of entities, properties, and values may be accurately understood by someone wishing to reuse metadata in an external application.
- *Technical interoperability*: Encoded metadata is compatible with an external application for reuse, or the original encoding can be successfully interpreted and processed for reuse.

In order for a MAP to play its role in supporting interoperability, effort is required during the design phase, prior to the point of metadata creation. This is needed to answer questions related to each component of a MAP and define each. MAPs resulting from this work serve as guides for local metadata creators and managers who will generate and publish metadata instances, as well as for external stakeholders who will reuse them.

MAP design can best support interoperability by addressing all three components of a metadata model: entities, properties, and values. The definition of entities and properties alone is not sufficient for the goal of supporting metadata reuse in external systems. To support accurate interpretation and reuse of a

metadata instance, a MAP must include well-defined value constraints. This may include requiring that textual or numeric values conform to syntax encoding schemes (SEs) such as data types or other formatting rules, that they come from certain vocabulary encoding schemes (VESs), that values for a given property are required or may or may not be repeated, and so on.

## Assessing Source Vocabularies to Support Interoperability

To model all components of a MAP, implementers will need to assess published vocabularies and select needed elements. By viewing vocabularies and applying criteria from the perspective of interoperability, it is possible to build a MAP using components that support the accurate interpretation and reuse of metadata. Three such criteria are given below and can be applied when assessing sources for entities, properties, syntax encoding schemes, and vocabulary encoding schemes.

### IS THE SOURCE WELL-KNOWN AND WIDELY USED IN THE RELEVANT KNOWLEDGE DOMAIN?

Do many collections in the same domain use terms from the vocabulary? The degree to which source vocabularies are well-known and well-used in relevant knowledge domains is an important factor in selection. Metadata modeled using well-known standards is more likely to be reusable in a wide variety of external applications.

For some implementations general-purpose terms may be desired. The use of properties such as the Dublin Core Metadata Initiative (DCMI) Elements or Terms, which are less specific to any particular knowledge domain and more broadly applicable, may be preferable for some collections.

### IS THE SOURCE CLEARLY DEFINED? IS A PUBLISHED DEFINITION READILY AVAILABLE ONLINE?

Clear definitions of entities, properties, SEs, and VESs provide a basis for agreement on usage among implementers, and this consistent usage enables interoperability.

Terms should be described in clear language. For an SES specifying data types, technical definitions should be provided. Linked-data terms should be modeled in RDF using OWL, RDF Schema, SKOS, or another data-modeling vocabulary or ontology language. It is highly preferable that these definitions be published in well-maintained and easily findable websites that include information about the organization that publishes them (see figure 2.2).

RDA Registry
Elements
Values
Data
Tools
Releases
About
FAQ
Guide
Project
RDA Toolkit

## RDA element sets

### Work properties

Properties that represent the attribute and relationship elements of the RDA Work entity.

Each property in the element set:

- has a domain of the class that represents the Work entity.
- is linked from its child **datatype** property in [RDA Work datatype properties](#) by *rdfs:subPropertyOf*.
- is linked from its child **object** property in [RDA Work object properties](#) by *rdfs:subPropertyOf*.

Number of active elements: 590

Namespace: <http://rdaregistry.info/Elements/w/>

Suggested prefix\*: rdaw

Example curie: rdaw:P10001

\* registered at [prefix.cc](#)

### Downloads

- CSV (text/csv) (English language only)
- JSON-LD (application/json | application/json+ld)
- N-Triples (text/rdf+nt)
- RDF/XML (application/rdf+xml)

### Languages

[Catalan](#)
[Danish](#)
[Dutch](#)
[English](#)
[Finnish](#)
[French](#)
[German](#)
[Hungarian](#)
[Norwegian](#)
[Swedish](#)
[Vietnamese](#)

### Properties Index

Show  entries

Search:

#	CURIE	Label	Definition	Subproperty of
#	<a href="#">rdaw:P10061</a>	"has author agent"	"Relates a work to an agent who is responsible for creating a textual work."	<a href="http://rdaregistry.info/Elements/w/P10065">http://rdaregistry.info/Elements/w/P10065</a> ["has creator agent of work" (en)]

Showing 1 to 1 of 1 entries (filtered from 607 total entries)

Previous  Next

RDA Vocabularies and RDA Registry are licensed under a [Creative Commons Attribution 4.0 International License](#). Based on a work at <http://rdaregistry.info>.

This page was last updated 26 Jan 2021.

**Figure 2.2**  
A property definition viewed online at the RDA Registry, providing a brief overview and links to download RDF definitions

### IS THE SOURCE STABLE?

If the vocabulary has a history of development, this may be a positive indicator of high visibility and engagement with a community of use. However, changes made to term definitions have the potential to change the interpretation of metadata values created using them. If development is continuing—for example, if multiple versions of a vocabulary have been released—is information about these changes available?

### Creating New Terms

In some cases, implementers may not be able to find needed components in published vocabularies. For a project creating descriptions of resources in specialized knowledge domains or one that must meet unique local needs, implementers may find that needed entity types, properties, VESs, or other terms have not

been defined in existing sources. In such cases, they may wish to create new properties to record unique or highly specific attributes, or define a new RDF resource class for description, or create other new terms for use.

To support data reuse in such a case, implementers can provide and publish term definitions in a way that meets the same requirements as those for existing vocabularies. In the case of new RDF classes, properties, or other resources, this means modeling using a linked-data modeling language such as RDF Schema, SKOS, or OWL. Importantly, these modeling tools allow for describing relationships to existing terms, for example, as subclasses and subproperties of existing classes and properties. The identifiers for new terms should be dereferenceable on the web, allowing users of the metadata to access the definitions for each. In non-RDF implementations, clear and easily accessible definitions for new properties, as well as for

any local controlled vocabularies or value-formatting rules, will be important for users to make sense of the data and reuse it in the future.

## Additional Considerations for Assembling MAP Components

Approaching the selection of MAP components using the criteria outlined above supports the creation of well-defined metadata that can be accurately interpreted and successfully reused. Moving into completing the assembly and implementation of a MAP, there are additional considerations. These are based upon the selection of MAP components and concern the ways in which they fit together.

### *PAIRING ENTITIES AND PROPERTIES*

Working through entities, properties, and values—the three facets of a metadata model that a MAP can define—and how they fit together, it may be best to start with the relationship of entities to properties. When combining published entity classes (in the case of RDF implementations) and properties, the question of whether or not a given property may be used with a given entity requires looking at the specifications provided in the source for each.

In non-RDF implementations, the usage of properties to describe a given resource type should not conflict with published definitions or guidance for the properties in question, or with a commonsense understanding of the entity and properties alike. In an RDF implementation, property definitions will often include a domain, indicating a resource class or classes that a property may be used to describe. Note that these may also have subclasses that will fall within the property's domain. This value should be checked against the resource type of the entity that the property is to be used to describe.

### *PAIRING PROPERTIES AND VALUE CONSTRAINTS*

It is essential that definitions and guidance for properties do not conflict with those for the SESs and VESs used to constrain their values. In non-RDF implementations, the combination of a given property with a value constraint or constraints should not conflict with a commonsense understanding of the meaning of the property and its expected value. In linked-data implementations, property definitions will often include a range, indicating the class or classes of RDF resources that are allowed as a value for the property. If IRIs from a given VES will be used with the property, the class or classes defined for those resources should be checked against the property's range.

From the perspective of data reuse, any constraints included in published definitions of properties

can be considered a minimum or baseline to which values are expected to conform. Implementing value constraints for a property in a MAP that match or are more restrictive than those in the property's definition can be expected to facilitate interoperability for a created metadata instance. Requirements or constraints that contradict those provided in published definitions can be expected to cause problems for data reuse.

## Supporting External Applications

Much of the work required to support the creation of metadata instances that can be accurately interpreted and reused successfully takes place as a MAP is designed for use, prior to the creation of any metadata. The benefits of this work are realized after the creation of metadata instances based on the MAP, when metadata is harvested or exported and processed for reuse in an external system. Metadata instances that conform to MAPs designed as described above will be clearly defined semantically and technically and well-suited for reuse.

### Supporting Semantic Interoperability

If the meaning of values in a metadata instance can be accurately interpreted by agents who wish to reuse it in new applications or environments, it is semantically interoperable. This interpretation should be possible given a reasonable amount of effort and will be based not only on the instance itself but also on documentation such as a MAP. The following characteristics of metadata instances created in conformance with MAPs support semantic interoperability:

- Clear identification of MAP components adopted for use from external vocabularies, enabling agents wishing to reuse metadata to access source definitions.
- The availability of definitions for MAP components—entities, properties, SESs, and VESs—in these published sources.
- When a MAP specifies that values must come from a particular VES, the degree to which IRIs are entered accurately and match the source, or to which textual representations match the source exactly, can allow for future disambiguation of literal values and reconciliation with source vocabularies.
- The absence of conflicts with published definitions for entities, properties, and value constraint components that have been combined in a MAP.
- The extent to which catalogers and specialists creating metadata understood and adhered to MAP component definitions and value constraints, or

the extent to which a machine-readable MAP passed specifications to a data entry form or validation tool to restrict nonconformant values.

It is useful to consider technical or syntactic interoperability separately from semantic interoperability for several reasons. The applications and data formats used to manage and serve metadata are distinct from semantic models. Additionally, these applications and formats evolve rapidly and independently of semantic models and vocabularies.

### Supporting Technical Interoperability

Technical interoperability may be defined as the availability of a metadata instance in an encoded format that can be ingested, displayed, queried, or otherwise used by an external application. As differing applications require differing data formats, there will be many cases in which data for reuse will not be compatible with a new application. In these cases, clearly defined technical constraints in a MAP, such as SESs, will aid the processing of a metadata instance for reuse.

Semantic interoperability requires the meaning of and relationships between entities, properties, and values in a metadata instance to align with published definitions for the entities, properties, VESs, and SESs that were used to create it. Technical interoperability requires entities, properties, and values in an instance

to be encoded and formatted in a way that can be used by external systems or interpreted accurately to allow for reuse with a reasonable amount of processing. As a starting point, these must be available in a character encoding standard such as UTF-8 that can be read by the new system. In RDF implementations, it will be essential that all IRIs for entities, properties, and values be valid and actionable as Uniform Resource Locators (URLs). These will be needed to retrieve natural-language labels and other information needed for reuse and display of the data.

The requirements for technically interoperable values in particular can be challenging. Non-IRI values must be in a format that the new application can make use of, for example with date values formatted in a manner that can be accurately interpreted, queried, or displayed by the software. When a MAP specifies that textual or numeric values must conform to a particular SES, ensuring that values do so during metadata creation provides strong support for data reuse.

### Notes

1. Sinopia home page, Linked Data for Production 2 (LD4P2), <https://sinopia.io>.
2. “Metadata Interoperability,” Dublin Core Metadata Initiative, last updated May 6, 2021, [https://www.dublincore.org/resources/glossary/metadata\\_interoperability](https://www.dublincore.org/resources/glossary/metadata_interoperability).